9.3

# *IBM MQ konfigurieren*

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen” auf Seite 1165 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 3 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Wenn Sie Informationen an IBMsenden, erteilen Sie IBM ein nicht ausschließliches Recht, die Informationen in beliebiger Weise zu verwenden oder zu verteilen, ohne dass eine Verpflichtung für Sie entsteht.

# Inhaltsverzeichnis

# IBM MQ konfigurieren

Erstellen Sie einen oder mehrere Warteschlangenmanager auf einem oder mehreren Computern, und konfigurieren Sie sie auf Ihren Entwicklungs-, Test-und Produktionssystemen, um Nachrichten zu verarbeiten, die Ihre Geschäftsdaten enthalten.

### Informationen zu diesem Vorgang

Bevor Sie IBM MQ konfigurieren, lesen Sie die IBM MQ-Konzepte in IBM MQ Technical Overview. Hier erfahren Sie, wie Sie Ihre IBM MQ-Umgebung in Planung planen.

Es gibt eine Reihe verschiedener Methoden, die Sie für die Erstellung, Konfiguration und Verwaltung Ihrer Warteschlangenmanager und der zugehörigen Ressourcen in IBM MQ verwenden können. Zu diesen Methoden gehören Befehlszeilenschnittstellen, eine grafische Benutzerschnittstelle und eine Verwaltungs-API. Weitere Informationen zu diesen Schnittstellen finden Sie im Abschnitt Verwaltung von IBM MQ.

Anweisungen zum Erstellen, Starten, Stoppen und Löschen eines Warteschlangenmanagers finden Sie im Abschnitt „Warteschlangenmanager auf Multiplatforms erstellen" auf Seite 7.

Weitere Informationen zum Erstellen der Komponenten, die für eine gemeinsame Verbindung zu Ihren IBM MQ-Installationen und -Anwendungen erforderlich sind, finden Sie im Abschnitt „Verteilte Warteschlangensteuerung konfigurieren" auf Seite 206.

Anweisungen zum Verbinden Ihrer Clients mit einem IBM MQ-Server unter Verwendung unterschiedlicher Methoden finden Sie in „Verbindungen zwischen Client und Server konfigurieren" auf Seite 15.

Anweisungen zum Konfigurieren eines WS-Manager-Clusters finden Sie im Abschnitt „WS-Manager-Cluster konfigurieren" auf Seite 312.

Sie können das Verhalten von IBM MQ oder eines Warteschlangenmanagers ändern, indem Sie Konfigurationsinformationen ändern. Weitere Informationen finden Sie unter „IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern" auf Seite 90. Im Allgemeinen müssen Sie einen Warteschlangenmanager nicht erneut starten, damit die Konfigurationsänderungen wirksam werden, mit Ausnahme der in dieser Produktdokumentation genannten Änderungen.

z/OS Anweisungen zur Konfiguration von IBM MQ for z/OS finden Sie im Abschnitt „Warteschlangenmanager unter z/OS erstellen" auf Seite 966.

**Zugehörige Konzepte**

IBM MQ - Technische Übersicht

**Zugehörige Tasks**

Lokale IBM MQ-Objekte verwalten

Ferne IBM MQ-Objekte verwalten

IBM i IBMi verwalten

z/OS IBM MQ for z/OS verwalten

Planung

z/OS IBM MQ-Umgebung unter z/OS planen

„Warteschlangenmanager unter z/OS erstellen" auf Seite 966

Verwenden Sie diese Anweisungen zum Konfigurieren von Warteschlangenmanagern unter IBM MQ for z/OS.

## Multi Warteschlangenmanager auf Multiplatforms erstellen

Bevor Sie Nachrichten und Warteschlangen verwenden können, müssen Sie mindestens einen WS-Manager und die zugehörigen Objekte erstellen und starten. Ein Warteschlangenmanager verwaltet die Ressourcen, die ihm zugeordnet sind, insbesondere die Warteschlangen, die er besitzt. Er stellt Warte-

schlangenservices für Anwendungen für MQI-Aufrufe (Message Queuing Interface) und Befehle zum Erstellen, Ändern, Anzeigen und Löschen von IBM MQ-Objekten bereit.

## Vorbereitende Schritte

**Wichtig:** IBM MQ unterstützt keine Maschinennamen, die Leerzeichen enthalten. Wenn Sie IBM MQ auf einem Computer mit einem Maschinennamen installieren, der Leerzeichen enthält, können Sie keine WS-Manager erstellen.

Bevor Sie einen WS-Manager erstellen können, müssen Sie einige Punkte berücksichtigen, insbesondere in einer Produktionsumgebung. Arbeiten Sie durch die folgende Prüfliste:

**Die dem WS-Manager zugeordnete Installation**
Zum Erstellen eines Warteschlangenmanagers verwenden Sie den IBM MQ-Steuerbefehl **crtmqm**. Der Befehl **crtmqm** ordnet der Installation, von der der Befehl **crtmqm** ausgegeben wurde, automatisch einen Warteschlangenmanager zu. Für Befehle, die auf einem WS-Manager ausgeführt werden, müssen Sie den Befehl von der Installation absetzen, die dem Warteschlangenmanager zugeordnet ist. Sie können die zugeordnete Installation eines Warteschlangenmanagers mit dem Befehl setmqm ändern. Beachten Sie, dass das Windows-Installationsprogramm den Benutzer, der die Installation durchführt, nicht zur Gruppe 'mqm' hinzufügt. Weitere Informationen hierzu finden Sie unter Berechtigung zum Verwalten von IBM MQ unter AIX, Linux®, and Windows.

**Namenskonventionen**
Verwenden Sie Namen in Großbuchstaben, so dass Sie mit Warteschlangenmanagern auf allen Plattformen kommunizieren können. Denken Sie daran, dass Namen genau so zugeordnet werden, wie Sie sie eingeben. Verwenden Sie nicht unnötig lange Namen, um die Unannehmlichkeiten zu vermeiden, wenn Sie viele Eingabefehler eingeben.

**Geben Sie einen eindeutigen WS-Manager-Namen an**
Wenn Sie einen Warteschlangenmanager erstellen, müssen Sie sicherstellen, dass kein anderer Warteschlangenmanager denselben Namen an einer beliebigen Position in Ihrem Netz hat. WS-Manager-Namen werden beim Erstellen des Warteschlangenmanagers nicht überprüft, und Namen, die nicht eindeutig sind, verhindern, dass Kanäle für die verteilte Steuerung von Warteschlangen erstellt werden. Wenn Sie das Netz für Publish/Subscribe-Messaging verwenden, werden Subskriptionen außerdem dem Namen des WS-Managers zugeordnet, der sie erstellt hat. Wenn Warteschlangenmanager in dem Cluster oder in der Hierarchie denselben Namen haben, kann dies dazu führen, dass Veröffentlichungen nicht erreicht werden.

Eine Möglichkeit, die Eindeutigkeit zu gewährleisten, besteht darin, jedem WS-Managernamen seinen eigenen eindeutigen Knotennamen zu stellen. Wenn ein Knoten beispielsweise ACCOUNTS heißt, können Sie Ihren Warteschlangenmanager ACCOUNTS.SATURN.QUEUE.MANAGERnennen, wobei SATURN einen bestimmten Warteschlangenmanager identifiziert und QUEUE.MANAGER eine Erweiterung ist, die Sie allen Warteschlangenmanagern geben können. Alternativ können Sie diese Option weglassen, beachten Sie jedoch, dass ACCOUNTS.SATURN und ACCOUNTS.SATURN.QUEUE.MANAGER unterschiedliche Warteschlangenmanagernamen sind.

Wenn Sie IBM MQ für die Kommunikation mit anderen Unternehmen verwenden, können Sie auch Ihren eigenen Unternehmensnamen als Präfix einschließen. Dies wird in den Beispielen nicht gezeigt, da sie dadurch schwieriger zu folgen sind.

**Anmerkung:** Bei Warteschlangenmanagernamen in Steuerbefehlen muss die Groß-/Kleinschreibung beachtet werden. Dies bedeutet, dass Sie zwei Warteschlangenmanager mit den Namen `jupiter.queue.manager` und `JUPITER.queue.manager` erstellen dürfen. Es ist jedoch besser, solche Komplikationen zu vermeiden.

**Anzahl der WS-Manager begrenzen**
Sie können so viele WS-Manager wie Ressourcen zulassen. Da jeder WS-Manager jedoch seine eigenen Ressourcen benötigt, ist es im Allgemeinen besser, einen Warteschlangenmanager mit 100 Warteschlangen auf einem Knoten zu haben, als zehn Warteschlangenmanager mit jeweils zehn Warteschlangen zu haben.

In Produktionssystemen können viele Prozessoren mit einem einzigen Warteschlangenmanager genutzt werden, aber größere Servermaschinen werden möglicherweise effizienter mit mehreren Warteschlangenmanagern ausgeführt.

**Geben Sie einen Standardwarteschlangenmanager an**

Jeder Knoten sollte über einen Standardwarteschlangenmanager verfügen, obwohl es möglich ist, IBM MQ auf einem Knoten ohne einen solchen WS-Manager zu konfigurieren. Der Standardwarteschlangenmanager ist der Warteschlangenmanager, zu dem Anwendungen eine Verbindung herstellen, wenn er keinen Warteschlangenmanagernamen in einem `MQCONN` -Aufruf angegeben hat. Es ist außerdem der Warteschlangenmanager, der MQSC-Befehle verarbeitet, wenn Sie den Befehl `runmqsc` ohne Angabe eines Warteschlangenmanagernamens aufrufen.

Durch die Angabe eines Warteschlangenmanagers wird die vorhandene Standard-WS-Manager-Spezifikation für den Knoten standardmäßig ersetzt.

Das Ändern der Standardwarteschlangenverwaltung kann sich auf andere Benutzer oder Anwendungen auswirken. Die Änderung hat keine Auswirkungen auf derzeit verbundene Anwendungen, da sie die Kennung aus ihrem ursprünglichen Verbindungsaufruf in allen weiteren MQI-Aufrufen verwenden können. Mit dieser Kennung wird sichergestellt, dass die Aufrufe an denselben Warteschlangenmanager übertragen werden. Alle Anwendungen, die *nach* verbinden, haben die Verbindung des Standardwarteschlangenmanagers mit dem neuen Standardwarteschlangenmanager geändert. Dies ist möglicherweise die Absicht, die Sie beabsichtigen, aber Sie sollten dies berücksichtigen, bevor Sie die Standardeinstellung ändern.

Weitere Informationen zum Erstellen eines Standardwarteschlangenmanagers finden Sie unter „Standardwarteschlangenmanager erstellen” auf Seite 12.

**Geben Sie eine Warteschlange für nicht zustellbare Nachrichten an.**

Die Warteschlange für nicht zustellbare Nachrichten ist eine lokale Warteschlange, in die Nachrichten gestellt werden, wenn sie nicht an die Zieladresse weitergeleitet werden können.

Es ist wichtig, dass auf jedem Warteschlangenmanager in Ihrem Netzwerk eine Warteschlange für nicht zustellbare Nachrichten vorhanden ist. Wenn Sie keinen Fehler definieren, kann es zu Fehlern in den Anwendungsprogrammen kommen, dass Kanäle geschlossen werden, und Antworten auf Verwaltungsbefehle werden möglicherweise nicht empfangen.

Wenn eine Anwendung beispielsweise versucht, eine Nachricht in eine Warteschlange in einem anderen Warteschlangenmanager zu stellen, aber den falschen Warteschlangennamen angibt, wird der Kanal gestoppt und die Nachricht verbleibt in der Übertragungswarteschlange. Andere Anwendungen können diesen Kanal dann nicht für ihre Nachrichten verwenden.

Die Kanäle sind nicht betroffen, wenn die Warteschlangenmanager Warteschlangen für nicht zustellbare Nachrichten haben. Die unzustellbare Nachricht wird am empfangenden Ende in die Warteschlange für nicht zustellbare Nachrichten gestellt, wobei der Kanal und die zugehörige Übertragungswarteschlange verfügbar sind.

Verwenden Sie beim Erstellen eines Warteschlangenmanagers das Flag **-u**, um den Namen der Warteschlange für nicht zustellbare Nachrichten anzugeben. Sie können auch einen MQSC-Befehl verwenden, um die Attribute eines Warteschlangenmanagers zu ändern, den Sie bereits definiert haben, um die zu verwendende Warteschlange für nicht zustellbare Nachrichten anzugeben. Ein Beispiel für den MQSC-Befehl ALTER finden Sie im Abschnitt Warteschlangenmanagerattribute anzeigen und ändern.

**Geben Sie eine Standardübertragungswarteschlange an.**

Bei einer Übertragungswarteschlange handelt es sich um eine lokale Warteschlange, in der Nachrichten, die sich im Transit zu einem fernen Warteschlangenmanager befinden, vor der Übertragung in die Warteschlange gestellt werden. Die Standardübertragungswarteschlange ist die Warteschlange, die verwendet wird, wenn keine Übertragungswarteschlange explizit definiert ist. Jedem WS-Manager kann eine Standardübertragungswarteschlange zugeordnet werden.

Verwenden Sie beim Erstellen eines Warteschlangenmanagers das Flag **-d**, um den Namen der Standardübertragungswarteschlange anzugeben. Dadurch wird die Warteschlange nicht tatsächlich erstellt. Sie müssen dies später explizit tun. Weitere Informationen finden Sie im Abschnitt Mit lokalen Warteschlangen arbeiten .

**Geben Sie die erforderlichen Protokollierungsparameter an.**
Sie können Protokollierungsparameter für den Befehl `crtmqm` angeben, einschließlich der Art der Protokollierung und des Pfads und der Größe der Protokolldateien.

In einer Entwicklungsumgebung sollten die Standardprotokollierungsparameter ausreichend sein. Sie können die Standardwerte jedoch ändern, wenn Sie z. B.:

- Sie verfügen über eine Konfiguration mit niedrigem Endsystem, die keine großen Protokolle unterstützen kann.
- Es wird erwartet, dass eine große Anzahl langer Nachrichten gleichzeitig in den Warteschlangen enthalten ist.
- Sie erwarten eine Menge persistenter Nachrichten, die durch den Warteschlangenmanager passieren.

Nachdem Sie die Protokollierungsparameter festgelegt haben, können einige von ihnen nur geändert werden, indem der Warteschlangenmanager gelöscht und mit demselben Namen, aber mit unterschiedlichen Protokollierungsparametern erneut erstellt wird.

Weitere Informationen zu Protokollierungsparametern finden Sie im Abschnitt „Hochverfügbarkeit, Wiederherstellung und Neustart konfigurieren” auf Seite 501.

AIX **Nur für IBM MQ for UNIX-Systeme**
Sie können das Warteschlangenmanagerverzeichnis `/var/mqm/qmgrs/`*`qmgr`*auch in einem separaten lokalen Dateisystem erstellen, bevor Sie den Befehl **crtmqm** verwenden. Wenn Sie **crtmqm**verwenden und das Verzeichnis `/var/mqm/qmgrs/`*`qmgr`* vorhanden, leer und Eigentum von mqm ist, wird es für die Warteschlangenmanagerdaten verwendet. Wenn der Eigner des Verzeichnisses nicht 'mqm' ist, schlägt die Erstellung mit einer First Failure Support Technology-Nachricht ( FFST ) fehl. Wenn das Verzeichnis nicht leer ist, wird ein neues Verzeichnis erstellt.

## Informationen zu diesem Vorgang

Zum Erstellen eines Warteschlangenmanagers verwenden Sie den IBM MQ-Steuerbefehl **crtmqm**. Weitere Informationen finden Sie unter **crtmqm**. Der Befehl **crtmqm** erstellt automatisch die erforderlichen Standardobjekte und Systemobjekte (siehe Systemstandardobjekte). Standardobjekte bilden die Basis für alle Objektdefinitionen, die Sie machen. Systemobjekte sind für die WS-Manageroperation erforderlich.

Windows Auf Windows-Systemen haben Sie die Möglichkeit, mehrere Instanzen des Warteschlangenmanagers mit der Option *sax* des Befehls **crtmqm** zu starten.

Wenn Sie einen Warteschlangenmanager und seine Objekte erstellt haben, können Sie den Warteschlangenmanager mit dem Befehl **strmqm** starten.

## Prozedur

- Informationen zur Unterstützung bei der Erstellung und Verwaltung von Warteschlangenmanagern finden Sie in den folgenden Unterabschnitten:
  - „Standardwarteschlangenmanager erstellen” auf Seite 12
  - „Vorhandenen WS-Manager als Standardwert erstellen” auf Seite 14
  - „Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern” auf Seite 15

**Zugehörige Konzepte**
Mit Warteschlangenmanagern arbeiten
**Zugehörige Tasks**
Erstellen eines Warteschlangenmanagers mit dem Namen QM1
„IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern” auf Seite 90
Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen, indem Sie die Informationen in den Konfigurationsdateien (`.ini`) bearbeiten. Sie können auch Konfigurationsoptionen für IBM MQ MQI clientsändern.

„Warteschlangenmanager unter z/OS erstellen” auf Seite 966
Verwenden Sie diese Anweisungen zum Konfigurieren von Warteschlangenmanagern unter IBM MQ for z/OS.

**Zugehörige Verweise**
System-und Standardobjekte
crtmqm

## Linux AIX Konfigurierbares kurzlebiges Verzeichnis

Das konfigurierbare, kurzlebige Verzeichnis definiert die Position, an die kurzlebige Daten für den Warteschlangenmanager gesendet werden. Damit können die Sockets für die AIX and Linux-Domäne in einem nicht angehängten Dateisystem in einer Red Hat® OpenShift®-Umgebung platziert werden.

Vor IBM MQ 9.2.0 werden auf den AIX and Linux-Plattformen die AIX and Linux-Domänensockets unter dem Verzeichnis `/var/mqm/sockets` erstellt, wenn ein Warteschlangenmanager aktiv ist. Wenn Sie den Warteschlangenmanager in einem Container ausführen und `/var/mqm` als angehängtes Dateisystem verwenden, können einige Linux-Plattformen die Erstellung dieser Domänensockets verhindern, da sie einige Prozesse von außerhalb des Containers ermöglichen, um Operationen innerhalb des Containers zu stören. Dieses Problem verhindert, dass IBM MQ in einer Red Hat OpenShift-Containerplattform unter dem Standardsicherheitskontext ausgeführt wird.

Ab IBM MQ 9.2.0 kann mit dem Attribut **EphemeralPrefix** die Position des kurzlebigen Verzeichnisses konfiguriert werden. Wenn Sie dieses Attribut nicht verwenden, werden Sie keine Veränderung beim Verhalten feststellen.

Wenn in `mqs.ini` ein Warteschlangenmanagereintrag erstellt wird (mit dem Befehl **crtmqm** oder **addmqinf**), wird das Attribut **EphemeralPrefix** in folgenden Fällen hinzugefügt:

- Legen Sie das Attribut **DefaultEphemeralPrefix** in „Zeilengruppe 'AllQueueManagers' in der Datei 'mqs.ini'” auf Seite 96fest.
- Legen Sie die Umgebungsvariable **MQ_EPHEMERAL_PREFIX** fest.
- Bei der Angabe von **-v** EphemeralPrefix nur für den Befehl **addmqinf**.

Sie können das Attribut **EphemeralPrefix** auch explizit einem gestoppten Warteschlangenmanager hinzufügen. Das Attribut wird beim Neustart des Warteschlangenmanagers hinzugefügt.

Wenn Sie das Attribut **EphemeralPrefix** angeben, werden beim Start des Warteschlangenmanagers Daten, die für den Warteschlangenmanager kurzlebig sind, nicht an der üblichen Position, sondern unter diesem Präfix erstellt. Im Einzelnen bedeutet dies Folgendes:

- Socketdateien, die normalerweise unter `/var/mqm/sockets/<QM>` vorhanden sind, befinden sich jetzt unter `/<EphemeralPrefix>/sockets/<QM>`
- Subpooldateien, die normalerweise unter `/<Prefix>/qmgrs/<QM>/@<Subpool>` vorhanden sind, befinden sich jetzt unter `/<EphemeralPrefix>/qmgrs/<QM>/@<Subpool>`.

**Anmerkungen:**

- `/var/mqm/sockets/@SYSTEM` verbleibt an seiner festgelegten Position und ist nicht Teil des Attributs **EphemeralPrefix**.
- `AMQCLCHL.TAB` verbleibt unter `/<Prefix>/qmgrs/<QM>/@ipcc` und ist nicht Teil des Attributs **EphemeralPrefix** .

Die Anzahl der Zeichen, die das Attribut **EphemeralPrefix** einschließen kann, hängt von Ihrer Plattform ab:

- Linux AIX Auf AIX and Linux-Plattformen ist die Anzahl auf 12 Zeichen begrenzt.
- IBM i Unter IBM i ist sie auf 24 Zeichen begrenzt.

Wenn Sie ein **EphemeralPrefix**-Attribut angeben, das zu lang oder nicht vorhanden ist, erhalten Sie die Nachricht AMQ7001E:

```
AMQ7001E: The location specified for the queue manager is not valid
.
```

## Multi Verzeichnis 'userdata'

Sie können das Verzeichnis `userdata` verwenden, um den persistenten Anwendungsstatus zu speichern.

Jeder IBM MQ-Warteschlangenmanager verfügt über ein dediziertes Dateisystem für seinen persistenten Status, was sowohl seine Warteschlangendaten als auch das Wiederherstellungsprotokoll einschließt. Das Dateisystem enthält ein `userdata`-Verzeichnis, das Sie zum Speichern persistenter Statusinformationen für Ihre Anwendungen verwenden können. Siehe Verzeichnisinhalt auf UNIX- und Linux-Systemen und Verzeichnisinhalt auf Windows-Systemen.

Das Verzeichnis `userdata` kann in einer Reihe von Situationen nützlich sein, z. B.:

- In RDQM-Konfigurationen, da bei der Übernahme eines Warteschlangenmanagers auf einen anderen Knoten auch die Anwendungsinformationen verschoben werden (siehe „Persistenten Anwendungsstatus speichern” auf Seite 627).
- Bei Multi-Instanz-Warteschlangenmanagern, sodass sich deren Anwendungsstatusinformationen zusammen mit den Warteschlangenmanagerdaten im gemeinsam genutzten Netzdateisystem befinden.
- Generell dann, wenn Anwendungen als Warteschlangenmanagerservices konfiguriert sind.

Wenn Sie den Anwendungsstatus im Verzeichnis `userdata` speichern, müssen Sie sich darüber im Klaren sein, dass Daten, die an diese Position geschrieben werden, möglicherweise den verfügbaren Plattenspeicherplatz belegen, der dem Warteschlangenmanager zugeordnet ist. Sie müssen sicherstellen, dass für den Warteschlangenmanager genügend Plattenspeicherplatz zum Speichern von Warteschlangendaten, Protokollen und anderen persistenten Statusinformationen verfügbar ist.

Das Verzeichnis `userdata` hat den Benutzer 'mqm' und das Gruppeneigentum, und es ist weltweit lesbar, sodass Benutzer darauf zugreifen können, ohne sich in der Administratorgruppe von IBM MQ (d. h. mqm) befinden zu müssen. Sie können die Berechtigungen des Verzeichnisses `userdata` nicht ändern, aber Sie können Inhalte in ihr erstellen, unabhängig davon, welche Eigentümer und Berechtigungen Sie benötigen.

## Multi Standardwarteschlangenmanager erstellen

Der Standardwarteschlangenmanager ist der Warteschlangenmanager, zu dem Anwendungen eine Verbindung herstellen, wenn er in einem MQCONN-Aufruf keinen Warteschlangenmanagernamen angibt. Es ist auch der Warteschlangenmanager, der MQSC-Befehle verarbeitet, wenn Sie den Befehl **runmqsc** aufrufen, ohne einen Warteschlangenmanagernamen anzugeben. Zum Erstellen eines Warteschlangenmanagers verwenden Sie den IBM MQ-Steuerbefehl **crtmqm**.

### Vorbereitende Schritte

Bevor Sie einen Standardwarteschlangenmanager erstellen, lesen Sie die in „Warteschlangenmanager auf Multiplatforms erstellen” auf Seite 7 beschriebenen Hinweise.

Linux AIX Wenn Sie **crtmqm** verwenden, um einen Warteschlangenmanager unter AIX and Linux zu erstellen, wenn das Verzeichnis `/var/mqm/qmgrs/qmgr` bereits vorhanden, im Eigentum von mqm und leer ist, wird es für die Warteschlangenmanagerdaten verwendet. Wenn der Eigner des Verzeichnisses nicht mqm ist, schlägt die Erstellung des Warteschlangenmanagers mit einer First Failure Support Technology-Nachricht (FFST) fehl. Wenn das Verzeichnis nicht leer ist, wird ein neues Verzeichnis für die WS-Manager-Daten erstellt.

Diese Überlegung gilt auch dann, wenn das Verzeichnis `/var/mqm/qmgrs/qmgr` bereits in einem separaten lokalen Dateisystem vorhanden ist.

## Informationen zu diesem Vorgang

Wenn Sie einen Warteschlangenmanager mit dem Befehl **crtmqm** erstellen, erstellt der Befehl automatisch die erforderlichen Standardobjekte und Systemobjekte. Standardobjekte bilden die Basis für alle Objektdefinitionen, die Sie erstellen, und Systemobjekte sind für die WS-Manageroperation erforderlich.

Durch die Einbeziehung der relevanten Parameter in den Befehl können Sie beispielsweise auch den Namen der Standardübertragungswarteschlange definieren, die vom Warteschlangenmanager verwendet werden soll, und den Namen der Warteschlange für nicht zustellbare Nachrichten.

Windows Unter Windows können Sie die Option **sax** des Befehls **crtmqm** verwenden, um mehrere Instanzen des Warteschlangenmanagers zu starten.

Weitere Informationen zum Befehl **crtmqm** und seiner Syntax finden Sie unter **crtmqm**.

## Prozedur

- Verwenden Sie zum Erstellen eines Standardwarteschlangenmanagers den Befehl **crtmqm** mit dem Flag **-q** .

  Im folgenden Beispiel für den Befehl **crtmqm** wird ein Standardwarteschlangenmanager mit dem Namen SATURN.QUEUE.MANAGERerstellt:

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

  Dabei gilt:

  **-q**
  Gibt an, dass dieser WS-Manager der Standardwarteschlangenmanager ist.

  **-d MY.DEFAULT.XMIT.QUEUE**
  Der Name der Standardübertragungswarteschlange, die von diesem WS-Manager verwendet werden soll.

  **Anmerkung:** IBM MQ erstellt keine Standardübertragungswarteschlange für Sie. Sie müssen sie selbst definieren.

  **-u SYSTEM.DEAD.LETTER.QUEUE**
  Ist der Name der Standardwarteschlange für nicht zustellbare Nachrichten, die von IBM MQ bei der Installation erstellt wurde.

  **SATURN.QUEUE.MANAGER**
  Ist der Name dieses Warteschlangenmanagers. Dies muss der letzte Parameter sein, der im Befehl `crtmqm` angegeben wurde.

## Nächste Schritte

Wenn Sie einen Warteschlangenmanager und dessen Objekte erstellt haben, verwenden Sie den Befehl **strmqm** zum Starten des Warteschlangenmanagers.

**Zugehörige Konzepte**

Mit lokalen Warteschlangen arbeiten

**Zugehörige Tasks**

„Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern” auf Seite 15
Die Konfigurationsinformationen für IBM MQ werden unter AIX, Linux, and Windows in Konfigurationsdateien gespeichert. Sichern Sie nach der Erstellung eines Warteschlangenmanagers Ihre Konfigurationsdateien. Wenn Sie dann einen anderen WS-Manager erstellen, der Probleme verursacht, können Sie die Sicherungen erneut erstellen, wenn Sie die Ursache des Problems entfernt haben.

Warteschlangenmanagerattribute anzeigen und ändern

**Zugehörige Verweise**

System-und Standardobjekte

## Multi Vorhandenen WS-Manager als Standardwert erstellen

Sie können einen vorhandenen Warteschlangenmanager entweder manuell mithilfe eines Texteditors oder unter Windows und Linux mithilfe von IBM MQ Explorermanuell erstellen.

### Informationen zu diesem Vorgang

Wenn Sie einen Texteditor verwenden möchten, um einen vorhandenen Warteschlangenmanager als Standardwarteschlangenmanager zu erstellen, führen Sie die folgenden Schritte aus.

Windows Linux Wenn Sie auf Windows- und Linux-Systemen (x86- und x86-64-Plattformen) lieber den IBM MQ Explorer verwenden möchten, um diese Änderung vorzunehmen, lesen Sie den Abschnitt „Mit IBM MQ Explorer einen Warteschlangenmanager zum Standard machen” auf Seite 14.

Wenn Sie einen Standardwarteschlangenmanager erstellen, wird sein Name im Attribut `Name` der Zeilengruppe `DefaultQueueManager` in der Konfigurationsdatei IBM MQ (`mqs.ini`) eingefügt. Die Zeilengruppe und ihr Inhalt werden automatisch erstellt, wenn sie nicht vorhanden sind.

### Prozedur

- Wenn Sie einen vorhandenen Warteschlangenmanager als Standardwarteschlange verwenden möchten, ändern Sie den Namen des WS-Managers im Attribut `Name` in den Namen des neuen Standardwarteschlangenmanagers. Dies können Sie mit Hilfe eines Texteditors manuell ausführen.
- Wenn auf dem Knoten kein Standardwarteschlangenmanager vorhanden ist und Sie einen vorhandenen WS-Manager als Standardwarteschlange verwenden möchten, erstellen Sie die Zeilengruppe *`DefaultQueueManager`* mit dem erforderlichen Namen selbst.
- Wenn Sie versehentlich einen anderen Warteschlangenmanager als Standardwert verwenden und den ursprünglichen Standard-WS-Manager zurücksetzen möchten, bearbeiten Sie die Zeilengruppe `DefaultQueueManager` in `mqs.ini`, und ersetzen Sie dabei den nicht erwünschten Standardwarteschlangenmanager durch den gewünschten Standardwarteschlangenmanager, der von Ihnen gewünscht wird.

**Zugehörige Tasks**

„IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern” auf Seite 90
Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen, indem Sie die Informationen in den Konfigurationsdateien (`.ini`) bearbeiten. Sie können auch Konfigurationsoptionen für IBM MQ MQI clientsändern.

## Windows Linux Mit IBM MQ Explorer einen Warteschlangenmanager zum Standard machen

Auf Windows- und Linux-Systemen (x86- und x86-64-Plattformen) können Sie IBM MQ Explorer verwenden, um einen vorhandenen Warteschlangenmanager als Standardwarteschlangenmanager zu definieren.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um IBM MQ Explorer zu verwenden, um einen vorhandenen Warteschlangenmanager auf Windows- und Linux-Systemen (x86- und x86-64-Plattformen) als Standardwarteschlangenmanager zu definieren.

Wenn Sie diese Änderung lieber manuell in einem Texteditor vornehmen möchten, lesen Sie den Abschnitt „Vorhandenen WS-Manager als Standardwert erstellen” auf Seite 14.

### Vorgehensweise

1. Öffnen Sie IBM MQ Explorer.
2. Klicken Sie auf **IBM MQ** und wählen Sie dann **Eigenschaften ...** aus. Die Anzeige **Eigenschaften für IBM MQ** wird angezeigt.

3. Geben Sie den Namen des Standardwarteschlangenmanagers in das Feld **Name des Standardwarteschlangenmanagers** ein.
4. Klicken Sie auf **OK**.

## ALW Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern

Die Konfigurationsinformationen für IBM MQ werden unter AIX, Linux, and Windows in Konfigurationsdateien gespeichert. Sichern Sie nach der Erstellung eines Warteschlangenmanagers Ihre Konfigurationsdateien. Wenn Sie dann einen anderen WS-Manager erstellen, der Probleme verursacht, können Sie die Sicherungen erneut erstellen, wenn Sie die Ursache des Problems entfernt haben.

### Informationen zu diesem Vorgang

Sichern Sie Ihre Konfigurationsdateien in der Regel jedes Mal, wenn Sie einen neuen Warteschlangenmanager erstellen.

Es gibt zwei Typen von Konfigurationsdateien:

- Wenn Sie das Produkt installieren, wird die IBM MQ-Konfigurationsdatei (`mqs.ini`) erstellt. Sie enthält eine Liste der Warteschlangenmanager, die bei jedem Erstellen oder Löschen eines Warteschlangenmanagers aktualisiert wird. Pro Knoten gibt es eine `mqs.ini`-Datei.
- Wenn Sie einen neuen Warteschlangenmanager erstellen, wird automatisch eine neue WS-Manager-Konfigurationsdatei (`qm.ini`) erstellt. Dieser Parameter enthält Konfigurationsparameter für den Warteschlangenmanager.

Wenn Sie den AMQP-Service installiert haben, gibt es eine zusätzliche Konfigurationsdatei, die Sie sichern müssen:

- Windows Auf Windows-Systemen: `amqp_win.properties`
- Linux AIX Auf AIX and Linux-Systemen: `amqp_unix.properties`

**Zugehörige Tasks**

„IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern” auf Seite 90
Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen, indem Sie die Informationen in den Konfigurationsdateien (`.ini`) bearbeiten. Sie können auch Konfigurationsoptionen für IBM MQ MQI clientsändern.

„IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen” auf Seite 723
Sie können Warteschlangenmanager vor möglichen Beschädigungen durch Hardwarefehler schützen, indem Sie Warteschlangenmanager und WS-Manager-Daten sichern, nur die Konfiguration des Warteschlangenmanagers sichern und einen Sicherungswarteschlangenmanager verwenden.

# Verbindungen zwischen Client und Server konfigurieren

Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

### Informationen zu diesem Vorgang

In IBM MQ werden die logischen Kommunikationsverbindungen zwischen Objekten als *Kanäle* bezeichnet. Die Kanäle, die für die Verbindung von IBM MQ MQI clients zu Servern verwendet werden, werden als 'MQI-Kanäle' bezeichnet. Sie definieren Kanaldefinitionen an jedem Ende Ihrer Verbindung, damit Ihre IBM MQ-Anwendung auf dem IBM MQ MQI client mit dem Warteschlangenmanager auf dem Server kommunizieren kann.

Bevor Sie Ihre MQI-Kanäle definieren, müssen Sie entscheiden, in welcher Form der Kommunikation Sie die Verbindung verwenden möchten, und definieren Sie die Verbindung an jedem Ende des Kanals.

Wenn Sie einen MQI-Kanal zwischen einem IBM MQ MQI client und einem Warteschlangenmanager definieren, die sich in verschiedenen physischen Netzen befinden oder die über eine Firewall kommunizieren, kann die Verwendung von IBM MQ Internet Pass-Thru die Konfiguration vereinfachen. Weitere Informationen finden Sie unter IBM MQ Internet Pass-Thru.

### Vorgehensweise

1. Entscheiden Sie sich für die Form der Kommunikation, die Sie verwenden werden.
   Weitere Informationen finden Sie unter „Zu verwendende Übertragungsart" auf Seite 16.
2. Definieren Sie die Verbindung an jedem Ende des Kanals.
   Um die Verbindung zu definieren, müssen Sie folgende Schritte ausführen:
   a) Konfigurieren Sie die Verbindung.
   b) Notieren Sie die Werte der Parameter, die Sie für die Kanaldefinitionen benötigen.
   c) Aktivieren Sie den Server, um eingehende Netzanforderungen von Ihrem IBM MQ MQI client zu erkennen, indem Sie einen *Listener* starten.

**Zugehörige Konzepte**

„IBM MQ MQI client -Konfigurationsdatei, mqclient.ini" auf Seite 168
Sie können Ihre Clients mithilfe von Attributen in einer Textdatei konfigurieren. Diese Attribute können durch Umgebungsvariablen oder auf andere plattformspezifische Methoden überschrieben werden.

**Zugehörige Tasks**

„Verwendung von IBM MQ-Umgebungsvariablen" auf Seite 67
Sie können Befehle verwenden, um die aktuellen Einstellungen anzuzeigen oder um die Werte von IBM MQ-Umgebungsvariablen zurückzusetzen.

IBM MQ-MQI-Clientanwendungen mit Warteschlangenmanagern verbinden

**Zugehörige Verweise**

ANZEIGEN CHLAUTH

SET CHLAUTH

## Zu verwendende Übertragungsart

Unterschiedliche Plattformen unterstützen unterschiedliche Kommunikationsprotokolle. Ihre Auswahl des Übertragungsprotokolls hängt von Ihrer Kombination von IBM MQ MQI client- und Serverplattformen ab.

### Typen des Übertragungsprotokolls für MQI-Kanäle

Abhängig von Ihren Client-und Serverplattformen gibt es bis zu vier Typen von Übertragungsprotokolltypen für MQI-Kanäle:

- TCP/IP
- LU 6.2
- NetBIOS
- SPX

Wenn Sie Ihre MQI-Kanäle definieren, muss jede Kanaldefinition ein Übertragungsprotokoll (Transporttyp) angeben. Ein Server ist nicht auf ein Protokoll beschränkt, so dass verschiedene Kanaldefinitionen unterschiedliche Protokolle angeben können. Für IBM MQ MQI clients kann es nützlich sein, alternative MQI-Kanäle mit unterschiedlichen Übertragungsprotokollen zu verwenden.

Ihre Auswahl des Übertragungsprotokolls hängt auch von Ihrer speziellen Kombination von IBM MQ-Client- und Serverplattformen ab. Die möglichen Kombinationen sind in der folgenden Tabelle aufgeführt.

| *Tabelle 1. Übertragungsprotokolle - Kombination von IBM MQ MQI client- und Serverplattformen* | | |
|---|---|---|
| **Übertragungsprotokoll** | **IBM MQ MQI client** | **IBM MQ-Server** |
| TCP/IP<br>„1" auf Seite 17 | IBM i IBM i<br>AIX AIX<br>Linux Linux<br>Windows Windows | IBM i IBM i<br>AIX AIX<br>Linux Linux<br>Windows Windows<br>z/OS z/OS |
| LU 6.2 | AIX AIX<br>Linux Linux„2" auf Seite 17<br>Windows Windows | IBM i IBM i<br>AIX AIX<br>Linux Linux„2" auf Seite 17<br>Windows Windows<br>z/OS z/OS |
| NetBIOS | Windows Windows | Windows Windows |
| SPX | Windows Windows | Windows Windows |

**Anmerkungen:**

1. MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.
2. Mit Ausnahme von Linux (POWER-Plattform)

**Zugehörige Konzepte**

„TCP-Verbindung unter Windows definieren" auf Seite 276
Definieren Sie eine TCP-Verbindung, indem Sie einen Kanal auf der sendenden Seite konfigurieren, um die Adresse des Ziels anzugeben, und indem Sie ein Empfangsprogramm auf der Empfangsseite ausführen.

„TCP-Verbindung unter AIX and Linux definieren" auf Seite 284
Die Kanaldefinition auf der sendenden Seite gibt die Adresse des Ziels an. Der Listener-oder inet-Dämon ist für die Verbindung an der empfangenden Seite konfiguriert.

„TCP-Verbindung unter IBM i definieren" auf Seite 304
Sie können eine TCP-Verbindung in der Kanaldefinition mit Hilfe des Felds Verbindungsname definieren.

„TCP-Verbindung unter z/OS definieren" auf Seite 1074
Um eine TCP-Verbindung zu definieren, gibt es eine Reihe von Einstellungen für die Konfiguration.

„LU 6.2-Verbindung unter Windows definieren" auf Seite 278
SNA muss so konfiguriert werden, dass ein LU 6.2-Dialog zwischen den beiden Maschinen aufgebaut werden kann.

„LU 6.2-Verbindung unter AIX and Linux definieren" auf Seite 288
SNA muss so konfiguriert werden, dass ein LU 6.2-Dialog zwischen den beiden Maschinen aufgebaut werden kann.

„LU 6.2-Verbindung unter IBM i definieren” auf Seite 306
Definieren Sie die LU 6.2-Kommunikationsdetails unter Verwendung eines Modusnamens, eines TP-Namens und des Verbindungsnamens einer vollständig qualifizierten LU 6.2-Verbindung.

„NetBIOS-Verbindung in Windows definieren” auf Seite 280
Eine NetBIOS-Verbindung gilt nur für einen Client und Server, auf dem Windows ausgeführt wird. IBM MQ verwendet drei Typen von NetBIOS-Ressourcen, wenn eine NetBIOS-Verbindung zu einem anderen IBM MQ-Produkt hergestellt wird: Sitzungen, Befehle und Namen. Jede dieser Ressourcen hat einen Grenzwert, der entweder standardmäßig oder nach Auswahl während der Installation von NetBIOS festgelegt wird.

**Zugehörige Tasks**

„Aspera gateway -Verbindung auf Linux -oder Windows -Plattformen definieren” auf Seite 913
IBM Aspera faspio Gateway stellt einen schnellen TCP/IP-Tunnel bereit, der den Netzdurchsatz für IBM MQerheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert.

**Zugehörige Verweise**

„TCP/IP-Verbindungslimits” auf Seite 18
Die Anzahl der ausstehenden Verbindungsanforderungen, die an einem einzelnen TCP/IP-Port in die Warteschlange gestellt werden können, hängt von der Plattform ab. Ein Fehler tritt auf, wenn der Grenzwert erreicht ist.

„Definieren einer LU6.2-Verbindung für z/OS mit APPC/MVS” auf Seite 1077
Um eine LU6.2-Verbindung definieren zu können, müssen Sie eine Reihe von Einstellungen konfigurieren.

## TCP/IP-Verbindungslimits

Die Anzahl der ausstehenden Verbindungsanforderungen, die an einem einzelnen TCP/IP-Port in die Warteschlange gestellt werden können, hängt von der Plattform ab. Ein Fehler tritt auf, wenn der Grenzwert erreicht ist.

Dieses Verbindungslimit ist nicht mit der maximalen Anzahl der Clients identisch, die Sie an einen IBM MQ-Server anhängen können. Sie können mehr Clients mit einem Server verbinden, bis zu der Ebene, die von den Serversystemressourcen bestimmt wird. Die Rückprotokollwerte für Verbindungsanforderungen werden in der folgenden Tabelle angezeigt:

*Tabelle 2. Maximale Anzahl ausstehender Verbindungsanforderungen, die an einem TCP/IP-Port in die Warteschlange*

| **Serverplattform** | **Maximale Verbindungsanforderungen** |
|---|---|
| AIX AIX | 100 |
| Linux Linux | 100 |
| IBM i IBM i | 255 |
| Windows Windows-Server | 100 |
| Windows Windows-Workstation | 100 |
| z/OS z/OS | 255 |

Wenn der Verbindungsgrenzwert erreicht ist, empfängt der Client den Rückkehrcode MQRC_HOST_NOT_AVAILABLE aus dem Aufruf `MQCONN` und einen Fehler AMQ9202 im Clientfehlerprotokoll ( `/var/mqm/errors/AMQERR0n.LOG` auf AIX and Linux-Systemen oder `amqerr0n.log` im Unterverzeichnis 'errors' der IBM MQ-Clientinstallation unter Windows). Wenn der Client die `MQCONN` -Anforderung wiederholt, kann er erfolgreich sein.

Um die Anzahl der Verbindungsanforderungen zu erhöhen, die von dieser Einschränkung generiert werden, können Sie mehrere Empfangsprogramme, die jeweils an einem anderen Port empfangsbereit sind, oder mehrere Warteschlangenmanager haben, vermeiden.

# IBM MQ MQI client einrichten

In diesem Abschnitt wird beschrieben, wie ein Client eingerichtet wird.

## Vorbereitende Schritte

Damit ein IBM MQ MQI client eingerichtet werden kann, muss der IBM MQ-Server, zu dem der Client eine Verbindung herstellen soll, bereits installiert und aktiv sein.

## Vorgehensweise

1. Überprüfen Sie, ob Sie über eine geeignete Plattform für einen IBM MQ MQI-Client verfügen und ob Hardware und Software die Anforderungen erfüllen.

   Die Plattformunterstützung wird im Abschnitt Plattformunterstützung für IBM MQ -Clientsbeschrieben.
2. Entscheiden Sie, wie IBM MQ auf Ihrer Client-Workstation installiert werden soll, und befolgen Sie dann die Anweisungen für Ihre spezielle Kombination von Client und Serverplattformen.

   Die Installation wird in den folgenden Abschnitten beschrieben:

   - AIX IBM MQ-Client unter AIX installieren
   - Linux IBM MQ-Client unter Linux installieren
   - Windows IBM MQ-Client unter Windows installieren
   - IBM i IBM MQ-Client unter IBM i installieren
3. Stellen Sie sicher, dass die Kommunikationsverbindungen konfiguriert und verbunden sind.

   Die Konfiguration von Kommunikationsverbindungen wird im Abschnitt Verbindungen zwischen Server und Client konfigurieren beschrieben.
4. Überprüfen Sie, ob die Installation fehlerfrei arbeitet.

   Lesen Sie den Abschnitt zur Überprüfung des Installationsverfahrens für die Plattform bzw. Plattformen, die von Ihrem Unternehmen verwendet wird bzw. werden.
5. Nach der Überprüfung der IBM MQ MQI clientinstallation müssen Sie sich überlegen, ob Sicherheitsfunktionen für den Client konfiguriert werden müssen.

   Die Clientsicherheit wird im Abschnitt IBM MQ MQI client-Sicherheit einrichten beschrieben.
6. Richten Sie die Kanäle zwischen dem IBM MQ MQI-Client und -Server ein, die für die IBM MQ -Anwendungen erforderlich sind, die Sie auf dem Client ausführen wollen.

   Die Einrichtung von Kanälen wird im Abschnitt MQI-Kanäle definieren beschrieben. Bei Verwendung von TLS müssen zusätzliche Punkte berücksichtigt werden.

   Diese Punkte werden im Abschnitt Verwendung von TLS für einen MQI-Kanal konfigurieren beschrieben. Unter Umständen müssen Sie die Kanäle mithilfe einer IBM MQ MQI clientkonfigurationsdatei oder mithilfe von IBM MQ-Umgebungsvariablen einrichten. IBM MQ-Umgebungsvariablen werden im Abschnitt IBM MQ-Umgebungsvariablen verwenden beschrieben.
7. Eine vollständige Beschreibung der IBM MQ -Anwendungen finden Sie unter Anwendungen entwickeln .
8. Beim Entwerfen, Erstellen und Ausführen von Anwendungen in der IBM MQ MQI client -Umgebung müssen Sie die Unterschiede zu einer Warteschlangenmanagerumgebung berücksichtigen.

   Informationen zu diesen Unterschieden finden Sie in den folgenden Abschnitten:

   - Schnittstelle für Nachrichtenwarteschlangen (MQI) in einer Clientanwendung verwenden
   - Anwendungen für IBM MQ MQI clients erstellen
   - IBM MQ MQI client-Anwendungen mit Warteschlangenmanagern verbinden

- Probleme mit IBM MQ MQI clients beheben

# Einen erweiterten transaktionsorientierten Client konfigurieren

In dieser Themensammlung wird beschrieben, wie die erweiterte transaktionsorientierte Funktion für jede Kategorie von Transaktionsmanager konfiguriert wird.

Für jede Plattform bietet der erweiterte transaktionsorientierte Client Unterstützung für die folgenden externen Transaktionsmanager:

**XA-kompatible Transaktionsmanager**
Der erweiterte transaktionsorientierte Client stellt die Schnittstelle des XA-Ressourcenmanagers bereit, um XA-kompatible Transaktionsmanager wie z. B. CICS und Tuxedo zu unterstützen.

Windows **Microsoft Transaction Server (nur Windows-Systeme)**
Nur auf Windows-Systemen unterstützt die Schnittstelle des XA-Ressourcenmanagers auch Microsoft Transaction Server (MTS). Die IBM MQ MTS-Unterstützung, die mit dem erweiterten transaktionsorientierten Client geboten wird, stellt die Brücke zwischen MTS und der Schnittstelle des XA-Ressourcenmanagers bereit.

**WebSphere Application Server**
WebSphere Application Server 6 und höher enthält einen IBM MQ -Messaging-Provider, sodass Sie den erweiterten transaktionsorientierten Client nicht verwenden müssen.

## XA-kompatible Transaktionsmanager konfigurieren

Konfigurieren Sie zuerst den IBM MQ-Basisclient und konfigurieren Sie anschließend die erweiterte transaktionsorientierte Funktion mit Hilfe der Informationen in diesen Themen.

**Anmerkung:** In diesem Abschnitt wird davon ausgegangen, dass Sie über ein grundlegendes Verständnis für die XA-Schnittstelle verfügen, die von The Open Group in *Distributed Transaction Processing: The XA Specification* veröffentlicht wurde.

Wenn Sie einen erweiterten transaktionsorientierten Client konfigurieren möchten, müssen Sie zuerst den IBM MQ-Basisclient konfigurieren. Gehen Sie dabei wie in folgenden Abschnitten beschrieben vor:

- AIX IBM MQ-Client unter AIX installieren
- Linux IBM MQ-Client unter Linux installieren
- Windows IBM MQ-Client unter Windows installieren
- IBM i IBM MQ-Client unter IBM i installieren

Mithilfe der Informationen für Ihre jeweilige Plattform können Sie dann die erweiterte transaktionsorientierte Funktion für einen XA-konformen Transaktionsmanager wie CICS und Tuxedo konfigurieren.

Ein Transaktionsmanager kommuniziert mit einem Warteschlangenmanager als Ressourcenmanager, der denselben MQI-Kanal verwendet, wie er von der Clientanwendung verwendet wird, die mit dem Warteschlangenmanager verbunden ist. Wenn der Transaktionsmanager einen Funktionsaufruf des Ressourcenmanagers (xa_) ausgibt, wird der MQI-Kanal verwendet, um den Aufruf an den Warteschlangenmanager weiterzuleiten und die Ausgabe vom WS-Manager zurückzuerhalten.

Entweder kann der Transaktionsmanager den MQI-Kanal starten, indem er einen xa_open-Aufruf ausgibt, um den Warteschlangenmanager als Ressourcenmanager zu öffnen, oder die Clientanwendung kann den MQI-Kanal starten, indem er einen MQCONN-oder MQCONNX-Aufruf ausgibt.

- Wenn der Transaktionsmanager den MQI-Kanal startet und die Clientanwendung später MQCONN oder MQCONNX in demselben Thread aufruft, wird der MQCONN-oder MQCONNX-Aufruf erfolgreich ausgeführt und eine Verbindungskennung wird an die Anwendung zurückgegeben. Die Anwendung empfängt keinen Beendigungscode MQCC_WARNING mit einem Ursachencode MQRC_ALREADY_CONNECTED.

- Wenn die Clientanwendung den MQI-Kanal startet und der Transaktionsmanager später xa_open im selben Thread aufruft, wird der xa_open-Aufruf unter Verwendung des MQI-Kanals an den Warteschlangenmanager weitergeleitet.

In einer Wiederherstellungssituation nach einem Fehler, wenn keine Clientanwendungen ausgeführt werden, kann der Transaktionsmanager einen dedizierten MQI-Kanal verwenden, um alle unvollständigen Arbeitseinheiten wiederherzustellen, an denen der Warteschlangenmanager zum Zeitpunkt des Fehlers beteiligt war.

Beachten Sie die folgenden Bedingungen, wenn Sie einen erweiterten transaktionsorientierten Client mit einem XA-konformen Transaktionsmanager verwenden:

- In einem einzelnen Thread kann eine Clientanwendung immer nur mit einem Warteschlangenmanager verbunden sein. Diese Einschränkung gilt nur bei Verwendung eines erweiterten transaktionsorientierten Clients; eine Clientanwendung, die einen IBM MQ-Basisclient verwendet, kann gleichzeitig mit mehr als einem Warteschlangenmanager in einem einzigen Thread verbunden werden.
- Jeder Thread einer Clientanwendung kann eine Verbindung zu einem anderen WS-Manager herstellen.
- Eine Clientanwendung kann keine gemeinsam genutzten Verbindungskennungen verwenden.

Um die erweiterte transaktionsorientierte Funktion zu konfigurieren, müssen Sie dem Transaktionsmanager die folgenden Informationen für jeden Warteschlangenmanager bereitstellen, der als Ressourcenmanager fungiert:

- Eine xa_open-Zeichenfolge
- Ein Zeiger auf eine XA-Switchstruktur

Wenn der Transaktionsmanager xa_open aufruft, um den Warteschlangenmanager als Ressourcenmanager zu öffnen, übergibt er die xa_open-Zeichenfolge an den erweiterten transaktionsorientierten Client als Argument *xa_info* für den Aufruf. Der erweiterte transaktionsorientierte Client verwendet die Informationen in der Zeichenfolge xa_open auf die folgenden Arten:

- Gehen Sie wie folgt vor, um einen MQI-Kanal zum Server-WS-Manager zu starten, wenn die Clientanwendung noch nicht gestartet wurde.
- Überprüfen Sie, ob der Warteschlangenmanager, der der Transaktionsmanager als Ressourcenmanager geöffnet wird, mit dem Warteschlangenmanager identisch ist, zu dem die Clientanwendung eine Verbindung herstellt.
- Um die Funktionen ax_reg und ax_unreg des Transaktionsmanagers zu lokalisieren, wenn der Warteschlangenmanager die dynamische Registrierung verwendet

Für das Format einer xa_open-Zeichenfolge und für weitere Informationen darüber, wie die Informationen in der xa_open-Zeichenfolge von einem erweiterten transaktionsorientierten Client verwendet werden, finden Sie weitere Informationen in „Das Format einer xa_open-Zeichenfolge.” auf Seite 23.

Eine XA-Switchstruktur ermöglicht es dem Transaktionsmanager, die vom erweiterten transaktionsorientierten Client bereitgestellten xa-Funktionen zu lokalisieren und gibt an, ob der Warteschlangenmanager die dynamische Registrierung verwendet. Informationen zu den XA-Switch-Strukturen, die mit einem erweiterten transaktionsorientierten Client bereitgestellt werden, finden Sie in „Die XA-Switchstrukturen” auf Seite 27.

Informationen zum Konfigurieren der erweiterten transaktionsorientierten Funktion für einen bestimmten Transaktionsmanager und weitere Informationen zur Verwendung des Transaktionsmanagers mit einem erweiterten transaktionsorientierten Client finden Sie in den folgenden Abschnitten:

- „Konfigurieren eines erweiterten transaktionsorientierten Clients für CICS” auf Seite 28
- „Konfigurieren eines erweiterten transaktionsorientierten Clients für Tuxedo” auf Seite 30

**Zugehörige Konzepte**

„Die Parameter CHANNEL, TRPTYPE, CONNAME und QMNAME der Zeichenfolge 'xa_open'” auf Seite 25
Verwenden Sie diese Informationen, um zu verstehen, wie der erweiterte transaktionsorientierte Client diese Parameter verwendet, um den Warteschlangenmanager zu ermitteln, zu dem eine Verbindung hergestellt werden soll.

„Zusätzliche Fehlerverarbeitung für xa_open” auf Seite 26
Der Aufruf 'xa_open' schlägt unter bestimmten Umständen fehl.

**Zugehörige Tasks**

„Extended Transactional Client mit TLS-Kanälen verwenden” auf Seite 28
Sie können keinen TLS-Kanal mit der Zeichenfolge 'xa_open' konfigurieren. Führen Sie die folgenden Anweisungen aus, um die Definitionstabelle für den Clientkanal (ccdt) zu verwenden.

**Zugehörige Verweise**

„Die Parameter TPM und AXLIB” auf Seite 26
Ein erweiterter transaktionsorientierter Client verwendet die Parameter "TPM" und "AXLIB", um die Funktionen "ax_reg" und "ax_unreg" des Transaktionsmanagers zu lokalisieren. Diese Funktionen werden nur verwendet, wenn der WS-Manager eine dynamische Registrierung verwendet.

„Wiederherstellung nach einem Fehler in der erweiterten transaktionsorientierten Verarbeitung” auf Seite 26
Nach einem Fehler muss ein Transaktionsmanager in der Lage sein, alle unvollständigen Arbeitseinheiten wiederherzustellen. Dazu muss der Transaktionsmanager in der Lage sein, als Ressourcenmanager einen beliebigen Warteschlangenmanager zu öffnen, der zum Zeitpunkt des Ausfalls an einer unvollständigen UO- Unit beteiligt war.

### z/OS *Hinweise zu IBM MQ for z/OS für erweiterte transaktionsorientierte Clientverbindungen*

Einige XA-Transaktionsmanager verwenden Sequenzen von Transaktionskoordinieraufrufen, die mit den Funktionen, die normalerweise für Clients verfügbar sind, die eine Verbindung zu IBM MQ for z/OS herstellen, nicht kompatibel sind.

Wenn eine inkompatible Sequenz erkannt wird, gibt IBM MQ for z/OS möglicherweise einen Abbruch für die Verbindung aus und gibt eine Fehlerantwort an den Client zurück.

Beispiel: `xa_prepare` empfängt die abnormale Beendigung 5C6-00D4007D, wobei der Rückkehrcode -3 (XAER_RMERR) an den Client zurückgegeben wird.

Ein weiteres Beispiel ist, dass `xa_end` abnormale Beendigung 5C6-00D40079empfängt.

Führen Sie für Transaktionsmanager, die auf diese Situation stoßen, die folgende Aktion aus, damit der Transaktionsmanager mit IBM MQ for z/OSinteragieren kann:

Stellen Sie sicher, dass Sie Änderungen an XA-Client-Verbindungen in IBM MQ for z/OS aktiviert haben, die es dem Transaktionsmanager ermöglichen, eine Transaktion in einer anderen Verbindung vorzubereiten.

**Anmerkungen:**

- Die Änderung ist standardmäßig nicht aktiviert. Um die Änderung zu verwenden, müssen Sie das Schlüsselwort CSQSERVICE1 (in Großbuchstaben) an einer beliebigen Stelle im Beschreibungsfeld des SVRCONN-Kanals angeben, der vom XA-Client verwendet wird.
- Für Kanäle mit dem Schlüsselwort CSQSERVICE1 gelten folgende Einschränkungen:
  - Die Disposition GROUP der Wiederherstellungs-Disposition ist nicht zulässig. Es ist nur die Disposition QMGR der Wiederherstellungs-Disposition zulässig. Die Disposition wird durch den im Aufruf `xa_open` angegebenen Namen bestimmt. Wenn der Name der Gruppe mit gemeinsamer Warteschlange verwendet wird, fordert die XA-Verbindung eine Arbeitseinheit mit Wiederherstellung an.

    Ein `xa_open`-Aufruf, in dem der Name der Gruppe mit gemeinsamer Warteschlange im Parameter **`xa_info`** angegeben wird, schlägt mit *xaer_inval* fehl.
  - Die Optionen *MQGMO_LOCK* und *MQGMO_UNLOCK* sind nicht zulässig. Ein MQGET-Aufruf mit *MQGMO_LOCK* oder *MQGMO_UNLOCK* schlägt mit MQRC_ENVIRONMENT_ERROR fehl.

Die Änderung wurde in IBM MQ for z/OS 9.0 über APAR P173410 aktiviert.

**Zugehörige Konzepte**

„XA-kompatible Transaktionsmanager konfigurieren” auf Seite 20

Konfigurieren Sie zuerst den IBM MQ-Basisclient und konfigurieren Sie anschließend die erweiterte transaktionsorientierte Funktion mit Hilfe der Informationen in diesen Themen.

### *Das Format einer xa_open-Zeichenfolge.*

Eine xa_open-Zeichenfolge enthält Paare definierter Parameternamen und -werte.

Eine xa_open-Zeichenfolge hat das folgende Format:

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

Dabei ist *parm_name* der Name eines Parameters und *parm_value* der Wert eines Parameters. Die Namen der Parameter sind nicht die Groß-/Kleinschreibung, es sei denn, die Werte der Parameter werden von der Groß-/Kleinschreibung abhängig gemacht. Sie können die Parameter in beliebiger Reihenfolge angeben.

Die Namen, die Bedeutungen und die gültigen Werte der Parameter lauten wie folgt:

**Name**
**Bedeutung und gültige Werte**

**CHANNEL**
Der Name eines MQI-Kanals.

Dies ist ein optionaler Parameter. Wenn dieser Parameter angegeben wird, muss auch der Parameter CONNAME angegeben werden.

**TRPTYPE**
Das Übertragungsprotokoll für den MQI-Kanal. Die folgenden Protokolle sind gültige Werte:

**LU62**
SNA LU 6.2

**NETBIOS**
NetBIOS

**SPX**
IPX/SPX

**TCP**
TCP/IP

Dies ist ein optionaler Parameter. Wird dieser Parameter nicht angegeben, wird der Standardwert TCP angenommen. Bei den Werten des Parameters muss die Groß-/Kleinschreibung nicht beachtet werden.

**CONNAME**
Die Netzadresse des WS-Managers auf dem Serverende des MQI-Kanals. Die gültigen Werte für diesen Parameter hängen vom Wert des Parameters TRPTYPE ab:

**LU62**
Ein symbolischer Bestimmungsname, der einen CPI-C-Nebeninformationen-Eintrag identifiziert.

Der netzqualifizierte Name einer Partner-LU ist kein gültiger Wert und kein Aliasname der Partner-LU. Dies liegt daran, dass es keine zusätzlichen Parameter gibt, um einen Transaktionsprogrammnamen (TP) und einen Modusnamen anzugeben.

**NETBIOS**
Ein NetBIOS-Name.

**SPX**
Eine 4-Byte-Netzadresse, eine 6-Byte-Knotenadresse und eine optionale 2-Byte-Socket-Nummer. Diese Werte müssen in Hexadezimalschreibweise angegeben werden. Ein Punkt muss die Netz- und Knotenadressen voneinander trennen, und die Socket-Nummer, falls angegeben, muss in runde Klammern eingeschlossen werden. For example:

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

Wenn die Socketnummer weggelassen wird, wird der Standardwert 5e86 angenommen.

**TCP**
Ein Hostname oder eine IP-Adresse, optional gefolgt von einer Port-Nummer in runden Klammern. Wenn die Portnummer nicht angegeben wird, wird der Standardwert 1414 angenommen. Mehrere Hosts und Ports für einen Warteschlangenmanager können unter Verwendung eines Semikolon-Trennzeichens angegeben werden, z. B.:

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

Dies ist ein optionaler Parameter. Wenn dieser Parameter angegeben wird, muss auch der Parameter CHANNEL angegeben werden.

**QMNAME**
Der Name des Warteschlangenmanagers am Serverende des MQI-Kanals. Der Name darf weder leer noch ein einzelner Stern (*) sein, noch kann der Name mit einem Stern beginnen. Dies bedeutet, dass der Parameter einen bestimmten WS-Manager nach Namen identifizieren muss.

Dies ist ein obligatorischer Parameter.

Wenn eine Clientanwendung mit einem bestimmten WS-Manager verbunden ist, muss jede Transaktionswiederherstellung von demselben Warteschlangenmanager verarbeitet werden.

Wenn die Anwendung eine Verbindung zu einem z/OS-Warteschlangenmanager herstellt, kann die Anwendung entweder den Namen eines bestimmten Warteschlangenmanagers oder den Namen einer Gruppe mit gemeinsamer Warteschlange (QSG) angeben. Durch die Verwendung des Warteschlangenmanagernamens oder des Namens der Gruppe mit gemeinsamer Warteschlange steuert die Anwendung, ob sie an einer Transaktion mit einer Disposition QMGR der Arbeitseinheit mit Wiederherstellung oder eine Disposition GROUP der Arbeitseinheit mit Wiederherstellung teilnimmt. Die Disposition GROUP der Wiederherstellung ermöglicht die Wiederherstellung der Transaktion, die in einem beliebigen Mitglied der Gruppe QSG verarbeitet werden soll. Um GROUP-Arbeitseinheiten mit Wiederherstellung verwenden zu können, muss das WS-Manager-Attribut von **GROUPUR** aktiviert sein.

z/OS Weitere Informationen zur Verwendung der GROUP-Arbeitseinheit mit Wiederherstellung finden Sie unter Disposition der Arbeitseinheit mit Wiederherstellung in einer Gruppe mit gemeinsamer Warteschlange.

**TPM**
Der Transaktionsmanager, der verwendet wird. Die gültigen Werte sind CICS und TUXEDO.

Ein erweiterter transaktionsorientierter Client verwendet diesen Parameter und den Parameter AXLIB für den gleichen Zweck. Weitere Informationen zu diesen Parametern finden Sie in den Parametern für TPM und AXLIB .

Dies ist ein optionaler Parameter. Bei den Werten des Parameters muss die Groß-/Kleinschreibung nicht beachtet werden.

**AXLIB**
Der Name der Bibliothek, die die Funktionen "ax_reg" und "ax_unreg" des Transaktionsmanagers enthält.

Dies ist ein optionaler Parameter.

**UID**
Die Benutzer-ID, die dem Warteschlangenmanager für die Authentifizierung zur Verfügung gestellt wird. Wenn dieser Parameter angegeben wird, muss auch der Parameter **PWD** angegeben werden. Wenn die angegebene Benutzer-ID und das Kennwort authentifiziert sind, wird die Benutzer-ID für die Identifikation der Verbindung des Transaktionsmanagers verwendet. Die Benutzer-ID und das Kennwort füllen das MQCSP-Objekt im MQCONNX-Aufruf mit Daten aus.

Die Parameter **UID** und **PWD** sind sowohl für Client-als auch für Serverbindungsbindungen gültig.

**PWD**
Das Kennwort, das dem WS-Manager zur Authentifizierung zur Verfügung gestellt wird. Wenn dieser Parameter angegeben wird, muss auch der Parameter **UID** angegeben werden.

**Warnung:** In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Die Informationen im Abschnitt IBM MQCSP-Kennwortschutz erläutern, wie Sie sicherstellen können, dass Clientanwendungskennwörter angemessen geschützt sind.

Hier ist ein Beispiel für eine xa_open-Zeichenfolge:

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

### *Die Parameter CHANNEL, TRPTYPE, CONNAME und QMNAME der Zeichenfolge 'xa_open'*

Verwenden Sie diese Informationen, um zu verstehen, wie der erweiterte transaktionsorientierte Client diese Parameter verwendet, um den Warteschlangenmanager zu ermitteln, zu dem eine Verbindung hergestellt werden soll.

Wenn die Parameter **CHANNEL** und **CONNAME** in der Zeichenfolge xa_open angegeben werden, verwendet der erweiterte transaktionsorientierte Client diese Parameter und den Parameter **TRPTYPE** , um einen MQI-Kanal zum Server-WS-Manager zu starten.

Wenn die Parameter **CHANNEL** und **CONNAME** nicht in der Zeichenfolge 'xa_open' angegeben werden, verwendet der erweiterte transaktionsorientierte Client den Wert der Umgebungsvariablen MQSERVER, um einen MQI-Kanal zu starten. Wenn die Umgebungsvariable MQSERVER nicht definiert ist, verwendet der erweiterte transaktionsorientierte Client den Eintrag in der Clientkanaldefinition, die durch den Parameter **QMNAME** angegeben wird.

In jedem dieser Fälle überprüft der erweiterte transaktionsorientierte Client, ob der Wert des Parameters **QMNAME** der Name des Warteschlangenmanagers am Serverende des MQI-Kanals ist. Ist dies nicht der Fall, schlägt der Aufruf 'xa_open' fehl, und der Transaktionsmanager meldet den Fehler an der Anwendung.

z/OS Wenn die Anwendung den Namen einer Gruppe mit gemeinsamer Warteschlange im Parameterfeld **QMNAME** verwendet und die Eigenschaft GROUPUR auf dem Warteschlangenmanager inaktiviert ist, zu dem sie eine Verbindung herstellt, schlägt der Aufruf von xa_open fehl.

z/OS Wenn der Anwendungsclient eine Verbindung zu einem z/OS -Warteschlangenmanager herstellt, kann er für den Parameter **QMNAME** einen Namen für die Gruppe mit gemeinsamer Warteschlange angeben. Dies ermöglicht dem Anwendungsclient die Teilnahme an einer Transaktion mit einer Disposition der Gruppe 'GROUP' der Disposition. Weitere Informationen zur Disposition der Arbeitseinheit mit Wiederherstellung "GROUP" finden Sie unter Disposition der Arbeitseinheit mit Wiederherstellung.

Wenn die Clientanwendung MQCONN oder MQCONNX in demselben Thread aufruft, den der Transaktionsmanager zum Absetzen des Xa_open-Aufrufs verwendet hat, empfängt die Anwendung eine Verbindungskennung für den MQI-Kanal, der durch den Aufruf 'xa_open' gestartet wurde. Ein zweiter MQI-Kanal wurde nicht gestartet. Der erweiterte transaktionsorientierte Client überprüft, ob der Wert des Parameters **QMgrName** im MQCONN-oder MQCONNX-Aufruf der Name des Warteschlangenmanagers am Serverende des MQI-Kanals ist. Ist dies nicht der Fall, schlägt der MQCONN-oder MQCONNX-Aufruf mit einem Ursachencode von MQRC_ANOTHER_Q_MGR_CONNECTED fehl. Wenn der Wert des Parameters **QMgrName** leer oder ein einzelner Stern (*) ist oder mit einem Stern beginnt, schlägt der MQCONN-oder MQCONNX-Aufruf mit dem Ursachencode MQRC_Q_MGR_NAME_ERROR fehl.

Wenn die Clientanwendung bereits einen MQI-Kanal gestartet hat, indem sie MQCONN oder MQCONNX aufruft, bevor der Transaktionsmanager xa_open im selben Thread aufruft, verwendet der Transaktionsmanager stattdessen diesen MQI-Kanal. Ein zweiter MQI-Kanal wurde nicht gestartet. Der erweiterte transaktionsorientierte Client überprüft, ob der Wert des Parameters **QMNAME** in der Zeichenfolge 'xa_open' der Name des Server-WS-Managers ist. Ist dies nicht der Fall, schlägt der Aufruf 'xa_open' fehl.

Wenn eine Clientanwendung zuerst einen MQI-Kanal startet, kann der Wert des Parameters **QMgrName** im MQCONN-oder MQCONNX-Aufruf leer oder ein einzelner Stern (*) sein, oder er kann mit einem Stern

beginnen. Unter diesen Umständen müssen Sie jedoch sicherstellen, dass der Warteschlangenmanager, zu dem die Anwendung eine Verbindung herstellt, mit dem Warteschlangenmanager identisch ist, den der Transaktionsmanager als Ressourcenmanager öffnen will, wenn er zu einem späteren Zeitpunkt xa_open im selben Thread aufruft. Es treten möglicherweise weniger Probleme auf, wenn der Wert des Parameters *QMgrName* den Warteschlangenmanager explizit anhand des Namens identifiziert.

### *Die Parameter TPM und AXLIB*

Ein erweiterter transaktionsorientierter Client verwendet die Parameter "TPM" und "AXLIB", um die Funktionen "ax_reg" und "ax_unreg" des Transaktionsmanagers zu lokalisieren. Diese Funktionen werden nur verwendet, wenn der WS-Manager eine dynamische Registrierung verwendet.

Wenn der TPM-Parameter in einer xa_open-Zeichenfolge angegeben wird, der Parameter AXLIB jedoch nicht angegeben wird, nimmt der erweiterte transaktionsorientierte Client einen Wert für den Parameter AXLIB an, der auf dem Wert des Parameters TPM basiert. Informationen zu den angenommenen Werten für den Parameter AXLIB finden Sie im Abschnitt Tabelle 3 auf Seite 26.

*Tabelle 3. Angenommen, Werte des Parameters AXLIB*

| Wert von TPM | Plattform | Assumiereter Wert von AXLIB |
|---|---|---|
| CICS | AIX | /usr/lpp/encina/lib/libEncServer.a (EncServer_shr.o) |
| CICS | Systeme mit Windows | libEncServer |
| Tuxedo | AIX | /usr/lpp/tuxedo/lib/libtux.a(libtux.so.60) |
| Tuxedo | Systeme mit Windows | libtux |

Wenn der Parameter AXLIB in einer xa_open-Zeichenfolge angegeben wird, verwendet der erweiterte transaktionsorientierte Client seinen Wert, um jeden angenommenen Wert basierend auf dem Wert des TPM-Parameters zu überschreiben. Der Parameter AXLIB kann auch für einen Transaktionsmanager verwendet werden, für den der TPM-Parameter keinen angegebenen Wert hat.

### *Zusätzliche Fehlerverarbeitung für xa_open*

Der Aufruf 'xa_open' schlägt unter bestimmten Umständen fehl.

In den Themen in diesem Abschnitt werden Situationen beschrieben, in denen der Aufruf 'xa_open' fehlschlägt. Es schlägt auch fehl, wenn eine der folgenden Situationen eintritt:

- Es sind Fehler in der xa_open-Zeichenfolge vorhanden.
- Es sind nicht genügend Informationen zum Starten eines MQI-Kanals vorhanden.
- Beim Versuch, einen MQI-Kanal zu starten, ist ein Fehler aufgetreten (z. B. der Server-WS-Manager nicht aktiv).

### *Wiederherstellung nach einem Fehler in der erweiterten transaktionsorientierten Verarbeitung*

Nach einem Fehler muss ein Transaktionsmanager in der Lage sein, alle unvollständigen Arbeitseinheiten wiederherzustellen. Dazu muss der Transaktionsmanager in der Lage sein, als Ressourcenmanager einen beliebigen Warteschlangenmanager zu öffnen, der zum Zeitpunkt des Ausfalls an einer unvollständigen UO- Unit beteiligt war.

Daher müssen Sie sicherstellen, dass alle unvollständigen Arbeitseinheiten aufgelöst wurden, bevor Sie Änderungen an den Konfigurationsinformationen vornehmen.

Alternativ müssen Sie sicherstellen, dass die Konfigurationsänderungen die Fähigkeit des Transaktionsmanagers nicht beeinträchtigen, die Warteschlangenmanager zu öffnen, die er öffnen muss. Im Folgenden sind Beispiele für solche Konfigurationsänderungen zu finden:

- Inhalt einer xa_open-Zeichenfolge ändern
- Den Wert der Umgebungsvariablen MQSERVER ändern
- Einträge in der Definitionstabelle für den Clientkanal ändern (CCDT)
- Kanaldefinition für Serververbindung löschen

## *Die XA-Switchstrukturen*

Es werden zwei XA-Switchstrukturen mit dem erweiterten transaktionsorientierten Client auf jeder Plattform bereitgestellt.

Diese Switchstrukturen sind:

**MQRMIXASwitch**
Diese Switchstruktur wird von einem Transaktionsmanager verwendet, wenn ein Warteschlangenmanager, der als Ressourcenmanager fungiert, keine dynamische Registrierung verwendet.

**MQRMIXASwitchDynamic**
Diese Switchstruktur wird von einem Transaktionsmanager verwendet, wenn ein Warteschlangenmanager, der als Ressourcenmanager fungiert, die dynamische Registrierung verwendet.

Diese Switchstrukturen befinden sich in den in Tabelle 4 auf Seite 27 dargestellten Bibliotheken.

*Tabelle 4. IBM MQ-Bibliotheken, die die XA-Switchstrukturen enthalten*

| **Plattform** | **Bibliothek mit den XA-Switchstrukturen** |
|---|---|
| AIX AIX<br>Linux Linux | *MQ_INSTALLATION_PATH*/lib/libmqcxa |
| Systeme mit Windows Windows | *MQ_INSTALLATION_PATH*\bin\mqcxa.dll [1] |

*MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Der Name des IBM MQ-Ressourcenmanagers in jeder Switchstruktur ist MQSeries_XA_RMI, aber viele Warteschlangenmanager können dieselbe Switchstruktur gemeinsam nutzen.

**Zugehörige Konzepte**

„Dynamische Registrierung und erweiterte transaktionsorientierte Verarbeitung” auf Seite 27
Die Verwendung der dynamischen Registrierung ist eine Form der Optimierung, da sie die Anzahl der vom Transaktionsmanager ausgegebenen xa-Funktionsaufrufe verringern kann.

*Dynamische Registrierung und erweiterte transaktionsorientierte Verarbeitung*
Die Verwendung der dynamischen Registrierung ist eine Form der Optimierung, da sie die Anzahl der vom Transaktionsmanager ausgegebenen xa-Funktionsaufrufe verringern kann.

Wenn ein Warteschlangenmanager keine dynamische Registrierung verwendet, bezieht ein Transaktionsmanager den Warteschlangenmanager in jede UOWUOW.Einheit ein. Der Transaktionsmanager führt dies durch Aufrufen von xa_start, xa_end und xa_prepare aus, selbst wenn der Warteschlangenmanager keine Ressourcen hat, die innerhalb der UOI aktualisiert werden.

Wenn ein Warteschlangenmanager die dynamische Registrierung verwendet, beginnt ein Transaktionsmanager mit der Annahme, dass der Warteschlangenmanager nicht an einer Arbeitseinheit beteiligt ist, und ruft nicht xa_start auf. Der WS-Manager wird dann nur dann in die Arbeitseinheit einbezogen, wenn seine Ressourcen innerhalb der Synchronisationspunktsteuerung aktualisiert werden. Wenn dies der Fall ist,

ruft der erweiterte transaktionsorientierte Client ax_reg auf, um die Beteiligung des WS-Managers zu registrieren.

### *Extended Transactional Client mit TLS-Kanälen verwenden*

Sie können keinen TLS-Kanal mit der Zeichenfolge 'xa_open' konfigurieren. Führen Sie die folgenden Anweisungen aus, um die Definitionstabelle für den Clientkanal (ccdt) zu verwenden.

## Informationen zu diesem Vorgang

Aufgrund der begrenzten Größe der xa_open-Zeichenfolge 'xa_info' ist es nicht möglich, alle Informationen, die für die Einrichtung eines TLS-Kanals erforderlich sind, mit Hilfe der Methode 'xa_open', die eine Verbindung zu einem Warteschlangenmanager herstellen soll, zu übergeben. Daher müssen Sie entweder die Definitionstabelle für den Clientkanal verwenden oder, falls Ihr Transaktionsmanager zulässt, den Kanal mit MQCONNX erstellen, bevor Sie den Aufruf 'xa_open' absetzen.

Führen Sie die folgenden Schritte aus, um die Definitionstabelle für den Clientkanal zu verwenden:

## Vorgehensweise

1. Geben Sie eine xa_open-Zeichenfolge an, die nur den obligatorischen Parameter qmname (WS-Manager-Name) enthält, z. B. `XA_Open_String=qmname=MYQM`.
2. Verwenden Sie einen Warteschlangenmanager, um einen CLNTCONN-Kanal (Client-Connection-Kanal) mit den erforderlichen TLS-Parametern zu definieren. Geben Sie den Namen des Warteschlangenmanagers in das Attribut QMNAME in der CLNTCONN-Definition ein. Dieser Wert wird mit dem Namen des Befehls qmname in der Zeichenfolge xa_open abgeglichen.
3. Stellen Sie die CLNTCONN-Definition dem Clientsystem in einer Clientkanaldefinitionstabelle (CCDT) oder, unter Windows, im Active Directory zur Verfügung.
4. Wenn Sie eine CCDT verwenden, identifizieren Sie die CCDT, die die Definition des Kanals CLNTCONN enthält, indem Sie die Umgebungsvariablen MQCHLLIB und MQCHLTAB verwenden. Legen Sie diese Variablen in den Umgebungen fest, die sowohl von der Clientanwendung als auch vom Transaktionsmanager verwendet werden.

## Ergebnisse

Dadurch erhält der Transaktionsmanager eine Kanaldefinition für den entsprechenden Warteschlangenmanager mit den TLS-Attributen, die zur korrekten Authentifizierung erforderlich sind, einschließlich SSLCIPH, der CipherSpec.

### *Konfigurieren eines erweiterten transaktionsorientierten Clients für CICS*

Sie konfigurieren einen erweiterten transaktionsorientierten Client für die Verwendung durch CICS, indem Sie eine XAD-Ressourcendefinition zu einer CICS-Region hinzufügen.

Fügen Sie die XAD-Ressourcendefinition mit dem RDO-Befehl ( CICS resource definition online) **`cicsadd`**hinzu. Die XAD-Ressourcendefinition gibt die folgenden Informationen an:

- Eine xa_open-Zeichenfolge
- Der vollständig qualifizierte Pfadname einer Switchloaddatei.

Auf jeder der nachfolgenden Plattformen wird jeweils eine Switchloaddatei für die Verwendung durch CICS bereitgestellt:

- 

AIX
- 

Windows

Jede Switchloaddatei enthält eine Funktion, die einen Zeiger auf die XA-Switchstruktur zurückgibt, die für die dynamische Registrierung verwendet wird, MQRMIXASwitchDynamic. Den vollständig qualifizierten Pfadnamen der einzelnen Switchloaddateien finden Sie unter Tabelle 5 auf Seite 29.

| *Tabelle 5. Die Switchloaddateien* | |
|---|---|
| **Plattform** | **Switchloaddatei** |
| AIX AIX<br>Linux Linux | *MQ_INSTALLATION_PATH*/lib/amqczsc |
| Windows | *MQ_INSTALLATION_PATH*\bin\mqcc4swi.dll [1] |

*MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Im Folgenden finden Sie ein Beispiel für eine XAD-Ressourcendefinition für Windows-Systeme:

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

Weitere Informationen zum Hinzufügen einer XAD-Ressourcendefinition zu einer CICS-Region finden Sie im Handbuch *CICS Administration Reference* und im Handbuch *CICS Administration Guide* für Ihre Plattform.

Beachten Sie die folgenden Informationen zur Verwendung von CICS mit einem erweiterten transaktionsorientierten Client:

- Sie können nur eine XAD-Ressourcendefinition für IBM MQ zu einer CICS-Region hinzufügen. Dies bedeutet, dass nur ein Warteschlangenmanager einer Region zugeordnet werden kann und dass alle CICS-Anwendungen, die in der Region ausgeführt werden, nur eine Verbindung zu diesem Warteschlangenmanager herstellen können. Wenn Sie CICS-Anwendungen ausführen möchten, die eine Verbindung zu einem anderen WS-Manager herstellen, müssen Sie die Anwendungen in einer anderen Region ausführen.
- Jeder Anwendungsserver in einer Region ruft 'xa_open' auf, während er initialisiert wird, und startet einen MQI-Kanal zu dem Warteschlangenmanager, der der Region zugeordnet ist. Dies bedeutet, dass der Warteschlangenmanager gestartet werden muss, bevor ein Anwendungsserver gestartet wird, da andernfalls der Aufruf 'xa_open' fehlschlägt. Alle IBM MQ MQI client-Anwendungen, die später vom Anwendungsserver verarbeitet wurden, verwenden denselben MQI-Kanal.
- Wenn ein MQI-Kanal gestartet wird und kein Sicherheitsexit auf der Clientseite des Kanals vorhanden ist, ist die Benutzer-ID, die vom Clientsystem an die Serververbindung MCA fließt, `cics` . Unter bestimmten Umständen verwendet der WS-Manager diese Benutzer-ID für Berechtigungsprüfungen, wenn die Serververbindung MCA anschließend versucht, im Namen einer Clientanwendung auf die Ressourcen des Warteschlangenmanagers zuzugreifen. Wenn diese Benutzer-ID für Berechtigungsprüfungen verwendet wird, müssen Sie sicherstellen, dass sie über die Berechtigung zum Zugriff auf alle Ressourcen verfügt, auf die sie zugreifen müssen.

  Informationen darüber, wann der Warteschlangenmanager diese Benutzer-ID für Berechtigungsprüfungen verwendet, finden Sie unter Securing .
- Die CICS-Taskbeendigungsexits, die für die Verwendung auf IBM MQ-Clientsystemen bereitgestellt werden, werden in Tabelle 6 auf Seite 30 aufgelistet. Sie konfigurieren diese Exits auf die gleiche Weise, wie Sie die entsprechenden Exits für IBM MQ-Serversysteme konfigurieren. Informationen zu diesen Informationen finden Sie in der Veröffentlichung CICS-Benutzerexits aktivieren.

| Tabelle 6. CICS-Taskbeendigungsexits | | |
|---|---|---|
| **Plattform** | **Quelle** | **Bibliothek** |
| AIX AIX<br>Linux Linux | amqzscgx.c | amqczscg |
| Systeme mit Windows Windows | amqzscgn.c | mqcc1415.dll |

### *Konfigurieren eines erweiterten transaktionsorientierten Clients für Tuxedo*

Aktualisieren Sie die UBBCONFIG-Datei und die Ressourcenmanagertabelle, um die XAD-Ressourcendefinition für die Verwendung durch Tuxedo zu konfigurieren.

Führen Sie die folgenden Aktionen aus, um die XAD-Ressourcendefinition für die Verwendung durch Tuxedo zu konfigurieren:

- Verwenden Sie im Abschnitt "GROUPS" der Tuxedo-Datei "UBBCONFIG" für eine Anwendung den Parameter **OPENINFO** , um eine xa_open-Zeichenfolge anzugeben. Ein Beispiel hierfür finden Sie in der Beispieldatei UBBCONFIG, die für die Verwendung mit den Tuxedo-Beispielprogrammen bereitgestellt wird.

  AIX Auf den folgenden Plattformen lautet der Name der Datei `ubbstxcx.cfg`:

  – AIX

  Windows Windows, lautet der Name der Datei `ubbstxcn.cfg`.
- Geben Sie im Eintrag für einen WS-Manager in der Tuxedo-Ressourcenmanagertabelle den Namen einer XA-Switch-Struktur und den vollständig qualifizierten Pfadnamen der Bibliothek an, die die Struktur enthält:

  – AIX On AIX, geben Sie `udataobj/RM` an.

  – Windows Geben Sie unter Windows `udataobj\rm` an.

  Ein Beispiel für die Vorgehensweise bei den einzelnen Plattformen finden Sie unter TUXEDO-Beispiele . Tuxedo unterstützt die dynamische Registrierung eines Ressourcenmanagers, so dass Sie entweder MQRMIXASwitch oder MQRMIXASwitchDynamic verwenden können.

## Windows Microsoft -Transaktionsserver

Es ist keine zusätzliche Konfiguration erforderlich, bevor Sie Microsoft Transaction Server (MTS) als Transaktionsmanager verwenden können. Es gibt jedoch einige Punkte zu beachten.

Beachten Sie die folgenden Informationen zur Verwendung von MTS mit dem erweiterten transaktionsorientierten Client:

- Eine MTS-Anwendung startet immer einen MQI-Kanal, wenn er eine Verbindung zu einem Server-WS-Manager herstellt. MTS verwendet in seiner Rolle als Transaktionsmanager dann denselben MQI-Kanal, um mit dem Warteschlangenmanager zu kommunizieren.
- Nach einem Fehler muss MTS in der Lage sein, alle unvollständigen Arbeitseinheiten wiederherzustellen. Dazu muss MTS in der Lage sein, mit jedem WS-Manager zu kommunizieren, der zum Zeitpunkt des Ausfalls an einer unvollständigen Arbeitseinheit beteiligt war.

  Wenn eine MTS-Anwendung eine Verbindung zu einem Server-WS-Manager herstellt und einen MQI-Kanal startet, extrahiert der erweiterte transaktionsorientierte Client genügend Informationen aus den Parametern des MQCONN-oder MQCONNX-Aufrufs, damit der Kanal nach einem Fehler erneut gestartet

werden kann, falls dies erforderlich ist. Der erweiterte transaktionsorientierte Client übergibt die Informationen an MTS, und MTS zeichnet die Informationen in seinem Protokoll auf.

Wenn die MTS-Anwendung einen MQCONN-Aufruf ausgibt, ist diese Information lediglich der Name des Warteschlangenmanagers. Wenn die MTS-Anwendung einen MQCONNX-Aufruf absetzt und eine Kanaldefinitionsstruktur (MQCD) bereitstellt, enthält die Information auch den Namen des MQI-Kanals, die Netzadresse des Server-WS-Managers und das Kommunikationsprotokoll für den Kanal.

In einer Wiederherstellungssituation übergibt MTS diese Informationen an den erweiterten transaktionsorientierten Client zurück, und der erweiterte transaktionsorientierte Client verwendet ihn zum erneuten Starten des MQI-Kanals.

Wenn Sie jemals die Konfigurationsdaten ändern müssen, stellen Sie daher sicher, dass alle unvollständigen Arbeitseinheiten aufgelöst wurden, bevor Sie die Änderungen vornehmen. Stellen Sie alternativ sicher, dass die Konfigurationsänderungen die Fähigkeit des erweiterten transaktionsorientierten Clients nicht beeinträchtigen, einen MQI-Kanal unter Verwendung der von MTS erfassten Informationen erneut zu starten. Im Folgenden sind Beispiele für solche Konfigurationsänderungen zu finden:

- Den Wert der Umgebungsvariablen MQSERVER ändern
- Einträge in der Definitionstabelle für den Clientkanal ändern (CCDT)
- Kanaldefinition für Serververbindung löschen

- Beachten Sie die folgenden Bedingungen, wenn Sie einen erweiterten transaktionsorientierten Client mit MTS verwenden:
  - In einem einzelnen Thread kann eine Clientanwendung immer nur mit einem Warteschlangenmanager verbunden sein.
  - Jeder Thread einer Clientanwendung kann eine Verbindung zu einem anderen WS-Manager herstellen.
  - Eine Clientanwendung kann keine gemeinsam genutzten Verbindungskennungen verwenden.

## Definieren von MQI-Kanälen

Um einen neuen Kanal zu erstellen, müssen Sie **zwei** Kanaldefinitionen erstellen, eine für jedes Ende der Verbindung, die denselben Channel-Namen und kompatible Kanaltypen verwenden. In diesem Fall sind die Kanaltypen *Serververbindung* und *Clientverbindung*.

### Benutzerdefinierte Kanäle

Wenn der Server keine Kanäle automatisch definiert, gibt es zwei Möglichkeiten, die Kanaldefinitionen zu erstellen und der IBM MQ-Anwendung auf dem IBM MQ MQI client-System Zugriff auf den Kanal zu erteilen.

Diese beiden Methoden werden im Detail beschrieben:

1. Erstellen Sie eine Kanaldefinition auf dem IBM MQ-Client und die andere auf dem Server.

   Dies gilt für jede Kombination aus IBM MQ MQI client und Serverplattformen. Verwenden Sie es, wenn Sie auf dem System gestartet werden, oder um Ihre Konfiguration zu testen.

   Ausführliche Informationen zu dieser Vorgehensweise finden Sie im Abschnitt „Serververbindungs- und Clientverbindungsdefinitionen auf verschiedenen Plattformen erstellen” auf Seite 37.

2. Erstellen Sie beide Kanaldefinitionen auf der Servermaschine.

   Verwenden Sie diese Methode, wenn Sie mehrere Kanäle und IBM MQ MQI client-Maschinen gleichzeitig einrichten.

   Ausführliche Informationen zu dieser Vorgehensweise finden Sie im Abschnitt „Serververbindungs- und Clientverbindungsdefinitionen auf dem Server erstellen” auf Seite 43.

## Automatisch definierte Kanäle

IBM MQ-Produkte auf anderen Plattformen als z/OS umfassen ein Feature, das automatisch eine Kanaldefinition auf dem Server erstellen kann, wenn es nicht vorhanden ist.

Wenn eine eingehende Verbindungsanforderung von einem Client empfangen wird und eine entsprechende Serververbindungsdefinition auf diesem Warteschlangenmanager nicht gefunden werden kann, erstellt IBM MQ eine Definition automatisch und fügt sie dem Warteschlangenmanager hinzu. Die automatische Definition basiert auf der Definition des Standard-Serververbindungskanals SYSTEM.AUTO.SVRCONN. Sie aktivieren die automatische Definition von Serververbindungsdefinitionen, indem Sie das WS-Manager-Objekt mit dem Befehl ALTER QMGR mit dem Parameter CHAD (oder mit dem PCF-Befehl Change Queue Manager mit dem Parameter ChannelAutoDef ändern) aktualisieren.

**Zugehörige Konzepte**

„Kanalsteuerfunktion" auf Seite 239
Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

# ALW AMQP-Kanäle erstellen und verwenden

Wenn Sie die IBM MQ-Unterstützung für die AMQP-Servicekomponente in Ihrer IBM MQ -Installation installieren, können Sie IBM MQ MQSC-Befehle (**runmqsc**) ausführen, um einen Kanal zu definieren, zu ändern, zu löschen, zu starten und zu stoppen. Auch den Status eines Kanals können Sie anzeigen.

## Vorbereitende Schritte

Bei dieser Task wird davon ausgegangen, dass Sie den AMQP-Kanal installiert haben. Hierfür wählen Sie bei der Installation von IBM MQ die AMQP-Servicekomponente aus. Für weitere Informationen folgen Sie dem Link für Ihre Plattform und suchen Sie dann die Tabellenzeile für "AMQP Service":

- AIX IBM MQ-Komponenten für AIX-Systeme
- Linux IBM MQ-RPM-Komponenten für Linux-Systeme
- Linux IBM MQ-Debian-Komponenten für Linux-Ubuntu-Systeme
- Windows IBM MQ-Funktionen für Windows-Systeme

**Anmerkung:** Ein Beispiel für eine SERVICE-Komponente und weitere Informationen finden Sie im Abschnitt IBM MQ -Service für AMQP erneut starten , wenn Ihr AMQP-Service nicht mehr ordnungsgemäß funktioniert.

Bei dieser Task wird auch davon ausgegangen, dass ein Warteschlangenmanager vorhanden ist.

Um eine Testverbindung zum Warteschlangenmanager herzustellen, können Sie alle AMQP-Clients verwenden, die das Protokoll OASIS AMQP 1.0 implementieren, wie z. B. MQ Light- und Apache Qpid-Clients wie Apache Qpid Proton und Apache Qpid JMS.

V 9.3.0 Ab IBM MQ 9.3.0können Sie nur den Standardkanal SYSTEM.DEF.AMQPzum Testen von MQ Light -Verbindungen zum Warteschlangenmanager. Bei dem folgenden Verfahren wird der Standardkanal verwendet.

Der hier beschriebene Vorgang basiert auf dem MQ Light-Node.js-Client. Die Schritte, die sich auf den IBM MQ-Warteschlangenmanager beziehen, sind jedoch für alle Clients gleich.

**Anmerkung:** AMQP-Kanäle unterstützen keine benutzerdefinierten AMQP-Services. AMQP-Kanäle unterstützen nur den Systemstandardservice SYSTEM.AMQP.SERVICE. Sie können nur eine Instanz dieses Service für jeden Warteschlangenmanager definieren.

## Vorgehensweise

1. Starten Sie **runmqsc** im Verzeichnis *mqinstall*/bin/:

```
runmqsc QMNAME
```

2. (Nur erforderlich, wenn Ihr Warteschlangenmanager IBM MQ 9.0.4 oder früher angehört.) Überprüfen Sie, ob die AMQP-Funktion ordnungsgemäß installiert ist und korrekt funktioniert.

   Verwenden Sie den Befehl **START SERVICE** , um den IBM MQ -Service zu starten, der die JVM steuert:

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

   **Anmerkung:** Ab IBM MQ 9.1 ist das Attribut **CONTROL** von SYSTEM.AMQP.SERVICE auf *Warteschlangenmanager*gesetzt. Dadurch wird der Service beim Start des Warteschlangenmanagers automatisch gestartet. Wenn Sie das Attribut **CONTROL** auf *MANUELL* setzen, können Sie verhindern, dass der Service beim Start des Warteschlangenmanagers gestartet wird.

   Beim Start des Warteschlangenmanagers werden der AMQP-Service und der AMQP-Kanal, falls definiert, automatisch gestartet.

3. Legen Sie die MCAUSER-Benutzer-ID fest.

   Wenn ein AMQP-Client eine Verbindung zu einem Kanal herstellt, gibt der Kanal eine MCAUSER-Benutzer-ID an, die für Verbindungen zum Warteschlangenmanager verwendet wird. Standardmäßig ist für MCAUSER kein Wert angegeben. AMQP-Clients können erst eine Verbindung zum Warteschlangenmanager herstellen, wenn Sie einen Wert für MCAUSER angegeben haben, bei dem es sich um einen gültigen IBM MQ-Benutzer handelt, der zum Veröffentlichen und Subskribieren von IBM MQ-Topics berechtigt ist.

   **Anmerkung:** Windows Unter Windows wird die Festlegung der MCAUSER-Benutzer-ID vor IBM MQ 9.2.0 nur für Benutzer-IDs mit einer Länge von bis zu 12 Zeichen unterstützt. Ab IBM MQ 9.2.0 ist die Begrenzung auf 12 Zeichen aufgehoben.

   a) Verwenden Sie den Befehl **ALTER CHANNEL** , um die Benutzer-ID MCAUSER festzulegen:

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

   b) Verwenden Sie die beiden folgenden **setmqaut**-Befehle, um Ihre MCAUSER-Benutzer-ID zum Veröffentlichen und Subskribieren von Topics zu berechtigen:

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

   und

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

   Wenn der Kanal aktiv ist, während die MCAUSER-Benutzer-ID hinzugefügt oder geändert wird, müssen Sie den Kanal stoppen und erneut starten.

   **Anmerkung:** Wenn die MCAUSER-Benutzer-ID nicht festgelegt ist oder die MCAUSER-Benutzer-ID nicht zum Veröffentlichen oder Subskribieren von IBM MQ-Topics berechtigt ist, wird im AMQP-Client eine Fehlernachricht angezeigt.

4. Verwenden Sie den Befehl **START CHANNEL** , um das SYSTEM.DEF.AMQP -Kanal:

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

5. Wenn Sie den Kanalstatus überprüfen möchten, verwenden Sie den Befehl **DISPLAY CHSTATUS** :

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

Wenn der Kanal ordnungsgemäß ausgeführt wird, wird STATUS(RUNNING) in der Befehlsausgabe angezeigt.

6. Ändern Sie den Standardport.

   Der Standardport für AMQP 1.0-Verbindungen ist 5672. Wenn Sie Port 5672 bereits verwenden, was möglich ist, wenn Sie MQ Light zuvor installiert haben, müssen Sie den Port ändern, den Ihr AMQP-Kanal verwendet. Verwenden Sie den Befehl **ALTER CHANNEL**, um den Port zu ändern:

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

7. Wenn Sie keine Verbindungen zum AMQP-Kanal mithilfe von Kanalauthentifizierungsregeln (CHLAUTH) blockieren oder filtern möchten, inaktivieren Sie die Kanalauthentifizierung auf dem Warteschlangenmanager wie folgt:

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

   Es wird nicht empfohlen, die Verbindungsauthentifizierung auf einem Produktionswarteschlangenmanager zu inaktivieren. Sie sollten die Verbindungsauthentifizierung nur in einer Entwicklungsumgebung inaktivieren.

   Alternativ können Sie die Kanalauthentifizierungsregeln des Warteschlangenmanagers so konfigurieren, dass bestimmte Verbindungen zum AMQP-Kanal zulässig sind.

8. Optional: Wenn Sie die SSL/TLS-Verschlüsselung auf dem Kanal aktivieren möchten, müssen Sie für das Attribut SSLCIPH für den Kanal unter Verwendung des konfigurierten Schlüsselrepositorys für den Warteschlangenmanager eine geeignete Verschlüsselungsspezifikation festlegen. Standardmäßig ist keine Verschlüsselungsspezifikation angegeben, auf dem Kanal wird also keine SSL/TLS-Verschlüsselung verwendet. Verwenden Sie den Befehl **ALTER CHANNEL**, um eine Verschlüsselungsspezifikation festzulegen. For example:

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

   Zusätzlich gibt es in Zusammenhang mit der SSL/TLS-Verschlüsselung eine Reihe weiterer Kanalkonfigurationsoptionen, die Sie wie folgt festlegen können:

   - Standardmäßig ist das Zertifikat im Schlüsselrepository des Warteschlangenmanagers mit der Bezeichnung, die dem Attribut **CERTLABL** des Warteschlangenmanagers entspricht, der Name, der von der SSL/TLS-Verschlüsselung des Kanals verwendet wird. Sie können ein anderes Zertifikat auswählen, indem Sie **CERTLABL** festlegen. Verwenden Sie den Befehl **ALTER CHANNEL**, um die Bezeichnung für das erforderliche Zertifikat anzugeben:

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

   - Sie können den Kanal so einstellen, dass von SSL/TLS-Clientverbindungen ein Zertifikat verlangt wird. Sie können auswählen, ob ein Zertifikat aus einer SSL/TLS-Clientverbindung erforderlich ist, indem Sie das Attribut **SSLCAUTH** festlegen. Verwenden Sie den Befehl **ALTER CHANNEL**, um festzulegen, ob von einer SSL/TLS-Clientverbindung ein Zertifikat benötigt wird. For example:

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

   - Wenn Sie das Attribut **SSLCAUTH** auf ERFORDERLICH setzen, kann der definierte Name des Zertifikats vom Client überprüft werden. Legen Sie das Attribut **SSLPEER** fest, um den definierten Namen des Zertifikats vom Client zu überprüfen. Verwenden Sie den Befehl **ALTER CHANNEL** , um den definierten Namen des Zertifikats vom Client zu überprüfen. For example:

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

     Alternativ können Sie Kanalauthentifizierungsdatensätze auch verwenden, um Verbindungen zuzulassen oder zu blockieren, weil diese Methode eine größere Granularität bietet als das Attribut

**SSLPEER**. Weitere Informationen zum Festlegen von **SSLPEER** und zur Verwendung von Kanalauthentifizierungsdatensätzen als Alternative finden Sie in SSL-Peer.

9. Führen Sie den folgenden Befehl aus, um den MQ Light-Node.js-Client zu installieren:

```
npm install mqlight
```

10. Navigieren Sie zum Verzeichnis `node_modules/mqlight/samples`, und führen Sie die Beispielempfängeranwendung aus:
    - Wenn Sie die Standardportnummer verwenden, können Sie die Beispielempfängeranwendung ausführen:

      ```
      node recv.js
      ```

    - Wenn Sie Ihren AMQP-Kanal für die Verwendung einer anderen Portnummer konfiguriert haben, können Sie die Beispielempfängeranwendung mit einem Parameter zur Angabe der neuen Portnummer ausführen:

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

    Bei erfolgreicher Verbindung zum Standardkanal wird folgende Nachricht angezeigt:

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

    Die Anwendung ist jetzt mit dem Warteschlangenmanager verbunden und wartet auf den Empfang von Nachrichten. Sie ist für das Thema `public` abonniert.

    **Anmerkung:** Die `client-id` wird automatisch generiert, sofern Sie sie nicht mit dem Parameter `-i` angegeben haben.

11. Navigieren Sie in einem neuen Befehlsfenster zum Verzeichnis `node_modules/mqlight/samples`, und führen Sie die Beispielsenderanwendung aus, indem Sie den folgenden Befehl ausführen:

    ```
    node send.js
    ```

    Im Befehlsfenster für die Empfängeranwendung wird die Nachricht `Hello World` angezeigt.

12. Verwenden Sie das **AMQSSUB** IBM MQ-Beispiel, um eine MQ Light-Beispielnachricht zu empfangen.

    Unter Linux und Windows kann die Stichprobe an den folgenden Positionen gefunden werden:

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

    a) Führen Sie das Beispiel mit folgendem Befehl aus:

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

    b) Senden Sie eine Nachricht an die IBM MQ-Anwendung, indem Sie den folgenden Befehl erneut ausführen:

    ```
    node send.js
    ```

13. Verwenden Sie den Befehl **DEFINE CHANNEL**, um weitere AMQP-Kanäle zu erstellen:

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

Wenn Sie einen Kanal definieren, muss dieser mit dem Befehl **START CHANNEL** manuell gestartet werden:

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

Um zu prüfen, ob der Kanal ordnungsgemäß funktioniert, können Sie die Beispielempfängeranwendung ausführen und dabei den Port des neuen Kanals angeben:

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

## Nächste Schritte

Sie können die folgenden Befehle verwenden, um die IBM MQ-Verbindungen anzuzeigen, den Kanal zu stoppen und den Kanal zu löschen:

**DISPLAY CONN(*) TYPE(*CONN*) WHERE (*CHANNEL EQ SYSTEM.DEF.AMQP*)**
Zeigt die IBM MQ-Verbindung an, die der AMQP-Kanal auf dem Warteschlangenmanager hergestellt hat.

**DISPLAY CHSTATUS(*) CHLTYPE(*AMQP*) CLIENTID(*) ALL**
Zeigt eine Liste der AMQP-Clients an, die mit dem angegebenen Kanal verbunden sind.

**STOP CHANNEL (*MY.AMQP.CHANNEL*)**
Stoppt einen AMQP-Kanal und schließt den Port, an dem er empfangsbereit ist.

**DELETE CHANNEL (*MY.AMQP.CHANNEL*)**
Löscht alle Kanäle, die Sie erstellt haben.

**Anmerkung:** Löschen Sie den Standardkanal SYSTEM.DEF.AMQP nicht.

Sie können mithilfe von **runmqsc** oder PCF ermitteln, ob die AMQP-Funktionalität in der IBM MQ -Installation installiert ist und ob ihr ein Warteschlangenmanager zugeordnet ist:

- Zeigen Sie mithilfe von **runmqsc** die Attribute des Warteschlangenmanagers an und prüfen Sie, ob `AMQPCAP(YES)` angegeben ist.
- Verwenden Sie bei Verwendung von PCF den Befehl **MQCMD_INQUIRE_Q_MGR** und prüfen Sie den Wert von `MQIA_AMQP_CAPABILITY`.

**Zugehörige Tasks**
AMQP-Clientanwendungen entwickeln
AMQP-Clients schützen
**Zugehörige Verweise**
strmqm

# ALW AMQP-Kanal aus Warteschlangenmanagern entfernen

Sie können den AMQP-Kanal aus Warteschlangenmanagern entfernen, indem Sie die entsprechenden Ordner aus dem Installationsverzeichnis entfernen.

## Vorgehensweise

1. Stoppen Sie den Warteschlangenmanager.
2. Entfernen Sie die IBM MQ-Unterstützung für die APIs der AMQP-Servicekomponente:
   - AIX Führen Sie unter AIX folgenden Befehl aus:

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

- Linux Entfernen Sie unter Linux den AMQP-RPM. Wenn Sie den RPM vor der Installation erneut gepackt haben, geben Sie den Namen des neu gepackten RPM an.

```
rpm -e MQSeriesAMQP
```

- Windows Entfernen Sie unter Windows den Ordner amqp aus der IBM MQ-Installation. Achten Sie darauf, dass keine anderen Dateien oder Ordner aus dem IBM MQ-Installationspfad entfernt werden.

3. Starten Sie den Warteschlangenmanager erneut.

**Zugehörige Tasks**

AMQP-Clientanwendungen entwickeln

AMQP-Clients schützen

### ALW Protokolldateien für AMQP-Kanäle

Die Protokolldateien für AMQP-Kanäle werden in demselben IBM MQ-Datenverzeichnis wie die IBM MQ-Protokolldateien gespeichert.

Das Standarddatenverzeichnis unter Windows ist `C:\ProgramData\IBM\MQ`.

Das Standarddatenverzeichnis unter Linux ist `/var/mqm`.

Der AMQP-Kanal schreibt Protokollinformationen in die folgenden Protokolldateien, die sich im IBM MQ-Datenverzeichnis befinden:

- `amqp.stdout`, in den Ordner `qmgrs/`*`QM-name`* geschrieben.
- `amqp.stderr`, in den Ordner `qmgrs/`*`QM-name`* geschrieben.
- `amqp_*.log` , in den Ordner `qmgrs/`*`QM-name`*`/errors` geschrieben.

Wenn ein MQ Light-Client einen Authentifizierungs- oder Berechtigungsfehler empfängt, kann Ihr Administrator detaillierte Informationen zu dem Grund für den Sicherheitsfehler in der `amqp_0.log`-Datei und den `MQ AMQERR*.log`-Dateien finden.

Alle FDC-Dateien werden als `AMQP*.FDC`-Dateien erstellt, die in den Ordner *`data-directory`*`/errors` geschrieben werden.

Einige Konfigurationsdateien werden in das Verzeichnis `qmgrs/`*`QM-name`*`/amqp` geschrieben. Die Dateien in diesem Verzeichnis müssen Sie nicht bearbeiten.

**Zugehörige Konzepte**

Fehlerprotokolle unter AIX, Linux, and Windows

**Zugehörige Tasks**

AMQP-Clientanwendungen entwickeln

AMQP-Clients schützen

## Serververbindungs-und Clientverbindungsdefinitionen auf verschiedenen Plattformen erstellen

Sie können jede Kanaldefinition auf dem Computer erstellen, auf den sie angewendet wird. Es gibt jedoch Einschränkungen, wie Sie Kanaldefinitionen auf einem Client-Computer erstellen können.

### Informationen zu diesem Vorgang

Auf allen Plattformen können Sie IBM MQ Scriptbefehle (MQSC), PCF-Befehle (PCF = Programmable Command Format) oder den IBM MQ Explorer verwenden, um einen Serververbindungskanal auf der Servermaschine zu definieren.

z/OS Unter z/OS können Sie außerdem auch die Betriebs- und Steuerkonsolen verwenden.

IBM i Unter IBM i können Sie auch die Anzeigenschnittstelle verwenden.

Da MQSC-Befehle auf einer Maschine nicht verfügbar sind, auf der IBM MQ nur als IBM MQ MQI client installiert ist, müssen Sie zum Definieren eines Clientverbindungskanals auf der Clientmaschine andere Möglichkeiten nutzen.

Die folgenden Hinweise gelten für **runmqsc**:

- Sie können den Parameter **-c** und optional den Parameter **-u** angeben, um **runmqsc** als Client mit dem Warteschlangenmanager zu verbinden, den Sie verwalten wollen.
- Wenn Sie mit dem Parameter **-u** eine Benutzer-ID übergeben, werden Sie zur Eingabe eines passenden Kennworts aufgefordert.
- Wenn Sie den Datensatz `CONNAUTH AUTHINFO` mit `CHCKLOCL(REQUIRED)` oder `CHCKLOCL(REQ-DADM)` konfiguriert haben, müssen Sie den Parameter **-u** verwenden. Andernfalls können Sie Ihren Warteschlangenmanager nicht mit **runmqsc** verwalten.

### Prozedur

- Informationen zum Definieren eines Serververbindungskanals auf dem Server finden Sie unter „Definieren eines Serververbindungskanals auf dem Server" auf Seite 38.
- Informationen zum Erstellen eines Clientverbindungskanals auf einem IBM MQ MQI client unter Verwendung der Umgebungsvariable **MQSERVER** finden Sie unter „Clientverbindungskanal unter IBM MQ MQI client mit MQSERVER erstellen" auf Seite 39.
- Informationen zum Erstellen eines Clientverbindungskanals in einem IBM MQ MQI clientunter Verwendung der MQCNO-Struktur in einem MQCONNX-Aufruf finden Sie unter „Clientverbindungskanal auf dem IBM MQ MQI client mit MQCNO erstellen" auf Seite 43.

## Definieren eines Serververbindungskanals auf dem Server

Starten Sie bei Bedarf MQSC, und definieren Sie dann den Serververbindungskanal.

### Vorgehensweise

1. Optional: Wenn Ihre Serverplattform nicht z/OS ist, erstellen und starten Sie zuerst einen WS-Manager und starten Sie dann MQSC-Befehle.
   a) Erstellen Sie einen WS-Manager mit dem Namen QM1. Beispiel:

   ```
   crtmqm QM1
   ```

   b) Starten Sie den Warteschlangenmanager:

   ```
   strmqm QM1
   ```

   c) Starten Sie MQSC-Befehle:

   ```
   runmqsc QM1
   ```

2. Definieren Sie einen Kanal mit dem ausgewählten Namen und dem Kanaltyp *server-connection* .

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

   Diese Kanaldefinition ist dem Warteschlangenmanager zugeordnet, der auf dem Server ausgeführt wird.
3. Verwenden Sie den folgenden Befehl, um den Zugriff der Eingangsverbindung auf den Warteschlangenmanager zu ermöglichen:

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

- Dabei verwendet SET CHLAUTH den Namen des Kanals, der im vorherigen Schritt definiert wurde.
- Hierbei steht *'IP address'* für die IP-Adresse des Clients.
- Dabei ist *'userid'* die ID, die Sie dem Kanal für die Zugriffssteuerung für die Zielwarteschlangen bereitstellen möchten. Bei diesem Feld muss die Groß-/Kleinschreibung beachtet werden.

Sie können die eingehende Verbindung mithilfe einer Reihe unterschiedlicher Attribute identifizieren. Das Beispiel verwendet die IP-Adresse. Zu den Alternativattributen gehören die Clientbenutzer-ID und der TLS-Betreffname. Weitere Informationen finden Sie in Kanalauthentifizierungsdatensätze .

## Clientverbindungskanal unter IBM MQ MQI client mit MQSERVER erstellen

Sie können einen Clientverbindungskanal auf einer Client-Workstation mit der Umgebungsvariablen **MQSERVER** definieren.

### Informationen zu diesem Vorgang

Sie können die Umgebungsvariable **MQSERVER** verwenden, um eine einfache Definition eines Clientverbindungskanals anzugeben. Es ist einfach in dem Sinne, dass Sie mit dieser Methode nur einige Attribute des Kanals angeben können.

Wenn Sie die Umgebungsvariable **MQSERVER** verwenden, um den Kanal zwischen Ihrer IBM MQ MQI client -Maschine und einer Servermaschine zu definieren, ist dies der einzige Kanal, der für Ihre Anwendung verfügbar ist, und es wird nicht auf die Definitionstabelle für Clientkanäle (CCDT) verwiesen.

Wenn die MQCONN-oder MQCONNX-Anforderung einen anderen Warteschlangenmanager als den angibt, mit dem das Empfangsprogramm verbunden ist, oder wenn der **MQSERVER** -Parameter *Transport-Type* nicht erkannt wird, schlägt die MQCONN-oder MQCONNX-Anforderung mit dem Rückkehrcode MQRC_Q_MGR_NAME_ERRORfehl.

Linux AIX Unter AIX and Linuxkönnen Sie **MQSERVER** wie in einem der folgenden Beispiele definieren:

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

Alle MQCONN-oder MQCONNX-Anforderungen versuchen dann, den von Ihnen definierten Kanal zu verwenden, es sei denn, eine MQCD-Struktur wurde von der MQCNO-Struktur, die an MQCONNX übergeben wird, referenziert. In diesem Fall hat der von der MQCD-Struktur angegebene Kanal Vorrang vor allen von der Umgebungsvariablen **MQSERVER** angegebenen.

Die Umgebungsvariable **MQSERVER** hat Priorität vor allen Clientkanaldefinitionen, auf die die Umgebungsvariablen **MQCHLLIB** und **MQCHLTAB** verweisen.

### Prozedur

- Verwenden Sie je nach Plattform einen der folgenden Befehle, um die Kanaldefinition mit **MQSERVER**anzugeben.
  - Windows Geben Sie unter Windowseine einfache Kanaldefinition wie folgt an:

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

    For example:

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

– Linux AIX Geben Sie unter AIX and Linuxeine einfache Kanaldefinition wie folgt an:

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

For example:

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

– IBM i Geben Sie unter IBM ieine einfache Kanaldefinition wie folgt an:

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

For example:

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

**Anmerkungen:**

– Der *ChannelName* muss mit dem auf dem Server definierten Namen übereinstimmen. Er darf keinen Schrägstrich (/) enthalten, weil dieses Zeichen verwendet wird, um den Kanalnamen, den Transporttyp und den Verbindungsnamen zu trennen. Wenn die Umgebungsvariable **MQSERVER** zum Definieren eines Clientkanals verwendet wird, wird eine maximale Nachrichtenlänge (**MAXMSGL**) von 100 MB verwendet. Daher ist die maximale Nachrichtengröße, die für den Kanal wirksam ist, der im SVRCONN-Kanal auf dem Server angegebene Wert.

– Der Wert für *TransportType* kann abhängig von Ihrer IBM MQ -Clientplattform LU62, TCP, NETBI-0Soder SPXsein.

– Linux AIX Unter AIX and Linuxist *TransportType* von der Groß-/Kleinschreibung abhängig und muss in Großbuchstaben angegeben werden. Ein MQCONN -oder MQCONNX -Aufruf gibt 2058 zurück, wenn der Transporttyp nicht erkannt wird

– *ConnectionName* ist der Name des Servers, wie für das Kommunikationsprotokoll definiert (*TransportType*). Es muss sich um einen vollständig qualifizierten Netznamen handeln, z. B. AMACHI-NE.ACOMPANY.COM(1414).

– Der *ConnectionName* kann eine durch Kommas getrennte Liste von Verbindungsnamen sein. Die Verbindungsnamen in der Liste werden auf ähnliche Weise für mehrere Verbindungen in einer Clientverbindungstabelle verwendet. Die Liste der Verbindungsnamen kann als Alternative zu Warteschlangenmanagergruppen verwendet werden, um mehrere Verbindungen anzugeben, die der Client versuchen soll. Wenn Sie einen Warteschlangenmanager mit mehreren Instanzen konfigurieren, können Sie eine Verbindungsnamensliste verwenden, um verschiedene Warteschlangenmanagerinstanzen anzugeben.

- Geben Sie den folgenden Befehl ein, um **MQSERVER** abzubrechen und zur Definitionstabelle für den Clientkanal zurückzukehren, auf die **MQCHLLIB** und **MQCHLTAB**verweisen:

– Linux AIX Unter AIX and Linux:

```
unset MQSERVER
```

– Windows Unter Windows:

```
SET MQSERVER=
```

**Beispiel**

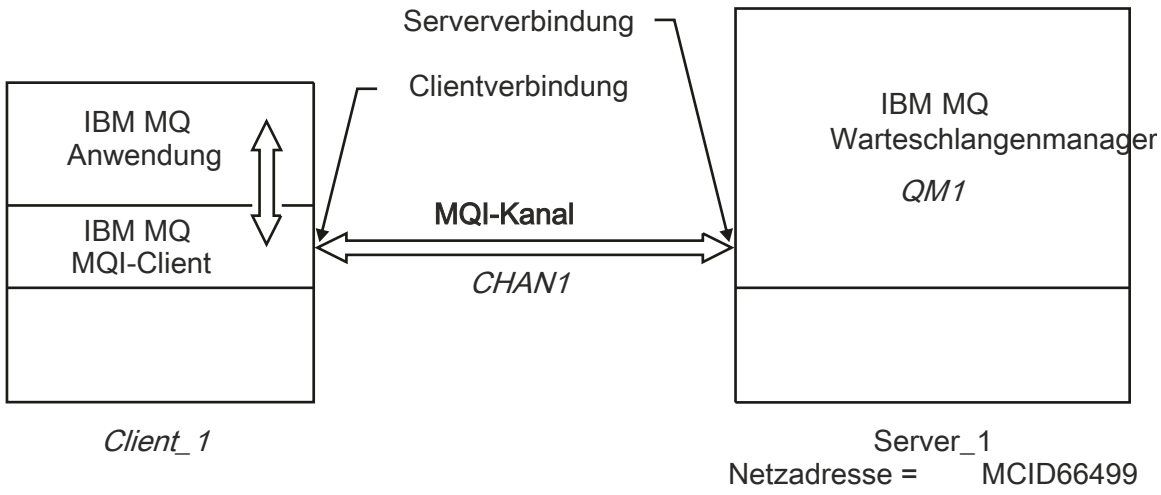



*Abbildung 1. Beispiel für eine einfache Kanaldefinition*

Verwenden Sie die folgenden Befehle, um die in Abbildung 1 auf Seite 41dargestellte einfache Kanaldefinition zu erstellen:

- Linux AIX Unter AIX and Linux:

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

- Windows Unter Windows:

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

**Anmerkung:** Informationen zum Ändern der TCP/IP-Portnummer finden Sie unter „TCP/IP-Standardport ändern” auf Seite 42.

Im Folgenden sind einige weitere Beispiele für einfache Kanaldefinitionen enthalten:

- Windows Unter Windows:

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

- Linux AIX Unter AIX and Linux:

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

  Hierbei steht B0X99 für den LU 6.2-Verbindungsnamen.

- IBM i Unter IBM i:

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

Auf dem IBM MQ MQI client versuchen alle **MQCONN**- oder **MQCONNX**-Anforderungen dann, den von Ihnen definierten Kanal zu verwenden, es sei denn, der Kanal wird in einer MQCD-Struktur überschrieben, die von der MQCNO-Struktur referenziert wird, die **MQCONNX** bereitgestellt wird.

**Zugehörige Tasks**

„Verwendung von IBM MQ-Umgebungsvariablen” auf Seite 67
Sie können Befehle verwenden, um die aktuellen Einstellungen anzuzeigen oder um die Werte von IBM MQ-Umgebungsvariablen zurückzusetzen.

„Clientverbindungskanal auf dem IBM MQ MQI client mit MQCNO erstellen” auf Seite 43
Sie können einen Clientverbindungskanal auf der Client-Workstation definieren, indem Sie die MQCNO-Struktur in einem MQCONNX-Aufruf verwenden.

### *TCP/IP-Standardport ändern*

Bei TCP/IP geht IBM MQ standardmäßig davon aus, dass der Kanal mit Port 1414 verbunden wird. Falls erforderlich, können Sie den Standardwert ändern.

## Informationen zu diesem Vorgang

Sie können die Portnummer mit einer der folgenden drei Optionen ändern:

- Durch Verwendung der Umgebungsvariablen **MQSERVER** .
- Durch Ändern der Datei `mqclient.ini` .
- Durch Hinzufügen von IBM MQ zur Servicedatei.

## Prozedur

- Um die Portnummer mithilfe der Umgebungsvariablen **MQSERVER** zu ändern, fügen Sie die Portnummer in eckigen Klammern als letzten Teil von *ConnectionName*hinzu. Beispiel:
  - Linux AIX Unter AIX and Linux:

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

  - Windows Unter Windows:

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

- Um die Portnummer mithilfe der Datei `mq.ini` zu ändern, fügen Sie die Portnummer zum Protokollnamen hinzu. Beispiel:

  ```
  TCP:
  port=2001
  ```

- Führen Sie die im Abschnitt „TCP/IP-Listener unter AIX and Linux verwenden” auf Seite 285beschriebenen Schritte aus, um die Portnummer durch Hinzufügen von IBM MQ zur Servicedatei zu ändern.

### Windows *SPX-Standardsocket ändern*

Für SPX geht IBM MQ standardmäßig davon aus, dass der Kanal mit dem Socket 5E86 verbunden wird. Falls erforderlich, können Sie den Standardwert ändern.

## Informationen zu diesem Vorgang

Sie können die Portnummer mit einer der folgenden Optionen ändern:

- Durch Verwendung der Umgebungsvariablen **MQSERVER** .

  Geben Sie bei SPX-Verbindungen den Verbindungsnamen und den Socket in der Form `network.node(socket)` an. Wenn sich der IBM MQ-Client und -Server in demselben Netz befinden, muss das Netz nicht angegeben werden. Wenn Sie den Standardsocket verwenden, muss der Socket nicht angegeben werden.
- Durch Ändern der SPX-Zeilengruppe der Datei `mqclient.ini` file.Changing der Datei `qm.ini` .

### Prozedur

- Wenn Sie die Portnummer für eine SPX-Verbindung mit der Umgebungsvariable **MQSERVER** ändern möchten, geben Sie den *ConnectionName* und das Socket im Format `network.node(socket)` an, wie im folgenden Beispiel gezeigt:

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

  **Anmerkung:** Wenn sich der IBM MQ -Client und -Server in demselben Netz befinden, müssen Sie das Netz nicht angeben. Wenn Sie den Standardsocket verwenden, müssen Sie den Socket nicht angeben.

- Um die Portnummer mithilfe der Datei `qm.ini` zu ändern, fügen Sie die Portnummer zum Protokollnamen hinzu. Beispiel:

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

## Clientverbindungskanal auf dem IBM MQ MQI client mit MQCNO erstellen

Sie können einen Clientverbindungskanal auf der Client-Workstation definieren, indem Sie die MQCNO-Struktur in einem MQCONNX-Aufruf verwenden.

### Informationen zu diesem Vorgang

Eine IBM MQ MQI client-Anwendung kann die Verbindungsoptionsstruktur (MQCNO) in einem **MQCONNX**-Aufruf verwenden, um auf eine Kanaldefinitionsstruktur (MQCD) zu verweisen, die die Definition eines Clientverbindungskanals enthält.

Auf diese Weise kann die Clientanwendung die **ChannelName**-, **TransportType**-und **ConnectionName** -Attribute eines Kanals während der Ausführung angeben, so dass die Clientanwendung gleichzeitig eine Verbindung zu mehreren Server-WS-Managern herstellen kann.

Beachten Sie Folgendes: Wenn Sie einen Kanal mit der Umgebungsvariablen **MQSERVER** definieren, können die Attribute **ChannelName**, **TransportType** und **ConnectionName** während der Ausführung nicht angegeben werden.

Eine Clientanwendung kann auch Attribute eines Kanals angeben, z. B. **MaxMsgLength** und **SecurityExit**. Wenn Sie solche Attribute angeben, kann die Clientanwendung Werte für die Attribute angeben, die nicht die Standardwerte sind, und ermöglicht es, Kanalexitprogramme am Clientende eines MQI-Kanals zu nennen.

Wenn ein Kanal Transport Layer Security (TLS) verwendet, kann eine Clientanwendung auch Informationen bereitstellen, die sich auf TLS in der MQCD-Struktur beziehen. Zusätzliche Informationen in Bezug auf TLS können in der Struktur der TLS-Konfigurationsoptionen (MQSCO) bereitgestellt werden, auf die auch die MQCNO-Struktur in einem **MQCONNX** -Aufruf verweist.

Weitere Informationen zu den MQCNO-, MQCD-und MQSCO-Strukturen finden Sie in MQCNO , MQCD und MQSCO .

**Anmerkung:** Das Beispielprogramm für MQCONNX heißt **amqscnxc**. Ein weiteres Beispielprogramm namens **amqsssl c** veranschaulicht die Verwendung der MQSCO-Struktur.

**Zugehörige Tasks**

„Clientverbindungskanal unter IBM MQ MQI client mit MQSERVER erstellen” auf Seite 39
Sie können einen Clientverbindungskanal auf einer Client-Workstation mit der Umgebungsvariablen **MQSERVER** definieren.

# Serververbindungs- und Clientverbindungsdefinitionen auf dem Server erstellen

Sie können beide Definitionen auf dem Server erstellen und dann die Clientverbindungsdefinition für den Client verfügbar machen.

## Informationen zu diesem Vorgang

Sie definieren zunächst einen Serververbindungskanal und definieren dann einen Clientverbindungskanal:

- Auf allen Plattformen können Sie IBM MQ-Scriptbefehle (MQSC-Befehle) oder PCF-Befehle (PCF = Programmable Command Format) verwenden, um einen Serververbindungskanal auf der Servermaschine zu definieren.
- Windows Linux Unter Linux und Windows können Sie auch IBM MQ Explorer verwenden.
- z/OS Unter z/OS können Sie auch die Fenster "Operation" und "Control" verwenden.
- IBM i Unter IBM i können Sie auch die Anzeigenschnittstelle verwenden.

Auf dem Server erstellte Clientverbindungskanaldefinitionen werden Clients unter Verwendung einer Definitionstabelle für Clientkanäle (CCDT, Client Channel Definition Table) zur Verfügung gestellt.

## Vorgehensweise

1. Informationen zum Definieren eines Serververbindungskanals finden Sie in „Serververbindungskanal auf dem Server definieren” auf Seite 58.
2. Informationen zum Definieren eines Clientverbindungskanals finden Sie in „Definieren des Clientverbindungskanals auf dem Server” auf Seite 58.

**Zugehörige Tasks**

„CCDT im Binärformat konfigurieren” auf Seite 45
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

„Serververbindungskanal auf dem Server definieren” auf Seite 58
Erstellen Sie eine Serververbindungskanaldefinition für den Warteschlangenmanager.

„Definieren des Clientverbindungskanals auf dem Server” auf Seite 58
Nachdem Sie den Serververbindungskanal definiert haben, können Sie den entsprechenden Clientverbindungskanal definieren.

„Zugreifen auf Clientverbindungskanaldefinitionen” auf Seite 60
Sie können die Clientkanaldefinitionstabelle (CCDT) für Clientanwendungen verfügbar machen, indem Sie sie kopieren oder gemeinsam nutzen. Geben Sie dann die Position und den Namen des Clients auf dem Client-Computer an. Sie können eine Definitionstabelle für Clientkanäle (CCDT) auch über eine URLlokalisieren.

# Definitionstabelle für den Clientkanal konfigurieren

In einer Definitionstabelle für Clientkanal (Client Channel Definition Table, CCDT) werden Clientverbindungskanäle und die zugehörigen Attribute definiert. Clients lesen die Datei, um zu ermitteln, zu welchen Warteschlangenmanagern eine Verbindung hergestellt werden soll. Die CCDT-Datei kann im JSON-oder Binärformat vorliegen.

## Informationen zu diesem Vorgang

Der WS-Manager liest die CCDT-Datei nicht. Sie wird nur zur Bereitstellung von Kanaldefinitionen und Authentifizierungsinformationen für Clients verwendet.

Vor IBM MQ 9.2.0 war die CCDT nur im Binärformat verfügbar. Ab IBM MQ 9.2.0 können Sie eine CCDT-Datei auch im JSON-Format (JavaScript Object Notation) erstellen.

Beim Erstellen eines Warteschlangenmanagers wird automatisch eine CCDT im Binärformat erstellt. Zum Aktualisieren der Clientkanaldefinitionen, die in dieser Tabelle gespeichert sind, verwenden Sie nur den Befehl **runmqsc**.

Bei einer CCDT im JSON-Format handelt es sich um eine unstrukturierte Textdatei mit der Erweiterung '.json'. Diese Tabelle wird manuell erstellt und aktualisiert, was weniger restriktiv als die Verwendung des Befehls **runmqsc** ist.

z/OS z/OS JMS-Clients, die in einem Anwendungsserver ausgeführt werden, verwenden eine CCDT, um auf Einzelheiten zur Verbindung zu einem fernen Warteschlangenmanager zu verweisen. Ab IBM MQ for z/OS 9.1 stellt IBM MQ Advanced for z/OS Value Unit Edition die Funktion bereit, JMS-Clients über Fernzugriff mit Warteschlangenmanagern auf anderen z/OS-LPARs zu verbinden. Deshalb können diese Clients ebenfalls CCDTs verwenden.

Informationen zur Konfiguration von CCDTs für die Arbeit mit Ihren Clients finden Sie in den folgenden Tasks:

## Prozedur

- „CCDT im Binärformat konfigurieren" auf Seite 45
- „CCDT im JSON-Format konfigurieren" auf Seite 47
- „Positionen für die CCDT" auf Seite 55
- „URL-Zugriff auf die CCDT" auf Seite 56

**Zugehörige Konzepte**

MQI-Client: Definitionstabelle für Clientkanäle (CCDT)

**Zugehörige Tasks**

„Uniform-Cluster konfigurieren" auf Seite 442

Mit Uniform-Clustern können Anwendungen im Hinblick auf Umfang und Verfügbarkeit entwickelt werden und sie können eine Verbindung zu jedem Warteschlangenmanager in diesem Uniform-Cluster herstellen.

### *CCDT im Binärformat konfigurieren*

Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

## Vorbereitende Schritte

Ab IBM MQ 9.1.2 können Sie eine CCDT auch im JSON-Format (JavaScript Object Notation) erstellen. Die Verwendung dieses alternativen Formats hat einige Vorteile gegenüber einer binären CCDT. Siehe „CCDT im JSON-Format konfigurieren" auf Seite 47.

Clients auf allen Plattformen können CCDTs anzeigen und verwenden. Allerdings kann die binäre CCDT nur unter IBM MQ for Multiplatforms erstellt und geändert werden.

## Informationen zu diesem Vorgang

Multi Unter Multiplatforms:

- Ein binäres CCDT wird automatisch im Verzeichnis `@ipcc` unter dem Datenverzeichnis für den Warteschlangenmanager erstellt.
- Die einem Warteschlangenmanager zugeordnete binäre CCDT wird automatisch erstellt und mit den Objektdefinitionen synchronisiert. Wenn Sie ein Clientkanalobjekt definieren, ändern oder löschen, werden die Definition des Warteschlangenmanagerobjekts und der Eintrag in der CCDT als Teil der gleichen Operation aktualisiert.

  **Anmerkungen:**

  – Die IBM MQ CCDT-Datei ist so ausgelegt, dass sie erst dann verkleinert wird, wenn alle vom Benutzer definierten Clientverbindungskanäle auch tatsächlich definiert sind. Wenn ein Clientverbindungska-

nal gelöscht wird, wird er in der CCDT-Datei nur als gelöscht markiert, aber er wird nicht physisch entfernt.

– Setzen Sie den folgenden Befehl ab, um die CCDT-Datei zu verkleinern, nachdem Sie einen oder mehrere Clientverbindungskanäle gelöscht haben:

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

- Mit dem Befehl **runmqsc** ändern sie die Position und die Inhalte der binären CCDT.

Clients auf allen Plattformen können eine binäre CCDT anzeigen und verwenden.

## Prozedur

- Multi
  Erstellen Sie eine binäre Standard-CCDT.

  Unter Multiplatforms wird beim Erstellen eines Warteschlangenmanagers ein Standard-Binär-CCDT mit dem Namen AMQCLCHL.TAB erstellt.

  AMQCLCHL.TAB befindet sich standardmäßig im folgenden Verzeichnis auf einem Server:

  – IBM i Unter IBM i im integrierten Dateisystem:

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

  – Linux AIX Auf Systemen mit AIX and Linux:

    */prefix*/qmgrs/*QUEUEMANAGERNAME*/@ipcc

    Bei dem Namen des Verzeichnisses, auf das *QUEUEMANAGERNAME* verweist, muss auf AIX and Linux -Systemen die Groß-/Kleinschreibung beachtet werden. Der Verzeichnisname ist möglicherweise nicht mit dem Namen des Warteschlangenmanagers identisch, wenn der Name des Warteschlangenmanagers Sonderzeichen enthält.

  – Windows Unter Windows:

    *MQ_INSTALLATION_PATH*\data\qmgrs*QUEUEMANAGERNAME*\@ipcc

    Dabei steht *MQ_INSTALLATION_PATH* für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

  Es kann jedoch sein, dass Sie ein anderes Verzeichnis für WS-Manager-Daten verwenden möchten. Sie können den Parameter **-md** *DataPath* angeben, wenn Sie den Befehl **crtmqm** verwendet haben. Wenn Sie dies tun, befindet sich AMQCLCHL.TAB im @ipcc-Verzeichnis des von Ihnen angegebenen *Datenpfad*s.

- CCDT suchen:

  – Auf dem Client-Computer
  – An einer Position, die von mehreren Clients gemeinsam genutzt wird
  – Auf dem Server als gemeinsam genutzte Datei

  Weitere Informationen finden Sie unter „Positionen für die CCDT” auf Seite 55.

  a) Erstellen Sie direkt auf der Clientmaschine eine binäre CCDT.

    – Verwenden Sie den Befehl runmqsc mit dem Parameter **-n**.
    – Die CCDT wird an der durch **MQCHLLIB** angegebenen Position und mit dem durch **MQCHLTAB** angegebenen Dateinamen erstellt, der standardmäßig AMQCLCHL.TAB lautet.
    – **Wichtig:** Wenn Sie den Parameter **-n** angeben, dürfen Sie keinen anderen Parameter angeben.

  b) Ändern Sie die Position.

Sie können den Pfad zur CCDT ändern, indem Sie **MQCHLLIB** festlegen. Beachten Sie, dass bei der Verwendung mehrerer Warteschlangenmanager auf dem gleichen Server diese nicht die gleiche CCDT-Position gemeinsam nutzen.

- Auf die CCDT zugreifen

  Sie können folgendermaßen auf die CCDT zugreifen:

  – Über Fernzugriff von einer Datei-, FTP-oder HTTP-URL, indem die Umgebungsvariable **MQCCDTURL** definiert wird.
  – Lokal durch Festlegen der Umgebungsvariablen **MQCHLLIB** und **MQCHLTAB**.
  – Lokal durch Definieren der Attribute **ChannelDefinitionDirectory** und **ChannelDefinitionFile** der Zeilengruppe CHANNELS in der Clientkonfigurationsdatei.

  Im Abschnitt „Positionen für die CCDT" auf Seite 55 finden Sie verschiedene Beispiele.

- Zeigen Sie die Inhalte der CCDT an oder bearbeiten Sie diese.

  Sie können die CCDT-Inhalte mit dem Befehl **runmqsc** anzeigen:

  1. Setzen Sie die Umgebungsvariablen auf Access the CCDT (Zugriff auf die CCDT)
  2. Führen Sie den Befehl `runmqsc -n` aus.
  3. Führen Sie den Befehl DISPLAY CHANNEL(*) aus, z. B.

  Multi Unter Multiplatforms können Sie die Inhalte der binären CCDT auch mit dem Befehl **runmqsc** bearbeiten. Jeder Eintrag einer CCDT stellt eine Clientverbindung zu einem bestimmten Warteschlangenmanager dar. Ein neuer Eintrag wird hinzugefügt, wenn Sie einen Clientverbindungskanal mit dem Befehl **DEFINE CHANNEL** definieren, und der Eintrag wird aktualisiert, sobald Sie die Clientverbindungskanäle mit dem Befehl **ALTER CHANNEL** ändern. Weitere Beispiele zur Verwendung des Befehls finden Sie unter **runmqsc**.

- Stellen Sie Clients die Authentifizierungsdaten bereit, um den Widerruf des TLS-Zertifikats zu prüfen.
  a) Definieren Sie eine Namensliste mit den Authentifizierungsdatenobjekten.
  b) Setzen Sie das Warteschlangenmanagerattribut **SSLCRLNL** auf den Namen der Namensliste.

**Zugehörige Konzepte**
Mit widerrufenden Zertifikaten arbeiten
**Zugehörige Tasks**
„CCDT im JSON-Format konfigurieren" auf Seite 47
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Verwenden Sie einen Texteditor, um eine CCDT im JSON-Format (JavaScript Object Notation) zu erstellen und zu aktualisieren.

## *CCDT im JSON-Format konfigurieren*

Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Verwenden Sie einen Texteditor, um eine CCDT im JSON-Format (JavaScript Object Notation) zu erstellen und zu aktualisieren.

### Vorbereitende Schritte

Multi Wenn Sie IBM MQ for Multiplatforms verwenden, können Sie stattdessen die CCDT im Binärformat verwenden, die beim Erstellen eines Warteschlangenmanagers automatisch erstellt wird. Siehe „CCDT im Binärformat konfigurieren" auf Seite 45.

### Informationen zu diesem Vorgang

Der Dateiname des CCDT-Schemas für das JSON-Format lautet:

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

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

Es gibt keine standardmäßige JSON-CCDT und IBM MQ stellt kein Tool bereit, mit dem CCDTs im JSON-Format erstellt oder bearbeitet werden. Sie haben beim manuellen Entwickeln einer JSON-CCDT dennoch mehr Konfigurationsoptionen als bei der Verwendung des Befehls **runmqsc** für die Arbeit mit einer binären CCDT:

- Sie müssen nicht IBM MQ for Multiplatforms zum Erstellen und Bearbeiten einer JSON-CCDT-Datei verwenden.
- Wenn Sie das JSON-Format verwenden können Sie doppelte Kanaldefinitionen mit dem gleichen Namen definieren. Beim Implementieren von IBM MQ in der Cloud wird Ihre Implementierung dadurch skalierbar und erhält eine hohe Verfügbarkeit.
- Die JSON-Datei ist eine lesbare Datei, was die Konfiguration des Warteschlangenmanagers vereinfacht.
- Ein flaches Dateiformat kann in die folgenden Komponenten integriert werden:
  - Tools zur Versionssteuerung für die Überwachung des CCDT-Protokolls
  - Tools für die Automatisierung in fortlaufender Bereitstellung
- Für die Verwaltung der CCDT-Datei ist kein spezialisiertes Tool erforderlich.
- Die Datei ist kleiner.
- Mit diesem Format wird die Auf- und Abwärtskompatibilität bereitgestellt.

**Anmerkungen:**

1. In der standardmäßigen JSON-Datei sind doppelte Schlüssel gültig. Der JSON-Parser verwendet beim Zuordnen von Attributen jedoch nur den letzten Lesewert von doppelten Schlüsseln. Daher muss jeder Kanal beim Definieren doppelter Kanäle ein Element eines Array-Werts sein, der dem Schlüssel 'channel' zugeordnet ist.
2. JSON-CCDTs unterstützen nicht das Speichern der Positionen von LDAP-Servern (Lightweight Directory Access Protocol) für die Standortinformationen von CRL-Respondern (Certificate Revocation Lists) und OCSP-Respondern (Online Certificate Status Protocol).

*Tabelle 7. Voraussetzungen für die Codierung nach Plattform*

| Plattform | JMS-Client-Codierung | C-Client-Codierung |
|---|---|---|
| IBM i IBM i | ASCII | EBCDIC |
| ALW AIX, Linux, and Windows | ASCII | ASCII |
| z/OS z/OS | ASCII oder EBCDIC | Nicht zutreffend |

**Achtung:** Wenn Sie eine Definition für einen Kanal über eine JSON-CCDT bereitstellen (einschließlich einer *Sparse* -Definition, die nicht alle Attribute enthält) wird eine vollständige Kanaldefinition mit allen definierten Attributen erstellt, wobei die Standardwerte für alle Elemente verwendet werden, die nicht in der JSON angegeben sind.

Daher müssen Sie bestimmte Werte für jedes Attribut angeben, für das der Standardwert nicht verwendet werden soll.

## Prozedur

- JSON-CCDT erstellen
  a) Erstellen Sie eine unstrukturierte Datei mit einer `.json`-Erweiterung mit einem generischen Texteditor.

b) Definieren Sie eine CCDT.

Siehe „JSON-CCDT-Beispiele” auf Seite 52 und „Kanalattribute, die von der JSON-CCDT unterstützt werden” auf Seite 50.

- CCDT suchen:
  - Auf dem Client-Computer
  - An einer Position, die von mehreren Clients gemeinsam genutzt wird
  - Auf dem Server als gemeinsam genutzte Datei

  Weitere Informationen finden Sie unter „Positionen für die CCDT” auf Seite 55.
- JSON-CCDT prüfen

  Prüfen Sie die CCDT anhand des Schemas mit einem JSON-Analysetool.

  Informationen zum Erstellen einer CCDT-Datei mit zwei Kanälen und zum Prüfen ihrer Funktionsweise finden Sie unter Vorgehensweise zum Validieren einer IBM MQ -CCDT-JSON-Datei anhand des Schemas .

  Das CCDT-Schema ist in die Produkt- und Clientpakete integriert:

  - Linux AIX Auf Systemen mit AIX and Linux:

    `$MQ_INSTALLATION_PATH/lib` und `/lib` in den Produkt- bzw. Clientpaketen.
  - Windows Unter Windows:

    `%MQ_INSTALLATION_PATH%\bin` und `\bin` in den Produkt- bzw. Clientpaketen.

  **Anmerkungen:**

  - JSON-Analysetools sind online verfügbar.
  - Das Schema definiert obligatorische Attribute mit dem Schlüssel '`required`'.
  - Das Schema definiert Attributdatentypen mit dem Schlüssel '`type`'.
- Auf die CCDT zugreifen

  Sie können folgendermaßen auf die CCDT zugreifen:

  - Über Fernzugriff von einer Datei-, FTP-oder HTTP-URL, indem die Umgebungsvariable **MQCCDTURL** definiert wird.
  - Lokal durch Festlegen der Umgebungsvariablen **MQCHLLIB** und **MQCHLTAB**.
  - Lokal durch Definieren der Attribute **ChannelDefinitionDirectory** und **ChannelDefinitionFile** der Zeilengruppe CHANNELS in der Clientkonfigurationsdatei.

  Im Abschnitt „Positionen für die CCDT” auf Seite 55 finden Sie verschiedene Beispiele.
- CCDT-Inhalte anzeigen oder bearbeiten

  Jeder Eintrag einer CCDT stellt eine Clientverbindung zu einem bestimmten Warteschlangenmanager dar. Sie können die CCDT-Inhalte mit einem Texteditor anzeigen oder bearbeiten.

  Wenn Sie die CCDT nur anzeigen möchten, können Sie dazu auch den Befehl **runmqsc** folgendermaßen verwenden:

  1. Legen Sie die Umgebungsvariable fest, damit Sie wie im vorherigen Schritt beschrieben auf die CCDT zugreifen können.
  2. Führen Sie den Befehl `runmqsc -n` aus. Weitere Informationen finden Sie im Abschnitt runmqsc.
  3. Führen Sie den Befehl **DISPLAY CHANNEL** aus. Führen Sie beispielsweise `DISPLAY CHANNEL(*)` aus.

**Zugehörige Konzepte**

Mit widerrufenden Zertifikaten arbeiten

**Zugehörige Tasks**

„CCDT im Binärformat konfigurieren" auf Seite 45
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

„Uniform-Cluster konfigurieren" auf Seite 442
Mit Uniform-Clustern können Anwendungen im Hinblick auf Umfang und Verfügbarkeit entwickelt werden und sie können eine Verbindung zu jedem Warteschlangenmanager in diesem Uniform-Cluster herstellen.

*Kanalattribute, die von der JSON-CCDT unterstützt werden*
In diesem Abschnitt finden Sie eine Liste der Attribute für den Clientverbindungskanal, die von der Definitionstabelle für den Clientkanal (Client Channel Definition Table, CCDT) im JSON-Format unterstützt werden. Bei dieser Liste handelt es sich um eine Untergruppe der Attribute, die von der binären CCDT unterstützt werden.

## Attributzuordnung

Diese Attribute werden in das folgende Kanalobjekt eingefügt:

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

Dabei ist $CHANNEL_X_KEY_VALUE_LIST eine durch Kommas getrennte Liste der Attribute, die in der folgenden Tabelle aufgeführt sind.

Im Abschnitt „JSON-CCDT-Beispiele" auf Seite 52 finden Sie grundlegende Anwendungsfälle.

Das JSON-Schema wird in `/opt/mqm/lib/ccdt_schema.json`ausgeliefert. Um herauszufinden, welche Werte für die einzelnen Attribute gültig sind, sehen Sie sich das JSON-Schema an.

In der folgenden Tabelle werden das JSON-Objekt, der JSON-Schlüssel und der JSON-Datentyp zusammen mit der zugehörigen Definition für das binäre Kanalattribut aufgeführt.

**Achtung:** Die erforderlichen Attribute sind der Kanal **name** und der Kanal **type**. Wenn Sie außerdem den Bereich **portRange** definieren, sind die Attribute *low* und *high* ebenfalls erforderlich.

| JSON-Objekt | JSON-Schlüssel | JSON-Datentyp | Binäre Attributdefinition |
|---|---|---|---|
| channel (array) | Name | STRING | CHANNEL |
| channel (array) | Typ | STRING | CHLTYPE |
| channel.clientConnection | queueManager | STRING | QMNAME |
| channel.clientConnection.connection (array) | host | STRING | CONNAME |
| channel.clientConnection.connection | port | INT | CONNAME |
| channel.compression.header (array) | Header | STRING | COMPHDR |
| channel.compression.message (array) | das Kundenstamms | STRING | COMPMSG |
| channel.connectionManagement | Affinität | STRING | AFFINITY |
| channel.connectionManagement | clientWeight | INT | CLNTWGHT |
| channel.connectionManagement | defaultReconnect | STRING | DEFRECON |
| channel.connectionManagement | disconnectInterval | INT | DISCINT |
| channel.connectionManagement | heartInterval | INT | HBINT |
| channel.connectionManagement | keepAliveInterval | INT | KAINT |

| JSON-Objekt | JSON-Schlüssel | JSON-Datentyp | Binäre Attributdefinition |
|---|---|---|---|
| channel.connectionManagement | sharingConversations | INT | SHARECNV |
| channel.connectionManagement.localAddress (array) | host | STRING | LOCLADDR |
| channel.connectionManagement.localAddress (array) | port | INT | LOCLADDR |
| channel.connectionManagement.localAddress.portRange | hoch | INT | LOCLADDR |
| channel.connectionManagement.localAddress.portRange | Niedrig | INT | LOCLADDR |
| channel.exits.receive (array) | Name | STRING | RCVEXIT |
| channel.exits.receive (array) | userData | STRING | RCVDATA |
| channel.exits.security | Name | STRING | SCYEXIT |
| channel.exits.security | userData | STRING | SCYDATA |
| channel.exits.send (array) | Name | STRING | SENDEXIT |
| channel.exits.send (array) | userData | STRING | SENDDATA |
| channel.general | Beschreibung | STRING | DESCR |
| channel.general | maximumMessageLength | INT | MAXMSGL |
| channel.timestamps | altered | STRING | ALTDATE und ALTTIME |
| channel.transmissionSecurity | certificateLabel | STRING | CERTLABL |
| channel.transmissionSecurity | certificatePeerName | STRING | SSLPEER |
| channel.transmissionSecurity | cipherSpecification | STRING | SSLCIPH |

**Anmerkungen:**

- `channel.connectionManagement.localAddress` kann als eine der folgenden Kombination aus Schlüsseln definiert sein:
  - host und port
  - host und portRange
  - port
  - portRange
- Der JSON-Schlüssel `channel.timestamps altered` ist optional und nimmt, falls nicht anders definiert, als Wert standardmäßig die Zeit der letzten Änderung der JSON-CCDT-Datei an. Wenn die Umgebung jedoch so konfiguriert ist, dass sie die CCDT aus einer URL abruft, ist der Standardwert die Zeit, zu der die Datei zuletzt heruntergeladen wurde.
- `channel.clientConnection.connection` muss die Schlüssel 'host' und 'port' enthalten.
- Der geänderte Schlüssel ist eine einzelne Zeichenfolge, die die Attribute ALTDATE und ALTTIME zusammenfasst.
- Der Transporttyp kann nur TCP sein, deshalb sind die folgenden Attribute nicht im Schema definiert:
  - **TRPTYPE**
  - **USERID**

- **PASSWORD**
- **MODENAME**
- **TPNAME**

**Zugehörige Verweise**
Kanalattribute für Kanaltypen

*JSON-CCDT-Beispiele*
Verwenden Sie die in diesem Abschnitt aufgeführten Beispiele als Basis für Ihre Anforderungen.

Öffnen Sie einen generischen Texteditor und kopieren Sie eines der folgenden Beispiele:



## Einfache Clientverbindung definieren

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

## Einen Kanal und einen Warteschlangenmanager mit TLS definieren

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

```
  ]
}
```

## Definieren Sie einen Kanal und einen Warteschlangenmanager, die TLS nicht verwenden

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

## Zwei Kanäle mit dem gleichen Namen definieren

Jeder Kanal stellt eine Verbindung zu zwei unterschiedlichen Warteschlangenmanagern her:

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

## Vollständige Liste der CCDT-Kanalattributdefinitionen für einen Clientverbindungskanal

```
{
  "channel":
  [
    {
      "compression":
      {
        "header": [ "system" ],
        "message": [ "zlibfast" ]
      },
      "connectionManagement":
      {
        "sharingConversations": 10,
        "clientWeight": 1,
        "affinity": "none",
        "defaultReconnect": "yes",
        "heartbeatInterval": 600,
        "keepAliveInterval": -1,
        "localAddress":
        [
          {
            "portRange":
            {
              "low": 2020,
              "high": 3030
            }
          }
        ]
      },
      "exits":
      {
        "receive":
        [
          {
          "name": "",
          "userData": ""
          }
        ],
        "security":
        {
          "name": "",
          "userData": ""
        },
        "send":
        [
           {
             "name": "",
             "userData": ""
           }
        ]
      },
      "general":
      {
        "description": "First channel",
        "maximumMessageLength": 4194304
      },
      "name": "the_channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ],
        "queueManager": "QM1"
      },
      "timestamps":
      {
        "altered": "2018-12-04T15:37:22.000Z"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "",
        "certificateLabel": "",
        "certificatePeerName": ""
      },
```

```
      "type": "clientConnection"
    }
  ]
}
```

**Zugehörige Verweise**
Kanalattribute für Kanaltypen
Kanalattribute in alphabetischer Reihenfolge

### *Positionen für die CCDT*

IBM MQ unterstützt das Abrufen einer CCDT aus einer Datei, über eine FTP- oder HTTP-URL. Sie können die CCDT als gemeinsam genutzte Datei für den Client zugänglich machen, während sie sich weiterhin auf dem Server befindet. Alternativ können Sie CCDT verteilen, indem Sie sie auf einzelne Client-Computer oder in eine Position kopieren, die von mehreren Clients gemeinsam genutzt wird.

Wenn Sie die Datei mit FTP kopieren, verwenden Sie die Option `bin` , um den Binärmodus festzulegen; verwenden Sie nicht den Standardmodus `ASCII` . Whichever Methode, die Sie auswählen, um die CCDT verfügbar zu machen, die Position muss sicher sein, um unberechtigte Änderungen an den Kanälen zu verhindern.

## CCDT-Datei auf einem Server hosten

Ab IBM MQ 9.0 kann die CCDT in einer zentralen Position gehostet werden, auf die über eine URL zugegriffen wird. Dadurch muss die CCDT nicht für jeden implementierten Client einzeln aktualisiert werden. In IBM MQ 9.0 wurde die Funktion für native (C/C++, COBOL und RPG) und nicht verwaltete .NET-Anwendungen hinzugefügt, um die CCDT aus einer URL zu extrahieren, bei der es sich um eine lokale Datei, eine FTP- oder HTTP-Ressource handeln kann.

Das standardmäßige Caching-Verhalten von IBM MQ-Clients besteht darin, dass eine CCDT-Datei erst dann abgerufen wird, wenn sich die Dateiänderungszeit vom Zeitpunkt, zu dem sie letztmalig abgerufen wurde, unterscheidet. Wie bei den meisten Clientkonfigurationsoptionen gibt es verschiedene Möglichkeiten, wie die URL-Position bereitgestellt werden kann:

- **`CCDTUrlPtr`** und **`CCDTUrlOffset`** über die MQCNO-Struktur, die an MQI-Aufruf MQCONNX übergeben wird
- Umgebungsvariable **`MQCCDTURL`**
- Attribut **`ChannelDefinitionDirectory`** in der Zeilengruppe 'Channels' von `mqclient.ini`

Es werden sowohl authentifizierte als auch nicht authentifizierte URLs unterstützt. Einige Beispiele:

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

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

Wenn Sie diese Unterstützung mit FTP oder HTTP verwenden möchten, müssen Sie die CCDT-Datei weiterhin auf einem Server hosten. Durch die in IBM MQ 9.0 hinzugefügte Unterstützung können aber alle Ihre Clientanwendungen Änderungen an Kanaldefinitionen automatisch übernehmen, ohne dass Aktualisierungen manuell durchgeführt werden müssen oder auf jedem Client ein Netzdateisystem angehängt werden muss. Weitere Informationen finden Sie unter „URL-Zugriff auf die CCDT” auf Seite 56.

## Informationen zum Angeben der Position der CCDT auf dem Client

Auf einem Clientsystem können Sie die Position von CCDT auf folgende Arten angeben:

- Verwenden Sie die Umgebungsvariablen **`MQCHLLIB`** , um das Verzeichnis anzugeben, in dem sich die Tabelle befindet, und **`MQCHLTAB`** , um den Dateinamen der Tabelle anzugeben.
- Die Clientkonfigurationsdatei wird verwendet. Verwenden Sie in der Zeilengruppe CHANNELS das Attribut **`ChannelDefinitionDirectory`** , um das Verzeichnis anzugeben, in dem sich die Tabelle befindet, und das Attribut **`ChannelDefinitionFile`** , um den Dateinamen anzugeben.

- Durch die Bereitstellung einer URL (Datei, FTP oder HTTP) für eine CCDT, die in einer zentralen Position wie zuvor beschrieben gehostet wird.

Wenn die Position sowohl in der Clientkonfigurationsdatei als auch unter Verwendung von Umgebungsvariablen angegeben wird, müssen die Umgebungsvariablen Priorität haben. Sie können diese Funktion verwenden, um eine Standardposition in der Clientkonfigurationsdatei anzugeben und diese bei Bedarf mit Umgebungsvariablen zu überschreiben.

Wenn Sie eine URL verwenden, um die Position der CCDT anzugeben, entspricht die Vorrangregelung für eine native Clientanwendung bei der Suche nach der Clientkanaldefinition der Beschreibung in „URL-Zugriff auf die CCDT" auf Seite 56.

### *URL-Zugriff auf die CCDT*

Eine Clientkanaldefinitionstabelle (CCDT) kann in einer zentralen Position gehostet werden, auf die über eine URL zugegriffen werden kann. Dadurch muss die CCDT nicht für jeden implementierten Client einzeln aktualisiert werden.

Ab IBM MQ 9.0 kann eine Clientkanaldefinitionstabelle über eine URL auf eine der folgenden Arten gefunden werden:

- Nach Programmierung mit MQCNO
- Durch Verwendung von Umgebungsvariablen

  **Achtung:** Sie können die Umgebungsvariablenoption verwenden, um die URL nur für native Programme bereitzustellen, die eine Verbindung als Clients herstellen, d. h. C-, COBOL-oder C + + -Anwendungen. Die Umgebungsvariablen haben keine Auswirkung auf Java-, JMS- oder verwaltete .NET-Anwendungen.

  IBM MQ unterstützt das Abrufen einer CCDT aus einer Datei, über eine FTP- oder HTTP-URL.
- Mit der Zeilengruppe `mqclient.ini` file CHANNELS.

Mit der Umgebungsvariablen **MQCCDTURL** können Sie eine Datei-, FTP-oder HTTP- URL als Einzelwert angeben, aus dem eine Definitionstabelle für Clientkanäle abgerufen werden kann.

Sie können auch den durch die Umgebungsvariable **MQCHLLIB** angegebenen Verzeichnispfad (oder den durch das Attribut **ChannelDefinitionDirectory** in der „Zeilengruppe 'CHANNELS' in der Clientkonfigurationsdatei" auf Seite 187angegebenen Pfad) verwenden, um eine CCDT-Datei zu lokalisieren, entweder über die Datei-, FTP-oder HTTP- URL, zusätzlich zum vorhandenen lokalen Dateisystemverzeichnis, d. h. `/var/mqm`). Beachten Sie, dass ein **MQCHLLIB** -Wert ein Verzeichnisstamm ist und in Kombination mit **MQCHLTAB** die vollständig qualifizierte URLableitet.

Die Basisauthentifizierung für Verbindungen wird durch die in der URL codierten Berechtigungsnachweise unterstützt:

**Authentifizierte Verbindungen**

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

**Nicht authentifizierte Verbindungen**

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

**Anmerkung:** Wenn Sie authentifizierte Verbindungen verwenden möchten, müssen Sie, wie bei JMS, den Benutzernamen und das Kennwort angeben, die in der URL codiert sind.

Die Rangfolge für eine native Clientanwendung, um eine Clientkanaldefinition zu finden, ist jetzt:

1. MQCD, die von **ClientConnOffset** und **ClientConnPtr** in MQCNO bereitgestellt wird.
2. URL, die von **CCDTUrlOffset** und **CCDTUrlPtr** in MQCNO bereitgestellt wird.
3. *MQSERVER* -Umgebungsvariable.

4. Wenn eine `mqclient.ini` -Datei definiert ist und die Zeilengruppe 'Channels' ein Attribut **ServerConnectionParms** enthält, dann wird der von ihr definierte Kanal verwendet. Weitere Informationen finden Sie unter „IBM MQ MQI client -Konfigurationsdatei, mqclient.ini" auf Seite 168 und „Zeilengruppe 'CHANNELS' in der Clientkonfigurationsdatei" auf Seite 187.
5. **MQCCDTURL** -Umgebungsvariable.
6. Umgebungsvariable **MQCHLLIB** und **MQCHLTAB** .
7. **ChannelDefinitionDirectory** und **ChannelDefinitionFile** in der „Zeilengruppe 'CHANNELS' in der Clientkonfigurationsdatei" auf Seite 187.

**Wichtig:** Der Zugriff auf eine CCDT-Datei mit einer URL öffnet immer eine schreibgeschützte Kopie der Datei, auch wenn das `file://`-Protokoll verwendet wird.

Bei dem Versuch, eine CCDT-Datei für Schreibzugriff zu öffnen, z. B. bei Verwendung des MQSC-Befehls **DEFINE CHANNEL** von einem Client, wird eine Fehlernachricht zurückgegeben, die angibt, dass die Datei nicht für Schreibzugriff geöffnet werden konnte.

Es ist jedoch möglich, Kanal-und Authentifizierungsinformationsdefinitionen mit **runmqsc**zu lesen.

**Zugehörige Tasks**

„Zugreifen auf Clientverbindungskanaldefinitionen" auf Seite 60
Sie können die Clientkanaldefinitionstabelle (CCDT) für Clientanwendungen verfügbar machen, indem Sie sie kopieren oder gemeinsam nutzen. Geben Sie dann die Position und den Namen des Clients auf dem Client-Computer an. Sie können eine Definitionstabelle für Clientkanäle (CCDT) auch über eine URLlokalisieren.

„CCDT im Binärformat konfigurieren" auf Seite 45
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

Verwenden einer CCDT mit IBM MQ classes for JMS

**Zugehörige Verweise**

CCDTURL
MQCNO - Verbindungsoptionen
XMSC_WMQ_CCDTURL

## Windows Clientverbindungskanäle im Active Directory

Auf Windows-Systemen, die das Active Directory unterstützen, veröffentlicht IBM MQ Clientverbindungskanäle im Active Directory, um eine dynamische Client/Serverbindung bereitzustellen.

Wenn Clientverbindungskanalobjekte definiert werden, werden sie in eine Clientkanaldefinitionsdatei geschrieben, die standardmäßig `AMQCLCHL.TAB` heißt. Wenn die Clientverbindungskanäle das TCP/IP-Protokoll verwenden, werden sie vom IBM MQ-Server auch in Active Directory veröffentlicht. Wenn der IBM MQ-Client festlegt, wie eine Verbindung zum Server hergestellt wird, sucht er mithilfe der folgenden Suchreihenfolge nach einer relevanten Kanalobjektdefinition für den Clientverbindungskanal:

1. MQCONNX MQCD-Datenstruktur
2. Umgebungsvariable **MQSERVER**
3. Clientkanaldefinitionsdatei
4. Active Directory

Diese Reihenfolge bedeutet, dass alle aktuellen Anwendungen von keiner Änderung betroffen sind. ie können sich diese Einträge im Active Directory als Datensätze in der Clientkanaldefinitionsdatei vorstellen, und der IBM MQ-Client verarbeitet sie auf die gleiche Weise. Verwenden Sie den Befehl `setmqscp` wie in setmqscp beschrieben, um die Unterstützung für die Veröffentlichung von Clientverbindungskanaldefinitionen in Active Directory zu konfigurieren und zu verwalten.

## Serververbindungskanal auf dem Server definieren

Erstellen Sie eine Serververbindungskanaldefinition für den Warteschlangenmanager.

### Vorgehensweise

1. Definieren Sie auf der Servermaschine einen Kanal mit dem ausgewählten Namen und dem Kanaltyp *server-connection* .
   For example:

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

2. Verwenden Sie den folgenden Befehl, um den Zugriff der Eingangsverbindung auf den Warteschlangenmanager zu ermöglichen:

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

   - Dabei verwendet **SET CHLAUTH** den Namen des im vorherigen Schritt definierten Kanals.
   - Dabei ist *'IP address'* IP-Adresse die IP-Adresse des Clients.
   - Dabei ist *'userid'* die ID, die Sie dem Kanal für die Zugriffssteuerung für die Zielwarteschlangen bereitstellen möchten. Bei diesem Feld muss die Groß-/Kleinschreibung beachtet werden.

   Sie können die eingehende Verbindung mithilfe einer Reihe unterschiedlicher Attribute identifizieren. Das Beispiel verwendet die IP-Adresse. Zu den Alternativattributen gehören die Clientbenutzer-ID und der TLS-Betreffname. Weitere Informationen finden Sie in Kanalauthentifizierungsdatensätze .

   Diese Kanaldefinition ist dem Warteschlangenmanager zugeordnet, der auf dem Server ausgeführt wird.

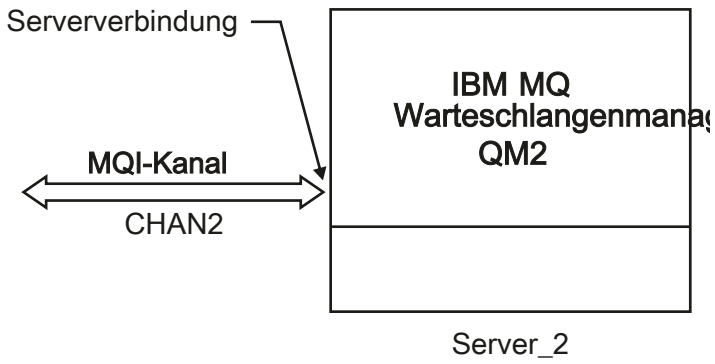



*Abbildung 2. Serververbindungskanal definieren*

**Zugehörige Tasks**

„Definieren des Clientverbindungskanals auf dem Server" auf Seite 58
Nachdem Sie den Serververbindungskanal definiert haben, können Sie den entsprechenden Clientverbindungskanal definieren.

## Definieren des Clientverbindungskanals auf dem Server

Nachdem Sie den Serververbindungskanal definiert haben, können Sie den entsprechenden Clientverbindungskanal definieren.

### Vorbereitende Schritte

Definieren Sie den Serververbindungskanal. Weitere Informationen finden Sie unter „Serververbindungskanal auf dem Server definieren" auf Seite 58.

## Vorgehensweise

1. Definieren Sie einen Kanal mit demselben Namen wie der Serververbindungskanal, aber einen Kanaltyp von *Clientverbindung* . Sie müssen den Verbindungsnamen (CONNAME) angeben. Bei TCP/IP ist der Verbindungsname die Netzadresse oder der Hostname der Servermaschine. Es empfiehlt sich außerdem, den Namen des Warteschlangenmanagers (QMNAME) anzugeben, zu dem Ihre in der Clientumgebung ausgeführte IBM MQ-Anwendung eine Verbindung herstellen soll. Wenn Sie den Namen des Warteschlangenmanagers ändern, können Sie eine Gruppe von Kanälen definieren, um eine Verbindung zu verschiedenen Warteschlangenmanagern herzustellen.

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

2. Verwenden Sie den folgenden Befehl, um den Zugriff der Eingangsverbindung auf den Warteschlangenmanager zu ermöglichen:

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

   - Dabei verwendet der Befehl **SET CHLAUTH** den Namen des im vorherigen Schritt definierten Kanals.
   - Hierbei steht *'IP address'* für die IP-Adresse des Clients.
   - Dabei ist *'userid'* die ID, die Sie dem Kanal für die Zugriffssteuerung für die Zielwarteschlangen bereitstellen möchten. Bei diesem Feld muss die Groß-/Kleinschreibung beachtet werden.

   Sie können die eingehende Verbindung mithilfe einer Reihe unterschiedlicher Attribute identifizieren. Das Beispiel verwendet die IP-Adresse. Zu den Alternativattributen gehören die Clientbenutzer-ID und der TLS-Betreffname. Weitere Informationen finden Sie in Kanalauthentifizierungsdatensätze .

## Ergebnisse

Multi Unter Multiplatforms wird diese Kanaldefinition in einer als Clientkanaldefinitionstabelle (CCDT) bezeichneten Datei gespeichert, die dem WS-Manager zugeordnet ist. Die Definitionstabelle für den Clientkanal kann mehr als eine Kanaldefinition für Clientverbindungen enthalten. Weitere Informationen zur Clientkanaldefinitionstabelle und zu den entsprechenden Informationen darüber, wie Clientverbindungskanaldefinitionen unter z/OS gespeichert werden, finden Sie in „CCDT im Binärformat konfigurieren” auf Seite 45.

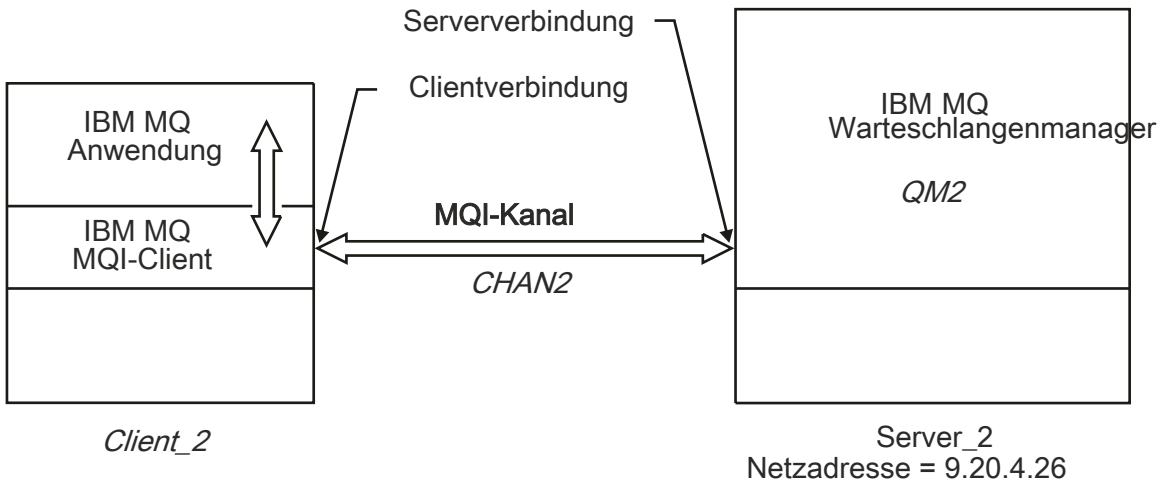



*Abbildung 3. Definieren des Clientverbindungskanals*

**Zugehörige Verweise**

DEFINE CHANNEL (Definieren eines neuen Kanals)

SET CHLAUTH (Kanalauthentifizierungsdatensatz erstellen oder ändern)

## Zugreifen auf Clientverbindungskanaldefinitionen

Sie können die Clientkanaldefinitionstabelle (CCDT) für Clientanwendungen verfügbar machen, indem Sie sie kopieren oder gemeinsam nutzen. Geben Sie dann die Position und den Namen des Clients auf dem Client-Computer an. Sie können eine Definitionstabelle für Clientkanäle (CCDT) auch über eine URLlokalisieren.

### Vorbereitende Schritte

Bei dieser Task wird vorausgesetzt, dass Sie die erforderlichen Clientverbindungskanäle in einer CCDT definiert haben. Weitere Informationen finden Sie unter „Definitionstabelle für den Clientkanal konfigurieren” auf Seite 44.

### Informationen zu diesem Vorgang

Damit eine Clientanwendung die Definitionstabelle für den Clientkanal (CCDT) verwenden kann, müssen Sie die CCDT zur Verfügung stellen und die zugehörige Position und den Namen angeben. Es gibt mehrere Möglichkeiten, dies zu tun:

- Sie können die CCDT auf den Client-Computer kopieren.
- Sie können die CCDT an eine Position kopieren, die von mehreren Clients gemeinsam genutzt wird.
- Sie können die CCDT als gemeinsam genutzte Datei für den Client zugänglich machen, während sie sich weiterhin auf dem Server befindet.

Native IBM MQ -Anwendungen (C/C + +, COBOL und RPG) und nicht verwaltete .NET-Anwendungen können die CCDT, die an einer zentralen Position gehostet wird, aus einer URLextrahieren, unabhängig davon, ob es sich um eine lokale Datei, eine FTP-oder HTTP-Ressource handelt.

### Vorgehensweise

1. Stellen Sie die CCDT für die Clientanwendungen auf eine der folgenden Arten zur Verfügung:
   a) Optional: Kopieren Sie die CCDT auf den Client-Computer.
   b) Optional: Kopieren Sie die CCDT in eine Position, die von mehr als einem Client gemeinsam genutzt wird.
   c) Optional: Lassen Sie die CCDT auf dem Server, lassen Sie sie jedoch vom Client shareable.
   d) Optional: Definieren Sie eine lokale Datei, eine FTP-oder HTTP-URL für eine CCDT, die in einer zentralen Position gehostet wird, sodass native (C/C + +-, COBOL-und RPG-) und nicht verwaltete .NET-Anwendungen die CCDT aus dieser URL extrahieren können.

   Die Position, die Sie für die CCDT auswählen, muss sicher sein, damit nicht autorisierte Änderungen an den Kanälen verhindert werden können.
2. Geben Sie auf dem Client die Position und den Namen der Datei an, die die CCDT auf eine der folgenden drei Arten enthält:
   a) Optional: Verwenden Sie die Zeilengruppe CHANNELS in der Clientkonfigurationsdatei. Weitere Informationen finden Sie unter „Zeilengruppe 'CHANNELS' in der Clientkonfigurationsdatei” auf Seite 187.
   b) Optional: Verwenden Sie die Umgebungsvariablen **MQCHLLIB** und **MQCHLTAB**.

      Sie können die Umgebungsvariablen beispielsweise festlegen, indem Sie Folgendes eingeben:

      - Linux AIX Auf Systemen mit AIX and Linux:

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

- IBM i Auf Systemen mit IBM i:

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

  Dabei steht *MQ_INSTALLATION_PATH* für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

c) Optional: Nur unter Windows: Verwenden Sie den Steuerbefehl **setmqscp**, um die Clientverbindungskanaldefinitionen in Active Directory zu veröffentlichen.

d) Geben Sie die Position eines zentral gehosteten CCDT über eine URL an, entweder durch Programmierung mit MQCNO, unter Verwendung von Umgebungsvariablen oder mithilfe von `mqclient.ini`-Dateizeilengruppen. Weitere Informationen hierzu finden Sie unter „Positionen für die CCDT” auf Seite 55 und „URL-Zugriff auf die CCDT” auf Seite 56.

Wenn die Umgebungsvariable **MQSERVER** festgelegt ist, verwendet ein IBM MQ -Client die Clientverbindungskanaldefinition, die von **MQSERVER** angegeben wird, anstelle aller Definitionen in der Clientkanaldefinitionstabelle.

**Zugehörige Tasks**

„CCDT im Binärformat konfigurieren” auf Seite 45
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

**Zugehörige Verweise**

MQI-Client: Definitionstabelle für Clientkanäle (CCDT)

## ALW Kanalexitprogramme für MQI-Kanäle

Es sind drei Typen von Kanalexits für die IBM MQ MQI client-Umgebung unter AIX, Linux, and Windows verfügbar.

Diese sind:

- Sendeexit
- Empfangsexit
- Sicherheitsexit

Diese Exits sind sowohl auf dem Client als auch auf dem Serverende des Kanals verfügbar. Die Exits sind für Ihre Anwendung nicht verfügbar, wenn Sie die Umgebungsvariable MQSERVER verwenden. Kanalexits werden im Abschnitt Kanalexitprogramme für Messaging-Kanäle erläutert.

Die Sende-und Empfangsexits arbeiten zusammen. Es gibt mehrere Möglichkeiten, wie Sie sie verwenden können:

- Nachricht trennen und erneut assemblieren
- Komprimieren und Dekomprimieren von Daten in einer Nachricht (diese Funktion wird als Teil von IBM MQ bereitgestellt, möglicherweise möchten Sie aber eine andere Komprimierungsmethode verwenden)
- Verschlüsseln und Entschlüsseln von Benutzerdaten (diese Funktion wird als Teil von IBM MQ bereitgestellt, möglicherweise möchten Sie aber eine andere Verschlüsselungsmethode verwenden)
- Journaling jeder gesendeten und empfangenen Nachricht

Sie können den Sicherheitsexit verwenden, um sicherzustellen, dass der IBM MQ-Client und -Server ordnungsgemäß identifiziert werden, und um den Zugriff zu steuern.

Wenn Sende-oder Empfangsexits auf der Serververbindungsseite der Kanalinstanz MQI-Aufrufe für die Verbindung ausführen müssen, der sie zugeordnet sind, verwenden sie die Verbindungskennung, die im

Feld MQCXP Hconn angegeben ist. Sie müssen sich darüber im Klaren sein, dass Sende-und Empfangsexits für Clientverbindungen keine MQI-Aufrufe vornehmen können.

**Zugehörige Konzepte**

„Sicherheitsexits auf einer Clientverbindung” auf Seite 62
Sie können Sicherheitsexitprogramme verwenden, um sicherzustellen, dass der Partner am anderen Ende eines Kanals echt ist. Besondere Hinweise gelten, wenn ein Sicherheitsexit auf eine Clientverbindung angewendet wird.

Benutzerexits, API-Exits und installierbare IBM MQ-Services

**Zugehörige Tasks**

Funktionen des Warteschlangenmanagers erweitern

**Zugehörige Verweise**

„Pfad zu Exits” auf Seite 62
Ein Standardpfad für die Position der Kanalexits ist in der Clientkonfigurationsdatei definiert. Kanalexits werden geladen, wenn ein Kanal initialisiert wird.

„API-Aufruf in einem Sende-oder Empfangsexitprogramm identifizieren” auf Seite 64
Wenn Sie MQI-Kanäle für Clients verwenden, identifiziert Byte 10 des Agentenpuffers den API-Aufruf, der verwendet wird, wenn ein Sende-oder Empfangsexit aufgerufen wird. Dies ist hilfreich, um zu ermitteln, welche Kanalflüsse Benutzerdaten enthalten und möglicherweise eine Verarbeitung wie die Verschlüsselung oder die digitale Signatur erfordern.

## ALW Pfad zu Exits

Ein Standardpfad für die Position der Kanalexits ist in der Clientkonfigurationsdatei definiert. Kanalexits werden geladen, wenn ein Kanal initialisiert wird.

Auf AIX, Linux, and Windows-Systemen wird während der Installation von IBM MQ MQI client eine Clientkonfigurationsdatei zu Ihrem System hinzugefügt. In dieser Datei wird unter Verwendung der Zeilengruppe ein Standardpfad für die Position der Kanalexits auf dem Client definiert:

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

Dabei steht *string* für eine Dateiposition in einem Format, das der Plattform entspricht.

Wenn ein Kanal initialisiert wird, wird nach einem MQCONN -oder MQCONNX -Aufruf die Clientkonfigurationsdatei durchsucht. Die Zeilengruppe 'ClientExitPath' wird gelesen, und alle Kanalexits, die in der Kanaldefinition angegeben sind, werden geladen.

## ALW Sicherheitsexits auf einer Clientverbindung

Sie können Sicherheitsexitprogramme verwenden, um sicherzustellen, dass der Partner am anderen Ende eines Kanals echt ist. Besondere Hinweise gelten, wenn ein Sicherheitsexit auf eine Clientverbindung angewendet wird.

Abbildung 4 auf Seite 64 veranschaulicht die Verwendung von Sicherheitsexits in einer Clientverbindung, wobei der IBM MQ-Objektberechtigungsmanager verwendet wird, um einen Benutzer zu authentifizieren.

Das Feld SecurityParmsPtr oder SecurityParmsOffset in der MQCNO-Struktur wird vom Client festgelegt und an beiden Enden des Kanals befinden sich Sicherheitsexits. Nachdem der normale Sicherheitsnachrichtenaustausch beendet wurde und der Kanal zur Ausführung bereit ist, wird die MQCSP-Struktur an den Clientsicherheitsexit übergeben. Der Exit kann über das Feld SecurityParms in der MQCXP-Struktur auf die MQCSP-Struktur zugreifen. Der Exittyp wird auf MQXR_SEC_PARMS gesetzt. Der Sicherheitsexit kann die Berechtigungsnachweise in der MQCSP-Struktur ändern oder unverändert lassen.

Die vom Exit zurückgegebenen Daten werden dann an die Serververbindungsseite des Kanals gesendet. Die MQCSP-Struktur wird auf der Serververbindungsseite des Kanals erneut erstellt und an den Sicher-

heitsexit der Serververbindung übergeben. Der Exit kann über das Feld SecurityParms in der MQCXP-Struktur auf die MQCSP-Struktur zugreifen. Der Sicherheitsexit empfängt und verarbeitet diese Daten. Bei dieser Verarbeitung werden in der Regel alle Änderungen rückgängig gemacht, die der Client-Exit an den Berechtigungsnachweisen in der MQCSP-Struktur vorgenommen hat, die dann zur Autorisierung der Warteschlangenmanagerverbindung verwendet werden. Die resultierende MQCSP-Struktur wird mithilfe von SecurityParmsPtr in der MQCNO-Struktur auf dem Warteschlangenmanagersystem referenziert.

Die Speicheradresse, die mit dem Feld SecurityParms der MQCXP-Struktur zurückgegeben wird, muss bis MQXR_TERM adressierbar und unverändert bleiben. Ein Exit darf den Speicher nicht ungültig machen oder den Speicher wieder frei machen, bevor der Exit für MQXR_TERM aufgerufen wird.

Wenn das Feld SecurityParmsPtr oder SecurityParmsOffset in der MQCNO-Struktur festgelegt ist und nur an einem Ende des Kanals ein Sicherheitsexit vorhanden ist, empfängt und verarbeitet der Sicherheitsexit die MQCSP-Struktur. Aktionen wie die Verschlüsselung sind für einen einzelnen Benutzerexit nicht geeignet, da kein Exit vorhanden ist, um die ergänzende Aktion auszuführen.

Wenn die Felder SecurityParmsPtr und SecurityParmsOffset in der MQCNO-Struktur nicht festgelegt sind und an einem oder beiden Enden des Kanals ein Sicherheitsexit vorhanden ist, werden die Sicherheitsexits aufgerufen. Jeder Sicherheitsexit kann seine eigene MQCSP-Struktur zurückgeben, die durch das Feld SecurityParmsPtr adressiert wird. Der Sicherheitsexit wird erst wieder aufgerufen, nachdem er beendet wurde (ExitReason von MQXR_TERM). Der Exit-Writer kann den Speicher freigeben, der für den MQCSP in dieser Phase verwendet wird.

Wenn eine Serververbindungskanalinstanz mehr als einen Datenaustausch gemeinsam verwendet, ist das Muster der Aufrufe an den Sicherheitsexit auf den zweiten und nachfolgenden Datenaustausch beschränkt.

Für den ersten Datenaustausch ist das Muster so, als ob die Kanalinstanz keine Dialoge gemeinsam verwendet. Für den zweiten und nachfolgenden Datenaustausch wird der Sicherheitsexit niemals mit MQXR_INIT, MQXR_INIT_SEC oder MQXR_SEC_MSG aufgerufen. Es wird mit MQXR_SEC_PARMS aufgerufen.

In einer Kanalinstanz mit gemeinsamen Gesprächen wird MQXR_TERM nur für den letzten Datenaustausch aufgerufen, der ausgeführt wird.

Jeder Dialog hat die Möglichkeit, im MQXR_SEC_PARMS-Aufruf des Exits die MQCD zu ändern; auf dem Serververbindungsende des Kanals kann diese Funktion hilfreich sein, um beispielsweise die Werte MCAUserIdentifier oder LongMCAUserIdPtr zu ändern, bevor die Verbindung zum Warteschlangenmanager hergestellt wird.

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

*Abbildung 4. Clientverbindung-Eingeganter Austausch mit Vereinbarung für Clientverbindung unter Verwendung von Sicherheitsparametern*

**Anmerkung:** Sicherheitsexitanwendungen, die vor dem Release von IBM WebSphere MQ 7.1 erstellt wurden, müssen möglicherweise aktualisiert werden. Weitere Informationen finden Sie im Abschnitt Kanalsicherheitsexitprogramme.

## ALW API-Aufruf in einem Sende-oder Empfangsexitprogramm identifizieren

Wenn Sie MQI-Kanäle für Clients verwenden, identifiziert Byte 10 des Agentenpuffers den API-Aufruf, der verwendet wird, wenn ein Sende-oder Empfangsexit aufgerufen wird. Dies ist hilfreich, um zu ermitteln, welche Kanalflüsse Benutzerdaten enthalten und möglicherweise eine Verarbeitung wie die Verschlüsselung oder die digitale Signatur erfordern.

In der folgenden Tabelle werden die Daten angezeigt, die in Byte 10 des Kanalflusses angezeigt werden, wenn ein API-Aufruf verarbeitet wird.

**Anmerkung:** Dies sind nicht die einzigen Werte dieses Bytes. Es sind weitere **reservierte** Werte vorhanden.

*Tabelle 8. API-Aufrufe identifizieren*

| API-Aufruf | Wert von Byte 10 für Anforderung | Wert von Byte 10 für Antwort |
|---|---|---|
| MQCONN „1” auf Seite 65, „2” auf Seite 65 | X'81 ' | X' 91 ' |

| *Tabelle 8. API-Aufrufe identifizieren (Forts.)* | | |
|---|---|---|
| **API-Aufruf** | **Wert von Byte 10 für Anforderung** | **Wert von Byte 10 für Antwort** |
| MQDISC „1" auf Seite 65 | X'82 ' | X' 92 ' |
| MQOPEN „3" auf Seite 65 | X'83 ' | X' 93 ' |
| MQCLOSE | X'84 ' | X' 94 ' |
| MQGET „4" auf Seite 65 | X'85 ' | X' 95 ' |
| MQPUT „4" auf Seite 65 | X'86 ' | X' 96 ' |
| MQPUT1-Anforderung „4" auf Seite 65 | X'87 ' | X' 97 ' |
| MQSET-Anforderung | X'88 ' | X' 98 ' |
| MQINQ-Anforderung | X'89 ' | X' 99 ' |
| MQCMIT-Anforderung | X'8A ' | X' 9A ' |
| MQBACK-Anforderung | X'8B ' | X' 9B ' |
| MQSTAT-Anforderung | X'8D ' | X' 9D ' |
| MQSUB-Anforderung | X'8E ' | X' 9E ' |
| MQSUBRQ-Anforderung | X'8F ' | X' 9F ' |
| xa_start-Anforderung | X'A1 ' | X'B1 ' |
| xa_end-Anforderung | X'A2 ' | X'B2 ' |
| xa_open-Anforderung | X'A3 ' | X'B3 ' |
| xa_close-Anforderung | X'A4 ' | X'B4 ' |
| xa_prepare-Anforderung | X'A5 ' | X'B5 ' |
| xa_commit-Anforderung | X'A6 ' | X'B6 ' |
| xa_rollback-Anforderung | X'A7 ' | X'B7 ' |
| Anforderung 'xa_forget' | X'A8 ' | X'B8 ' |
| xa_recover, Anforderung | X'A9 ' | X'B9 ' |
| Anforderung 'xa_abgeschlossen' | X'AA' | X'BA ' |

**Anmerkungen:**

1. Die Verbindung zwischen dem Client und dem Server wird von der Clientanwendung unter Verwendung von MQCONN eingeleitet. Daher gibt es für diesen Befehl vor allem mehrere andere Netzabläufe. Dasselbe gilt für MQDISC, das die Netzverbindung beendet.
2. MQCONNX wird in der gleichen Weise wie MQCONN für die Zwecke der Client/Server-Verbindung behandelt.
3. Wenn eine große Verteilerliste geöffnet wird, kann es mehr als einen Netzfluss pro MQOPEN-Aufruf geben, um alle erforderlichen Daten an den SVRCONN-MCA zu übergeben.
4. Große Nachrichten können die Größe des Übertragungssegments überschreiten. Wenn dies geschieht, kann es viele Netzabläufe geben, die aus einem einzigen API-Aufruf resultieren.

# z/OS Client mit einer Gruppe mit gemeinsamer Warteschlange verbinden

Sie können einen Client mit einer Gruppe mit gemeinsamer Warteschlange verbinden, indem Sie einen MQI-Kanal zwischen einem Client und einem Warteschlangenmanager auf einem Server erstellen, der Mitglied einer Gruppe mit gemeinsamer Warteschlange ist.

## Informationen zu diesem Vorgang

Eine Gruppe mit gemeinsamer Warteschlange wird von einer Gruppe von Warteschlangenmanagern gebildet, die auf dieselbe Gruppe gemeinsam genutzter Warteschlangen zugreifen können. Weitere Informationen zu gemeinsam genutzten Warteschlangen finden Sie unter Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange.

Ein Client, der Nachrichten in eine gemeinsam genutzte Warteschlange einreiht, kann eine Verbindung zu jedem Mitglied der Gruppe mit gemeinsamer Warteschlange herstellen. Die Vorteile einer Verbindung zu einer Gruppe mit gemeinsamer Warteschlange sind eine mögliche Erhöhung der Front-End- und Back-End-Verfügbarkeit sowie eine erhöhte Kapazität. Sie können eine Verbindung zu einem bestimmten WS-Manager oder zu der generischen Schnittstelle herstellen.

Die direkte Verbindung zu einem Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange bietet den Vorteil, dass Sie Nachrichten in eine gemeinsam genutzte Zielwarteschlange einreihen können, wodurch die Back-End-Verfügbarkeit erhöht wird.

Wenn Sie eine Verbindung zur generischen Schnittstelle einer Gruppe mit gemeinsamer Warteschlange herstellen, wird eine Sitzung mit einem der Warteschlangenmanager in der Gruppe geöffnet. Dies erhöht die Front-End-Verfügbarkeit, da der Client-WS-Manager eine Verbindung zu jedem WS-Manager in der Gruppe herstellen kann. Sie stellen über die generische Schnittstelle eine Verbindung zu der Gruppe her, wenn Sie keine Verbindung zu einem bestimmten Warteschlangenmanager innerhalb der Gruppe mit gemeinsamer Warteschlange herstellen möchten.

Die generische Schnittstelle kann eine Sysplex-Distributor-VIPA-Adresse oder ein generischer VTAM-Ressourcenname oder eine andere gemeinsame Schnittstelle für die Gruppe mit gemeinsamer Warteschlange sein. Weitere Informationen zum Einrichten einer generischen Schnittstelle finden Sie im Abschnitt Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten.

## Vorgehensweise

Um eine Verbindung zur generischen Schnittstelle einer Gruppe mit gemeinsamer Warteschlange herzustellen, müssen Sie Kanaldefinitionen erstellen, auf die jeder Warteschlangenmanager in der Gruppe zugreifen kann. Dazu müssen Sie die gleichen Definitionen auf jedem WS-Manager in der Gruppe haben.

1. Definieren Sie den SVRCONN-Kanal wie im folgenden Beispiel gezeigt:

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

   Kanaldefinitionen auf dem Server werden in einem gemeinsam genutzten Db2-Repository gespeichert. Jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange stellt eine lokale Kopie der Definition her, wodurch sichergestellt wird, dass Sie bei der Ausgabe eines MQCONN- oder MQCONNX-Aufrufs immer eine Verbindung zum richtigen Serververbindungskanal herstellen.
2. Definieren Sie den CLNTCONN-Kanal wie im folgenden Beispiel gezeigt:

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

### Ergebnisse

Da die generische Schnittstelle der Gruppe mit gemeinsamer Warteschlange im Feld CONNAME im Clientverbindungskanal gespeichert wird, können Sie jetzt eine Verbindung zu jedem Warteschlangenmanager in der Gruppe herstellen und Nachrichten in gemeinsam genutzte Warteschlangen dieser Gruppe einreihen.

## Verwendung von IBM MQ-Umgebungsvariablen

Sie können Befehle verwenden, um die aktuellen Einstellungen anzuzeigen oder um die Werte von IBM MQ-Umgebungsvariablen zurückzusetzen.

### Informationen zu diesem Vorgang

Sie können Umgebungsvariablen auf die folgenden Arten verwenden:

- So legen Sie die Variablen in Ihrem Systemprofil fest, um eine permanente Änderung vorzunehmen
- Geben Sie einen Befehl in der Befehlszeile ein, um eine Änderung nur für diese Sitzung vorzunehmen.
- Um einem oder mehreren Variablen einen bestimmten Wert in Abhängigkeit von der ausgeführten Anwendung zu geben, fügen Sie Befehle zu einer Befehlsscriptdatei hinzu, die von der Anwendung verwendet wird.

Für jede Umgebungsvariable können Sie mit Befehlen die aktuelle Einstellung anzeigen oder den Wert der Umgebungsvariablen zurücksetzen. Diese Befehle sind auf allen unterstützten Plattformen verfügbar, sofern nicht anders angegeben. Das Format des Befehls hängt von Ihrer Plattform ab. For example:

- Linux AIX Unter AIX and Linux:

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

- Windows Unter Windows:

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

- IBM i Unter IBM i:

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

- MQ Appliance Informationen zu IBM MQ Appliancefinden Sie unter Umgebungsvariablen für IBM MQ Appliance in der IBM MQ Appliance -Dokumentation.

Sofern zutreffend, verwendet IBM MQ Standardwerte für die Umgebungsvariablen, die Sie nicht festgelegt haben.

**Anmerkung:** z/OS IBM MQ for z/OS unterstützt keine IBM MQ-Umgebungsvariablen. Wenn Sie diese Plattform als Ihren Server verwenden, finden Sie in der Definitionstabelle für den Clientkanal Informationen darüber, wie die Definitionstabelle für den Clientkanal in z/OS generiert wird. Sie können weiterhin IBM MQ -Umgebungsvariablen auf Ihrer Clientplattform verwenden.

### Prozedur

- Windows
  Verwenden Sie in Windows für jede Umgebungsvariable die folgenden Befehle, um die aktuelle Einstellung anzuzeigen oder um den Wert einer Variablen zurückzusetzen:
  - Verwenden Sie folgenden Befehl, um den Wert einer Umgebungsvariable zu entfernen:

    ```
    SET MQSERVER=
    ```

  - Verwenden Sie folgenden Befehl, um die aktuelle Einstellung einer Umgebungsvariable anzuzeigen:

```
SET MQSERVER
```

– Verwenden Sie folgenden Befehl, um alle Umgebungsvariablen für die Sitzung anzuzeigen:

```
set
```

- Linux AIX
  Verwenden Sie in AIX and Linux für jede Umgebungsvariable die folgenden Befehle, um die aktuelle Einstellung anzuzeigen oder um den Wert einer Variablen zurückzusetzen:
  – Verwenden Sie folgenden Befehl, um den Wert einer Umgebungsvariable zu entfernen:

```
unset MQSERVER
```

  – Verwenden Sie folgenden Befehl, um die aktuelle Einstellung einer Umgebungsvariable anzuzeigen:

```
echo $MQSERVER
```

  – Verwenden Sie folgenden Befehl, um alle Umgebungsvariablen für die Sitzung anzuzeigen:

```
set
```

**Zugehörige Tasks**

Umgebungsvariablen für IBM MQ classes for JMS/Jakarta Messaging festlegen

Umgebungsvariablen, die für IBM MQ classes for Java relevant sind

Zusätzliche Umgebungsvariablen in der Datei service.env definieren

„IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern” auf Seite 90

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen, indem Sie die Informationen in den Konfigurationsdateien (`.ini`) bearbeiten. Sie können auch Konfigurationsoptionen für IBM MQ MQI clientsändern.

**Zugehörige Verweise**

Verwendung von Umgebungsvariablen in MFT-Eigenschaften

# Beschreibung der Umgebungsvariablen

Beschreibungen der Server-und Clientumgebungsvariablen, die für den Kunden bestimmt sind.

## Verwendungsbeispiele

- Linux AIX Verwenden Sie auf AIX and Linux -Systemen das folgende Format: `export [environment variable]=value`.
- Windows Verwenden Sie auf Windows -Systemen das folgende Format: `Set [environment variable]=value`.
- IBM i Verwenden Sie auf IBM i -Systemen das folgende Format: `ADDENVVAR ENVVAR(environment variable) VALUE(xx)`.
- MQ Appliance Informationen zu IBM MQ Appliancefinden Sie unter Umgebungsvariablen für IBM MQ Appliance in der IBM MQ Appliance -Dokumentation.

## AMQ_ALLOWED_CIPHERS

Multi

Ab IBM MQ 9.2.0können Sie mit der Umgebungsvariablen **AMQ_ALLOWED_CIPHERS** eine angepasste Liste von CipherSpecs angeben, die für die Verwendung mit IBM MQ -Kanälen auf Multiplatforms aktiviert sind. Die Umgebungsvariable verwendet dieselben Werte wie das SSL-Zeilengruppenattribut **AllowedCipherSpecs** der Datei `.ini`:

- Ein einzelner CipherSpec-Name oder
- Eine durch Kommas getrennte Liste von IBM MQ CipherSpec -Namen, die erneut aktiviert werden sollen, oder
- Der Sonderwert ALL, der alle CipherSpecs darstellt (nicht empfohlen).

**Anmerkung:** Die Aktivierung von **ALL**CipherSpecs wird nicht empfohlen, da dadurch SSL 3.0- und TLS 1.0-Protokolle sowie eine große Anzahl schwacher Verschlüsselungsalgorithmen aktiviert werden.

Weitere Informationen finden Sie unter Benutzerdefinierte Liste aktivierter CipherSpecs auf Multiplatforms bereitstellen in der Reihenfolge derCipherSpec im TLS-Handshake.

## AMQ_BAD_COMMS_DATA_FDCS

Die Umgebungsvariable **AMQ_BAD_COMMS_DATA_FDCS** ist wirksam, wenn sie auf einen Wert gesetzt wird.

Wenn die Daten, die IBM MQ von einem Host über TCP/IP empfängt, ein falsches Format haben, z. B. weil ein Netzclient eine Verbindung zu einem IBM MQ -Listener-Port hergestellt und versucht hat, mit einem nicht unterstützten Anwendungsprotokoll zu kommunizieren, schreibt der Warteschlangenmanager eine Fehlernachricht AMQ9207E in die Fehlerprotokolle des Warteschlangenmanagers. IBM MQ -Empfangsprogramme unterstützen TCP/IP-Verbindungen von Nachrichtenkanalagenten (MCAs) des Warteschlangenmanagers sowie von MQI-, JMS -und XMS -Clientanwendungen.

**Anmerkung:** IBM MQ -Empfangsprogramme unterstützen das von AMQP-und MQTT-Clients verwendete Anwendungsprotokoll nicht. Diese Clients sollten stattdessen eine Verbindung zu den Netzports herstellen, die im entsprechenden AMQP-Kanal oder MQXR-Telemetrieservice konfiguriert sind.

Möglicherweise wird auch ein FDC-Datensatz (Failure Data Capture) geschrieben, der die ungültigen Daten enthält, die IBM MQ empfangen hat. Eine FFST-Datei wird jedoch nicht generiert, wenn dies der Anfang eines Dialogs mit der fernen Seite ist und das Format ein einfaches bekanntes Format ist, wie z. B. eine GET-Anforderung von einem HTTP-Web-Browser. Wenn Sie dies überschreiben möchten, damit FFST-Dateien für fehlerhafte Daten geschrieben werden, einschließlich einfacher bekannter Formate, können Sie die Umgebungsvariable **AMQ_BAD_COMMS_DATA_FDCS** auf einen beliebigen Wert (z. B. TRUE) setzen und den Warteschlangenmanager erneut starten.

## AMQ_CONVEBCDICNEWLINE


Mit der Umgebungsvariablen **AMQ_CONVEBCDICNEWLINE** können Sie angeben, wie IBM MQ ein EBCDIC-NL-Zeichen in das ASCII-Format konvertieren soll. Die Umgebungsvariable hat dieselben Werte wie das Attribut **ConvEBCDICNewline** der `mqs.ini`, d. h. `NL_TO_LF`, `TABLE`oder `ISO` (siehe Zeilengruppe für alle Warteschlangenmanager in der Datei mqs.ini). Sie können beispielsweise statt des Zeilengruppenattributs **ConvEBCDICNewline** die Umgebungsvariable **AMQ_CONVEBCDICNEWLINE** verwenden, um **ConvEBCDICNewline**-Funktionalität auf der Clientseite in Situationen bereitzustellen, in denen die Datei `mqs.ini` nicht verwendet werden kann. Wenn sowohl das Zeilengruppenattribut als auch die Umgebungsvariable festgelegt sind, hat das Zeilengruppenattribut Vorrang.

Weitere Informationen finden Sie unter Datenkonvertierung zwischen codierten Zeichensätzen .

## AMQ_DIAGNOSTIC_MSG_SEVERITY

Wenn die Umgebungsvariable **AMQ_DIAGNOSTIC_MSG_SEVERITY** für einen IBM MQ -Prozess auf 1 gesetzt ist, wird die Nachrichtenbewertung als einzelnes alphabetisches Zeichen in Großbuchstaben an die Nachrichtennummer angehängt, wenn der IBM MQ -Prozess eine Nachricht in ein Fehlerprotokoll oder an die Konsole schreibt.

Das Verhalten, das **AMQ_DIAGNOSTIC_MSG_SEVERITY**aktiviert, ist standardmäßig festgelegt. Sie können dieses Verhalten inaktivieren, indem Sie die Umgebungsvariable auf `0`setzen.

Weitere Informationen finden Sie im Abschnitt Fehlerprotokolle verwenden.

## AMQ_DISABLE_CLIENT_AMS

Sie können die Umgebungsvariable **AMQ_DISABLE_CLIENT_AMS** verwenden, um IBM MQ Advanced Message Security (AMS) auf dem Client zu inaktivieren, wenn ein Fehler 2085 (MQRC_UNKNOWN_OBJECT_NAME) gemeldet wird, wenn Sie versuchen, eine Verbindung zu einem WS-Manager von einer früheren Version des Produkts herzustellen, und Sie einen der folgenden Clients verwenden:

- Eine andere Java runtime environment (JRE) als die IBM Java runtime environment (JRE)
- Ein IBM MQ IBM MQ classes for JMS -oder IBM MQ classes for Java -Client

**Anmerkung:** Sie können die Umgebungsvariable **AMQ_DISABLE_CLIENT_AMS** nicht für C-Clients verwenden. Sie müssen stattdessen die Umgebungsvariable **MQS_DISABLE_ALL_INTERCEPT** verwenden.

Weitere Informationen finden Sie unter Advanced Message Security auf dem Client inaktivieren.

## AMQ_DMPMQCFG_QSGDISP_DEFAULT

Die vom Befehl **dmpmqcfg** verwendeten Abfragen zur Disposition eines Warteschlangenmanagers fragen standardmäßig nur QSGDISP (QMGR) -Definitionen ab. Sie können zusätzliche Definitionen abfragen, indem Sie die Umgebungsvariable **AMQ_DMPMQCFG_QSGDISP_DEFAULT** verwenden, die auf einen der folgenden Werte gesetzt werden kann:

**LIVE**
Nur Objekte einschließen, die mit QSGDISP(QMGR) oder QSGDISP(COPY) definiert wurden.

**ALLE**
Objekte einschließen, die mit QSGDISP(QMGR) und QSGDISP(COPY) definiert wurden. Wenn der Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, werden auch QSGDISP(GROUP) und QSGDISP(SHARED) eingeschlossen.

**KOPIEREN**
Nur Objekte einschließen, die mit QSGDISP(COPY) definiert wurden.

**GRUPPE**
Nur Objekte einschließen, die mit QSGDISP(GROUP) definiert wurden. Der Zielwarteschlangenmanager muss Mitglied einer Gruppe mit gemeinsamer Warteschlange sein.

**QMGR**
Nur Objekte einschließen, die mit QSGDISP(QMGR) definiert wurden. Dies ist das Standardverhalten, wenn Sie diese Umgebungsvariable zwecks Übereinstimmung mit dem bestehenden Verhalten von **dmpmqcfg** verwenden.

**PRIVATE**
Nur Objekte einschließen, die mit QSGDISP(QMGR) oder QSGDISP(COPY) definiert wurden.

**SHARED**
Nur Objekte einschließen, die mit QSGDISP(SHARED) definiert wurden.

## AMQ_IODELAY, AMQ_IODELAY_INMS und AMQ_IODELAY_FFST


IBM MQ erkennt, wenn Lese-und Schreiboperationen für Protokolle oder Eingabe-und Ausgabeoperationen länger als erwartet dauern. Dies kann auf Probleme mit dem Betriebssystem oder dem Speichersystem zurückzuführen sein und die Leistung des Warteschlangenmanagers beeinträchtigen. Ab IBM MQ 9.3.4können Sie die **AMQ_IODELAY** -Umgebungsvariablen verwenden, um die Diagnose und die Ablaufsteuerungen zu optimieren, wenn die Eingabe und Ausgabe für das Protokoll und das Speicherdateisystem Ihres Warteschlangenmanagers langsam ist. Wenn die Nachricht AMQ6729W `Log I/O operation exceeded threshold` im Fehlerprotokoll des Warteschlangenmanagers angezeigt wird, untersuchen Sie die Ursache und nehmen Sie entsprechende Anpassungen vor. Verwenden Sie die Variablen wie in den folgenden Beispielen gezeigt:

**AMQ_IODELAYConstellation name (optional)**
Schwellenwertzeit in Sekunden. Der Standardwert ist 1 Sekunde. Wenn eine E/A-Operation länger als dieser Schwellenwert dauert, wird die Fehlernachricht AMQ6729W in den IBM MQ -Protokolldateien

gemeldet. Die Warnung wird höchstens alle 10 Sekunden wiederholt, wenn die Verzögerungen bestehen bleiben. Sie können diesen Wert erhöhen, um Fehler zu unterdrücken oder zu verringern, um bestimmte Leistungsprobleme zu untersuchen. Beispiel:

```
export AMQ_IODELAY=200000
```

**AMQ_IODELAY_INMS**
Ändern Sie die Zeitmessung in Mikrosekunden statt in Sekunden. Mit dieser Option können Sie einen unteren Schwellenwert festlegen, bevor Sie die Nachricht AMQ6729 im Warteschlangenmanagerprotokoll erhalten.

```
export AMQ_IODELAY_INMS=YES
```

**AMQ_IODELAY_FFST**
Zusätzlich zur Warnung im Fehlerprotokoll wird eine FFST-Datei mit Diagnoseinformationen generiert, wenn der Schwellenwert überschritten wird.

```
export AMQ_IODELAY_FFST=YES
```

Wird der Warteschlangenmanager wie in diesem Beispiel gestartet, wird eine FDC-oder FFST-Datei geschrieben, wenn eine Ein-/Ausgabeoperation länger als 200.000 Mikrosekunden (0.2s) dauert, was noch ein relativ großzügiger Schwellenwert ist.

Weitere Informationen finden Sie unter Statusprüfungsverhalten des Warteschlangenmanagers.

## AMQ_LDAP_TRACE

Wenn die Umgebungsvariable **AMQ_LDAP_TRACE** auf einen Wert ungleich null gesetzt ist, ist es möglich, den LDAP-Client-Trace ein-bzw. auszuschalten, ohne auch den Warteschlangenmanager zu stoppen oder zu starten.

Weitere Informationen finden Sie im Abschnitt Dynamisches Tracing für LDAP-Clientbibliothekscode aktivieren.

## AMQ_LICENSING_METRIC


Das Festlegen der Umgebungsvariablen **AMQ_LICENSING_METRIC=VPCMonthlyPeak** bewirkt, dass der Warteschlangenmanager Daten hochlädt, die sich auf monatliche VPC-Lizenztypen beziehen, anstatt das Standardverhalten des Hochladens von Daten zu stündlichen containerbasierten Lizenzen zu verwenden.

Weitere Informationen zum Konfigurieren von IBM MQ für die Verwendung mit dem Messservice IBM Cloud Private finden Sie unter IBM Cloud Private -Messservice in der IBM Cloud Private -Dokumentation.

## AMQ_MQS_INI_LOCATION


Auf AIX and Linux -Systemen können Sie die Position ändern, die für die Datei `mqs.ini` verwendet wird, indem Sie die Position der Datei `mqs.ini` in der Umgebungsvariable **AMQ_MQS_INI_LOCATION** festlegen. Diese Umgebungsvariable muss auf Systemebene festgelegt werden.

Weitere Informationen zur Datei `mqs.ini` , einschließlich der Verzeichnispositionen, finden Sie im Abschnitt IBM MQ -Konfigurationsdatei mqs.ini.

## AMQ_NO_BAD_COMMS_DATA_FDCS

Die Umgebungsvariable **AMQ_NO_BAD_COMMS_DATA_FDCS** ist wirksam, wenn sie auf einen Wert gesetzt wird.

Wenn IBM MQ die ursprüngliche Datenübertragung nicht erkennt, wenn versucht wird, einen Nicht-IBM MQ -Client mit einem IBM MQ -TCP/IP-Listener zu verbinden, führt dies dazu, dass der Warteschlangen-

manager eine AMQ9207E -Fehlernachricht in die Fehlerprotokolle des Warteschlangenmanagers schreibt. Ein FDC-Datensatz (FDC = Failure Data Capture) wird ebenfalls geschrieben. Sie können die Generierung dieser Diagnosedateien mit der Umgebungsvariablen **AMQ_NO_BAD_COMMS_DATA_FDCS** unterdrücken. Wenn **AMQ_NO_BAD_COMMS_DATA_FDCS** auf einen beliebigen Wert gesetzt ist (z. B. TRUE), weist dies IBM MQ an, keine FFSTs zu generieren, wenn Fehlernachrichten AMQ9207E im ursprünglichen Kommunikationsfluss gemeldet werden. Um wirksam zu sein, sollte die Umgebungsvariable vor dem Start des Warteschlangenmanagers und der Listenerprozesse gesetzt werden.

Die FDC wird weiterhin generiert, wenn ein Client gültige IBM MQ -Protokolldatenflüsse an den Warteschlangenmanager sendet und dann ungültige Daten sendet, da dies auf ein Clientproblem hinweist, das eine weitere Untersuchung erfordert.

**Anmerkung:** Ab IBM MQ 9.2.0wird die Erfassung von FFSTs bei der Meldung von AMQ9207E -Fehlernachrichten in anfänglichen Kommunikationsflüssen standardmäßig unterdrückt.

## AMQ_NO_IPV6

Die Umgebungsvariable **AMQ_NO_IPV6** ist wirksam, wenn sie auf einen Wert gesetzt wird. Wenn diese Umgebungsvariable gesetzt ist, inaktiviert sie die Verwendung von IPv6 beim Versuch, eine Verbindung herzustellen.

## AMQ_REVERSE_COMMIT_ORDER

Die Umgebungsvariable **AMQ_REVERSE_COMMIT_ORDER** konfiguriert einen Warteschlangenmanager so, dass in einer XA-Transaktion die Änderung des IBM MQ -Warteschlangenmanagers festgeschrieben wird, sobald die entsprechende Datenbankaktualisierung abgeschlossen ist. Anwendungen, die Nachrichten aus den Warteschlangen lesen, sehen eine Nachricht erst nach Abschluss der entsprechenden Datenbankaktualisierung.

**Anmerkung:** Legen Sie **AMQ_REVERSE_COMMIT_ORDER** nicht fest, ohne das unter Isolationsstufebeschriebene Szenario zu lesen und zu verstehen.

## AMQ_SSL_ALLOW_DEFAULT_CERT

Wenn die Umgebungsvariable **AMQ_SSL_ALLOW_DEFAULT_CERT** nicht festgelegt ist, kann eine Anwendung eine Verbindung zu einem Warteschlangenmanager mit einem persönlichen Zertifikat im Client-Keystore nur herstellen, wenn das Zertifikat den Kennsatznamen `ibmwebspheremq`*`userid`*enthält. Wenn die Umgebungsvariable **AMQ_SSL_ALLOW_DEFAULT_CERT** festgelegt ist, erfordert das Zertifikat nicht den Kennsatznamen `ibmwebspheremq`*`userid`*. Das heißt, das Zertifikat, das für die Verbindung zu einem Warteschlangenmanager verwendet wird, kann ein Standardzertifikat sein, sofern ein Standardzertifikat im Schlüsselrepository vorhanden ist und das Schlüsselrepository kein persönliches Zertifikat mit dem Präfix `ibmwebspheremq`*`userid`*enthält.

Der Wert 1 lässt die Verwendung eines Standardzertifikats zu.

Anstelle der Umgebungsvariablen **AMQ_SSL_ALLOW_DEFAULT_CERT** kann eine Anwendung die Einstellung **CertificateLabel** der SSL-Zeilengruppe in der Datei `mqclient.ini` verwenden. Weitere Informationen finden Sie in den Abschnitten Digitale Zertifikatsbezeichnungen - Anforderungen und SSL-Zeilengruppe in der Clientkonfigurationsdatei.

## AMQ_SSL_LDAP_SERVER_VERSION

Die Umgebungsvariable **AMQ_SSL_LDAP_SERVER_VERSION** kann verwendet werden, um sicherzustellen, dass LDAP v2 oder LDAP v3 von IBM MQ Verschlüsselungskomponenten verwendet wird, wenn CRL-Server erfordern, dass eine bestimmte Version des LDAP-Protokolls verwendet wird.

Setzen Sie die Umgebungsvariable auf den entsprechenden Wert in der Umgebung, die zum Starten des Warteschlangenmanagers oder Kanals verwendet wird:

- Legen Sie AMQ_SSL_LDAP_SERVER_VERSION=2fest, um die Verwendung von LDAP v2 anzufordern.
- Legen Sie AMQ_SSL_LDAP_SERVER_VERSION=3 fest, um die Verwendung von LDAP v3 anzufordern.

Diese Umgebungsvariable wirkt sich nicht auf LDAP-Verbindungen aus, die vom IBM MQ -Warteschlangenmanager für die Benutzerauthentifizierung oder Benutzerberechtigung eingerichtet wurden.

## AMQ_USE_ZLIBNX

V 9.3.0 AIX

Unter AIXkann die Umgebungsvariable **AMQ_USE_ZLIBNX** verwendet werden, um Nachrichtenkanalagenten (MCA) die Verwendung der hardwarebeschleunigten Bibliothek zlibNX für die Komprimierung und Dekomprimierung von Nachrichtendaten zu ermöglichen, wenn ZLIBFAST-oder ZLIBHIGH-Verfahren verwendet werden.

**Tipp:** Hoch komprimierbare Nachrichten mit einer Größe von mehr als 2 KB profitieren höchstwahrscheinlich von der Entscheidung, die Bibliothek zlibNX zu verwenden, indem sie die CPU-Auslastung reduzieren.

Die Bibliothek zlibNX ist in IBM AIX 7.2 mit Technology Level 4 Expansion Pack und höher verfügbar. Wenn die Umgebungsvariable gesetzt ist, die Bibliothek zlibNX (`/usr/opt/zlibNX/lib/libz.a`) aber nicht installiert ist, verwenden die Nachrichtenkanalagenten die in der Installation von IBM MQ for AIX bereitgestellte Standardbibliothek zlib.

## HOME

Linux IBM i AIX

Unter AIX, Linux und IBM igibt die Umgebungsvariable **HOME** den Namen des Verzeichnisses an, das nach der Datei `mqclient.ini` durchsucht wird. Diese Datei enthält Konfigurationsdaten, die von IBM MQ MQI clientsverwendet werden.

Weitere Informationen finden Sie unter IBM MQ MQI-Clientkonfigurationsdatei mqclient.ini und Position der Clientkonfigurationsdatei.

## HOMEDRIVE und HOMEPATH

Windows

Für die Verwendung müssen die Umgebungsvariablen **HOMEDRIVE** und **HOMEPATH** festgelegt werden. Sie werden auf Windows -Systemen verwendet, um den Namen des Verzeichnisses anzugeben, das nach der Datei `mqclient.ini` durchsucht wird. Diese Datei enthält Konfigurationsdaten, die von IBM MQ MQI clientsverwendet werden.

Weitere Informationen finden Sie unter IBM MQ MQI-Clientkonfigurationsdatei mqclient.ini und Position der Clientkonfigurationsdatei.

## LDAP_BASEDN

**LDAP_BASEDN** ist die erforderliche Umgebungsvariable für die Ausführung eines LDAP-Beispielprogramms. Sie gibt den Basis-DN für die Verzeichnissuche an.

## LDAP_HOST

**LDAP_HOST** ist eine optionale Umgebungsvariable für die Ausführung eines LDAP-Beispielprogramms. Sie gibt den Namen des Hosts an, auf dem der LDAP-Server ausgeführt wird. Wenn keine Angabe erfolgt, wird als Wert standardmäßig der lokale Host verwendet.

## LDAP_VERSION

**LDAP_VERSION** ist eine optionale Umgebungsvariable für die Ausführung eines LDAP-Beispielprogramms. Sie gibt die Version des zu verwendenden LDAP-Protokolls an und kann entweder 2 oder 3 sein. Die meisten LDAP-Server unterstützen nun Version 3 des Protokolls; die ältere Version 2 wird von allen LDAP-Servern unterstützt. Dieses Beispiel funktioniert mit beiden Protokollversionen gleich gut und es wird standardmäßig Version 2 verwendet, falls nichts anderes angegeben ist.

## MQ_CHANNEL_SUPPRESS_INTERVAL

Die Umgebungsvariable **MQ_CHANNEL_SUPPRESS_INTERVAL** gibt das Zeitintervall (in Sekunden) an, in dem die in **MQ_CHANNEL_SUPPRESS_MSGS** definierten Nachrichten unterdrückt werden sollen, damit sie in das Fehlerprotokoll geschrieben werden, sowie die Häufigkeit, mit der eine Nachricht während des angegebenen Zeitintervalls auftreten darf, bevor sie unterdrückt wird. Der Standardwert ist 60,5, was bedeutet, dass alle weiteren Vorkommen einer bestimmten Nachricht nach den ersten fünf Vorkommen dieser Nachricht in einem 60-Sekunden-Intervall unterdrückt werden. Weitere Informationen finden Sie im Abschnitt Kanalfehlernachrichten in Fehlerprotokollen auf Multiplatforms unterdrücken.

Die Umgebungsvariable **MQ_CHANNEL_SUPPRESS_INTERVAL** ist vergleichbar mit SuppressInterval in der Datei `qm.ini` .

## MQ_CHANNEL_SUPPRESS_MSGS

Die Umgebungsvariable **MQ_CHANNEL_SUPPRESS_MSGS** unterdrückt Kanalfehlernachrichten im Fehlerprotokoll. Sie können eine Liste der Nachrichten angeben, die unterdrückt werden sollen. **MQ_CHANNEL_SUPPRESS_MSGS** wird zusammen mit **MQ_CHANNEL_SUPPRESS_INTERVAL**verwendet. Dieses Attribut gibt an, wie oft jede Nachricht angezeigt wird, bevor sie unterdrückt wird, und wie lange Nachrichten unterdrückt werden. Weitere Informationen finden Sie im Abschnitt Kanalfehlernachrichten in Fehlerprotokollen auf Multiplatforms unterdrücken.

Die Umgebungsvariable **MQ_CHANNEL_SUPPRESS_MSGS** ist vergleichbar mit SuppressMessage in der Datei `qm.ini` , außer dass Sie jede Kanalnachricht unterdrücken können, indem Sie die Umgebungsvariable verwenden, während es eine einschränkende Liste für die Methode `qm.ini` gibt.

## MQ_CONNECT_TYPE


Auf Multiplatforms können Sie die Umgebungsvariable **MQ_CONNECT_TYPE** in Kombination mit dem Bindungstyp verwenden, der im Feld 'Optionen' der MQCNO-Struktur angegeben ist, die in einem MQCONNX-Aufruf verwendet wird. **MQ_CONNECT_TYPE** hat nur Auswirkungen auf STANDARD-Bindungen. Bei anderen Bindungen wird **MQ_CONNECT_TYPE** ignoriert.

Weitere Informationen finden Sie unter MQCONNX-Aufrufoptionen mit MQ_CONNECT_TYPE verwenden.

## MQ_CROSS_QUEUE_ORDER_ALL

Wenn Sie die Umgebungsvariable **MQ_CROSS_QUEUE_ORDER_ALL** auf einen Wert ungleich null setzen, wird die Reihenfolge der Nachrichteneinreihung in einer Arbeitseinheit beibehalten. Dies bedeutet Folgendes: Wenn Nachrichten in einer Arbeitseinheit (UOW) in mehrere Warteschlangen eingereiht werden (beispielsweise in Q1 und danach in Q2) werden sie bei einer MQCMIT-Ausgabe in der Reihenfolge übermittelt und zur Verfügung gestellt, in der sie EINGEREIHT wurden.

In einer Umgebung mit mehreren Warteschlangenmanagern muss **MQ_CROSS_QUEUE_ORDER_ALL** vorhanden sein und auf der sendenden und empfangenden Seite einen nicht leeren Wert haben, bevor jeder Warteschlangenmanager gestartet wird.

## MQ_EPHEMERAL_PREFIX

Die Umgebungsvariable **MQ_EPHEMERAL_PREFIX** gibt den Pfad zum ephemeren Verzeichnis des Warteschlangenmanagers an, in dem ephemere Warteschlangenmanagerdaten aufbewahrt werden, während der Warteschlangenmanager aktiv ist.

Als Alternative zum Ändern des ephemeren Präfix durch Ändern des Attributs **EphemeralPrefix** im Attribut **DefaultEphemeralPrefix** der Zeilengruppe AllQueueManagers in der Datei `mqs.ini` können Sie die Umgebungsvariable **MQ_EPHEMERAL_PREFIX** verwenden, um die **EphemeralPrefix** für den Befehl **crtmqm** zu überschreiben. Weitere Informationen finden Sie im Abschnitt Konfigurierbares ephemeres Verzeichnis.

## MQ_FILE_PATH

Windows

Die Umgebungsvariable **MQ_FILE_PATH** wird während der Installation des Laufzeitpakets auf der Plattform Windows konfiguriert. Diese Umgebungsvariable enthält die gleichen Daten wie der folgende Schlüssel in der Windows-Registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

Weitere Informationen finden Sie unter setmqenv (set IBM MQ environment) und crtmqenv (create IBM MQ environment).

## MQ_JAVA_DATA_PATH

Die Umgebungsvariable **MQ_JAVA_DATA_PATH** gibt das Verzeichnis für die Protokoll-und Traceausgabe für IBM MQ classes for JMS und IBM MQ classes for Jakarta Messaging und IBM MQ classes for Javaan. Es wird von den Scripts verwendet, die mit IBM MQ classes for JMS und IBM MQ classes for Jakarta Messaging und IBM MQ classes for Javabereitgestellt werden.

Weitere Informationen finden Sie unter Setting environment variables for IBM MQ classes for JMS/Jakarta Messaging und Environment variables relevant for IBM MQ classes for Java.

## MQ_JAVA_INSTALL_PATH

Die Umgebungsvariable **MQ_JAVA_INSTALL_PATH** gibt das Verzeichnis an, in dem IBM MQ classes for JMS und IBM MQ classes for Jakarta Messaging installiert sind (siehe Was ist für IBM MQ Classes for JMS installiert), sowie das Verzeichnis, in dem IBM MQ classes for Java installiert ist (siehe IBM MQ classes for Java -Installationsverzeichnisse ).

Weitere Informationen finden Sie unter Setting environment variables for IBM MQ classes for JMS/Jakarta Messaging und Environment variables relevant for IBM MQ classes for Java.

## MQ_JAVA_LIB_PATH

Die Umgebungsvariable **MQ_JAVA_LIB_PATH** gibt das Verzeichnis an, in dem IBM MQ classes for JMS und IBM MQ classes for Jakarta Messagingsowie die IBM MQ classes for Java -Bibliotheken gespeichert sind. Einige Scripts, z. B. IVTRun, die mit IBM MQ classes for JMS und IBM MQ classes for Jakarta Messaging oder IBM MQ classes for Java bereitgestellt werden, verwenden diese Umgebungsvariable.

Weitere Informationen finden Sie unter Setting environment variables for IBM MQ classes for JMS/Jakarta Messaging und Environment variables relevant for IBM MQ classes for Java.

## MQ_OVERRIDE_DATA_PFAD

Sie können die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** verwenden, um das Standardverzeichnis des IBM MQ -Datenpfads zu ändern.

## MQ_SET_NODELAYACK

AIX

Die Umgebungsvariable **MQ_SET_NODELAYACK** inaktiviert die verzögerte TCP-Bestätigung unter AIX.

Wenn Sie diese Umgebungsvariable festlegen, schaltet die Einstellung die verzögerte TCP-Bestätigung ab, indem sie den Aufruf 'setsockopt' des Betriebssystems mit der Option TCP_NODELAYACK aufruft. Diese Funktion wird nur von AIX unterstützt, daher wirkt sich die Umgebungsvariable **MQ_SET_NODELAYACK** nur auf AIXaus.

## MQ-BENUTZERNAME

Linux

Sie können die Umgebungsvariable **MQ_USER_NAME** verwenden, damit eine nicht registrierte Installation unter Linux den Namen eines nicht benannten Benutzers auswählen kann. Dies ist beispielsweise für die Verwendung von Publish/Subscribe-Hierarchien in OpenShifterforderlich.

Der Wert für **MQ_USER_NAME** darf keinem bereits vorhandenen Benutzer auf dem System entsprechen und muss kleiner-gleich 12 Byte sein.

## MQAPI_TRACE_LOGFILE

Das API-Exitprogramm generiert einen MQI-Trace in einer benutzerdefinierten Datei mit einem Präfix, das in der Umgebungsvariablen **MQAPI_TRACE_LOGFILE** definiert ist.

Weitere Informationen hierzu finden Sie im Abschnitt Beispielprogramm für API-Exits.

## MQAPPLNAME

ALW

Wenn der Anwendungsname noch nicht ausgewählt wurde, können Sie die Umgebungsvariable **MQAPPL-NAME** als Namen für die Identifizierung der Verbindung zu einem Warteschlangenmanager verwenden. Es werden nur die ersten 28 Zeichen verwendet, die nicht alle Leerzeichen oder Nullen sein dürfen.

Weitere Informationen finden Sie unter Anwendungsnamen in unterstützten Programmiersprachen verwenden.

## MQCCSID

Die Umgebungsvariable **MQCCSID** gibt die zu verwendende Nummer des codierten Zeichensatzes an und überschreibt den CCSID-Wert, mit dem der Server konfiguriert wurde. **MQCCSID** kann verwendet werden, um die native CCSID einer Anwendung zu überschreiben und die zu verwendende Nummer des codierten Zeichensatzes anzugeben, z. B. wenn die native CCSID eine nicht unterstützte CCSID ist oder nicht die erforderliche CCSID ist.

Verwenden Sie einen der folgenden Befehle, um **MQCCSID**festzulegen:

- Linux AIX Unter AIX and Linux:

```
export MQCCSID=number
```

- Windows Unter Windows:

```
SET MQCCSID=number
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

Weitere Informationen finden Sie unter Client-oder Server-CCSID auswählen.

## MQCCDTURL

Die Umgebungsvariable **MQCCDTURL** bietet die funktional entsprechende Funktionalität zum Festlegen einer Kombination der Umgebungsvariablen **MQCHLLIB** und **MQCHLTAB** . Es ermöglicht Ihnen, eine Datei-, FTP-oder HTTP-URL als einzelnen Wert anzugeben, aus dem eine Clientkanaldefinitionstabelle für native Programme abgerufen werden kann, die eine Verbindung als Clients herstellen, d. h. C-, COBOL-oder C + + -Anwendungen.

**Anmerkung:** Die Verwendung von Umgebungsvariablen zur Bereitstellung der URL hat keine Auswirkungen auf Java, JMS oder verwaltete .NET Anwendungen.

IBM MQ unterstützt das Abrufen einer CCDT aus einer Datei, über eine FTP- oder HTTP-URL. **MQCCDTURL** akzeptiert jedoch nur einen URL-Wert. Das vorhandene Verzeichnisformat des lokalen Dateisystems wird nicht akzeptiert.

Wenn Sie **MQCCDTURL** anstelle von **MQCHLLIB** und **MQCHLTAB** mit einer lokalen Datei verwenden möchten, können Sie ein `'file:///'` -Protokoll verwenden. Daher wie in diesem Beispiel für AIX und Linuxgezeigt:

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

ist äquivalent zu:

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

Sie können auch eine JSON-Datei wie im folgenden Beispiel für Windowsangeben:

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```

ist äquivalent zu:

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

Weitere Informationen finden Sie unter URL-Zugriff auf die CCDT.

## MQCERTLABL

Die Umgebungsvariable **MQCERTLABL** definiert die Zertifikatsbezeichnung einer Kanaldefinition, die IBM MQ zum Lokalisieren eines persönlichen Zertifikats verwendet, das während eines TLS-Handshakes gesendet wird.

Weitere Informationen finden Sie unter Bezeichnungen für digitale Zertifikate-Voraussetzungen.

## MQCERTVPOL

Die Umgebungsvariable **MQCERTVPOL** gibt den Typ der zu verwendenden Zertifikatsprüfrichtlinie an. Diese Umgebungsvariable überschreibt das Attribut **CertificateValPolicy** in der SSL-Zeilengruppe der Clientkonfigurationsdatei.

**MQCERTVPOL** kann auf einen von zwei Werten gesetzt werden:

**ANY**
Es wird eine Zertifikatsprüfrichtlinie verwendet, die von der zugrunde liegenden Secure-Sockets-Bibliothek unterstützt wird. Dies ist die Standardeinstellung.

**RFC5280**
Es wird nur eine Zertifikatsprüfung verwendet, die dem Standard RFC 5280 entspricht.

Verwenden Sie einen der folgenden Befehle, um **MQCERTVPOL**festzulegen:

- Linux AIX Für AIX and Linux-Systeme:

  ```
  export MQCERTVPOL= value
  ```

- Windows Für Windows-Systeme:

  ```
  SET MQCERTVPOL= value
  ```

- IBM i Für IBM i-Systeme:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

Weitere Informationen finden Sie unter Certificate validation policies in IBM MQ und Configuring certificate validation policies in IBM MQ.

## MQCHLLIB

Die Umgebungsvariable **MQCHLLIB** gibt den Verzeichnispfad zu der Datei an, die die Definitionstabelle für den Clientkanal (CCDT) enthält. Die Datei wird auf dem Server erstellt, kann aber auf die IBM MQ MQI client-Workstation kopiert werden.

Verwenden Sie einen der folgenden Befehle, um **MQCHLLIB**festzulegen:

- Windows Unter Windows:

```
SET MQCHLLIB=pathname
```

  For example:

```
SET MQCHLLIB=C:\wmqtest
```

- Linux AIX Für AIX and Linux-Systeme:

```
export MQCHLLIB=pathname
```

- IBM i Für IBM i:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

Wenn **MQCHLLIB** nicht festgelegt ist, nimmt der Pfad für den Client standardmäßig die folgenden Werte an:

- Linux AIX Unter AIX and Linux: `/var/mqm/`
- Windows Unter Windows: *MQ_INSTALLATION_PATH*
- IBM i Unter IBM i: `/QIBM/UserData/mqm/`

Für die Befehle **crtmqm** und **strmqm** wird standardmäßig einer von zwei Pfadgruppen verwendet. Wenn *datapath* festgelegt ist, wird der Pfad standardmäßig auf eine der ersten festgelegten Werte gesetzt. Wenn *datapath* nicht festgelegt ist, wird der Pfad standardmäßig auf einen der zweiten Gruppe gesetzt.

- Linux AIX Unter AIX and Linux: *datapath*`/@ipcc`
- Windows Unter Windows: *datapath*`\@ipcc`
- IBM i Unter IBM i: *datapath*`/&ipcc`

Oder:

- Linux AIX Unter AIX and Linux: `/prefix/qmgrs/`*qmgrname*`/@ipcc`
- Windows Unter Windows: *MQ_INSTALLATION_PATH*`\data\qmgrs\`*qmgrname*`\@ipcc`
- IBM i Unter IBM i: `/`*prefix*`/qmgrs/`*qmgrname*`/&ipcc`

Dabei gilt:

- *MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.
- Falls vorhanden, ist *datapath* der Wert von `DataPath` , der in der Zeilengruppe des Warteschlangenmanagers definiert ist.
- *prefix* ist der Wert von Prefix, der in der Zeilengruppe des Warteschlangenmanagers definiert ist. Präfix ist normalerweise einer der folgenden Werte:
  - Linux AIX `/var/mqm` auf AIX and Linux-Systemen.
  - IBM i `/QIBM/UserData/mqm/` unter IBM i.
- *qmgrname* ist der Wert des `Directory` -Attributs, das in der Zeilengruppe des Warteschlangenmanagers definiert ist. Der Wert kann sich von dem Namen des tatsächlichen Warteschlangenmanagers unterscheiden. Der Wert wurde möglicherweise geändert, um Sonderzeichen zu ersetzen.
- Wo die Zeilengruppe des Warteschlangenmanagers definiert ist, hängt von der Plattform ab:
  - Linux IBM i AIX In der `mqs.ini`-Datei unter IBM i, AIX and Linux.
  - Windows In der Registry unter Windows.

**Anmerkungen:**

1. z/OS Wenn Sie IBM MQ for z/OS als Server verwenden, muss die Datei auf der IBM MQ-Client-Workstation gespeichert werden.
2. Wenn diese Option festgelegt ist, überschreibt `MQCHLLIB` den Pfad, der zum Lokalisieren der CCDT verwendet wird.
3. `MQCHLLIB` kann eine URL enthalten, die in Kombination mit der Umgebungsvariablen MQCHLTAB funktioniert (siehe „URL-Zugriff auf die CCDT” auf Seite 56).
4. Umgebungsvariablen, wie z. B. **MQCHLLIB**, können auf eine plattformspezifische Weise auf einen Prozess oder einen Job oder systemweit zugeordnet werden.
5. Wenn Sie **MQCHLLIB** systemweit auf einem Server definieren, wird derselbe Pfad zur CCDT-Datei für alle Warteschlangenmanager auf dem Server festgelegt. Wenn Sie die Umgebungsvariable **MQCHLLIB** nicht festlegen, ist der Pfad für jeden Warteschlangenmanager unterschiedlich. Warteschlangenmanager lesen den Wert von **MQCHLLIB**im Befehl **crtmqm** oder **strmqm** , wenn er festgelegt ist.
6. Wenn Sie mehrere WS-Manager auf einem einzigen Server erstellen, ist die Unterscheidung wichtig, aus folgendem Grund: Wenn Sie **MQCHLLIB** systemweit definieren, aktualisiert jeder Warteschlangenmanager die gleiche CCDT-Datei. Die Datei enthält die Clientverbindungsdefinitionen von allen Warteschlangenmanagern auf dem Server. Wenn die gleiche Definition auf mehreren Warteschlangenmanagern vorhanden ist, z. B. `SYSTEM.DEF.CLNTCONN` , enthält die Datei die letzte Definition. Wenn Sie einen Warteschlangenmanager erstellen und **MQCHLLIB** festgelegt ist, wird `SYSTEM.DEF.CLNTCONN` in der CCDT aktualisiert. Die Aktualisierung überschreibt den `SYSTEM.DEF.CLNTCONN` , der von einem anderen WS-Manager erstellt wurde. Wenn Sie die frühere Definition geändert haben, gehen Ihre Änderungen verloren. Aus diesem Grund müssen Sie nach Alternativen suchen, um **MQCHLLIB** als systemweite Umgebungsvariable auf dem Server zu definieren.
7. Die Option MQSC und PCF `NOREPLACE` in einer Clientverbindungsdefinition prüft den Inhalt der CCDT-Datei nicht. Eine Clientverbindungskanaldefinition mit demselben Namen, die zuvor erstellt wurde, jedoch nicht von diesem Warteschlangenmanager, wird unabhängig von der Option `NOREPLACE` ersetzt. Wenn die Definition zuvor von demselben WS-Manager erstellt wurde, wird die Definition nicht ersetzt.
8. Der Befehl **rcrmqobj** `-t clchltab` löscht die CCDT-Datei und erstellt sie erneut. Die Datei wird mit nur den Clientverbindungsdefinitionen, die auf dem Warteschlangenmanager erstellt wurden, für den der Befehl ausgeführt wird, erneut erstellt.
9. Andere Befehle, die die CCDT aktualisieren, ändern nur die Clientverbindungskanäle, die denselben Kanalnamen haben. Andere Clientverbindungskanäle in der Datei werden nicht geändert.
10. Der Pfad für **MQCHLLIB** erfordert keine Anführungszeichen.

Weitere Informationen finden Sie unter Positionen für die CCDT, URL-Zugriff auf die CCDTund Clientanwendungen mithilfe von Umgebungsvariablen mit Warteschlangenmanagern verbinden.

## MQCHLTAB

Die Umgebungsvariable **MQCHLTAB** gibt den Namen der Datei an, die die Definitionstabelle für den Clientkanal (CCDT) enthält. Der Standarddateiname lautet AMQCLCHL.TAB.

Verwenden Sie einen der folgenden Befehle, um **MQCHLTAB**festzulegen:

- Linux AIX Unter AIX and Linux:

```
export MQCHLTAB=filename
```

- Windows Unter Windows:

```
SET MQCHLTAB=filename
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

For example:

```
SET MQCHLTAB=ccdf1.tab
```

Auf dieselbe Weise wie für den Client gibt die Umgebungsvariable **MQCHLTAB** auf dem Server den Namen der Clientkanaldefinitionstabelle an.

Weitere Informationen finden Sie unter Positionen für die CCDT, URL-Zugriff auf die CCDTund Clientanwendungen mithilfe von Umgebungsvariablen mit Warteschlangenmanagern verbinden.

## MQCLNTCF

Die Umgebungsvariable **MQCLNTCF** gibt die Position der Konfigurationsdatei IBM MQ MQI client an. Diese Datei enthält Konfigurationsdaten, die von IBM MQ MQI clientsverwendet werden.

Mit der Umgebungsvariablen **MQCLNTCF** können Sie den Dateipfad der Datei `mqclient.ini` ändern.

Das Format dieser Umgebungsvariablen ist eine vollständige URL. Dies bedeutet, dass der Dateiname möglicherweise nicht unbedingt `mqclient.ini`lautet, was die Platzierung der Datei in einem Network Attached File System erleichtert. Weitere Informationen finden Sie unter IBM MQ MQI-Clientkonfigurationsdatei mqclient.ini und Position der Clientkonfigurationsdatei.

## MQDOTNET_TRACE_ON

Die Umgebungsvariable **MQDOTNET_TRACE_ON** wird verwendet, um den Trace für weiterverteilbare IBM MQ .NET -Clients zu ermöglichen. Bei Werten kleiner und gleich 0 wird der Trace nicht aktiviert, bei Wert 1 wird der Standardtrace aktiviert und bei Werten größer als 1 wird der Detailtrace aktiviert.

Weitere Informationen finden Sie unter Tracing IBM MQ .NET applications und Tracing IBM MQ .NET applications using environment variables.

## MQIPADDRV

Die Umgebungsvariable **MQIPADDRV** gibt an, welches IP-Protokoll für eine Kanalverbindung verwendet werden soll. Sie hat die möglichen Zeichenfolgewerte `"MQIPADDR_IPV4"` oder `"MQIPADDR_IPV6"`. Diese Werte haben dieselbe Bedeutung wie IPv4 und IPv6 in **ALTER QMGR IPADDRV** und das Attribut

**IPAddressVersion** der TCP-Zeilengruppe der Clientkonfigurationsdatei. Wenn die Umgebungsvariable nicht festgelegt ist, wird "MQIPADDR_IPV4" angenommen.

Verwenden Sie einen der folgenden Befehle, um **MQIPADDRV**festzulegen:

- Linux AIX Unter AIX and Linux:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6"/>
```

- Windows Unter Windows:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPV4|MQIPADDR_IPV6)
```

## MQKEYRPWD

V 9.3.0 V 9.3.0

Wenn Sie die Umgebungsvariable *MQKEYRPWD* festlegen, gibt sie das Kennwort für das Schlüsselrepository an, das das digitale Zertifikat des Benutzers enthält. Wenn *MQKEYRPWD* verwendet wird, müssen Sie das Kennwort verschlüsseln, bevor Sie den Wert der Umgebungsvariablen festlegen.

Verwenden Sie einen der folgenden Befehle, um *MQKEYRPWD*festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQKEYRPWD=passphrase
```

- Windows Auf Systemen mit Windows:

```
SET MQKEYRPWD=passphrase
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQKEYRPWD) VALUE(passphrase)
```

Für diese Umgebungsvariable gibt es keinen Standardwert.

V 9.3.0 V 9.3.0 Weitere Informationen finden Sie unter

- ALW Schlüsselrepository-Kennwort für einen IBM MQ MQI client unter AIX, Linux, and Windows angeben und Schlüsselrepository-Kennwort verschlüsseln
- IBM i Schlüsselrepository-Kennwort für einen IBM MQ MQI client unter IBM i angeben und Schlüsselrepository-Kennwort verschlüsseln.

## mqlicense

Linux

Auf Linux -Systemen können Sie die Umgebungsvariable **MQLICENSE** verwenden, um eine IBM MQ -Lizenz zu akzeptieren oder anzuzeigen, nachdem Sie das Produkt installiert haben.

Weitere Informationen dazu, warum Sie dies wünschen oder tun müssen, finden Sie unter Lizenz unter IBM MQ für Linuxakzeptieren .

Die Umgebungsvariable **MQLICENSE** kann auf einen von zwei Werten gesetzt werden:

**accept**
Akzeptieren Sie die Lizenz nach der Installation.

**Ansicht**
Lizenz anzeigen, wenn die Lizenz akzeptiert wurde.

Verwenden Sie den folgenden Befehl, um die Lizenz nach der Installation zu akzeptieren:

```
export MQLICENSE=accept
```

Verwenden Sie den folgenden Befehl, um die Lizenz anzuzeigen:

```
export MQLICENSE=view
```

**Anmerkung:** Sie können die Lizenz auch mit den folgenden Befehlen akzeptieren und anzeigen:

- mqlicense (Lizenz nach der Installation annehmen)
- dspmqlic ( IBM MQ -Lizenz anzeigen)

## MQMAXERRORLOGSIZE


Die Umgebungsvariable **MQMAXERRORLOGSIZE** gibt die Größe des Fehlerprotokolls des Warteschlangenmanagers an, das in die Sicherung kopiert wird.

Weitere Informationen finden Sie im Abschnitt Fehlerprotokolle verwenden.

## MQNAME


Die Umgebungsvariable **MQNAME** gibt den lokalen NetBIOS -Namen an, den die IBM MQ -Prozesse verwenden können. Eine NetBIOS-Verbindung gilt nur für einen Client und Server, auf dem Windows ausgeführt wird.

Verwenden Sie den folgenden Befehl, um **MQNAME**festzulegen:

```
SET MQNAME=Your_env_Name
```

For example:

```
SET MQNAME=CLIENT1
```

Einige NetBIOS -Implementierungen erfordern für jede Anwendung einen eindeutigen Namen, der von **MQNAME**festgelegt wird, wenn Sie mehrere IBM MQ -Anwendungen gleichzeitig auf dem IBM MQ MQI clientausführen.

Weitere Informationen finden Sie unter Definieren des lokalen NetBIOS -Namens für IBM MQ.

## MQNOREMPOOL

Wenn Sie die Umgebungsvariable **MQNOREMPOOL** festlegen, wird das Kanalpooling inaktiviert und die Kanäle werden als Threads des Listeners ausgeführt.

Weitere Informationen finden Sie unter MCATYPE (Nachrichtenkanalagententyp).

## MQPSE_TRACE_LOGFILE

Sie verwenden die Umgebungsvariable **MQPSE_TRACE_LOGFILE** , wenn Sie das Publish-Exit-Beispielprogramm AMQSPSE0ausführen, bei dem es sich um ein C-Beispielprogramm eines Exits handelt, um eine

Veröffentlichung abzufangen, bevor sie einem Subskribenten zugestellt wird. Im Anwendungsprozess, für den ein Trace erstellt werden soll, beschreibt diese Umgebungsvariable, wohin die Tracedateien geschrieben werden sollen.

Weitere Informationen finden Sie im Abschnitt Beispielprogramm 'Publish Exit'.

## MQS_AMSCRED_KEYDATEI

Sie können die Umgebungsvariable **MQS_AMSCRED_KEYFILE** verwenden, um die ursprüngliche Schlüsseldatei zu überschreiben oder bereitzustellen, die zur Laufzeit von IBM MQ Advanced Message Security -Anwendungen (AMS) verwendet werden soll, oder wenn Sie eine Keystore-Konfigurationsdatei mit dem Befehl **runamscred** schützen.

Weitere Informationen finden Sie unter Keystores und Zertifikate mit AMS verwenden und Kennwörter in IBM MQ -Komponentenkonfigurationsdateien schützen.

## MQS_DISABLE_ALL_INTERCEPT

Sie können die Umgebungsvariable **MQS_DISABLE_ALL_INTERCEPT** verwenden, um IBM MQ Advanced Message Security (AMS) zu inaktivieren, wenn ein Fehler `2085 (MQRC_UNKNOWN_OBJECT_NAME)` gemeldet wird, wenn Sie versuchen, eine Verbindung zu einem WS-Manager aus einer früheren Version des Produkts herzustellen, und Sie IBM MQ mit nativen C-Clients verwenden.

**Anmerkung:** Sie können die Umgebungsvariable **MQS_DISABLE_ALL_INTERCEPT** nur für C-Clients verwenden. Für Java -Clients müssen Sie stattdessen die Umgebungsvariable **AMQ_DISABLE_CLIENT_AMS** verwenden.

Weitere Informationen finden Sie unter Advanced Message Security auf dem Client inaktivieren.

## MQS_IPC_HOST

Da IPC-Dateisystemobjekte vom System unterschieden werden müssen, wird dem Verzeichnispfad ein Unterverzeichnis für jedes System hinzugefügt, auf dem der Warteschlangenmanager ausgeführt wird. Wenn der generierte Wert des Hostnamens ein Fehler verursacht, können Sie den Hostnamen mithilfe der Umgebungsvariablen **MQS_IPC_HOST** festlegen.

Weitere Informationen finden Sie unter Gemeinsame Nutzung von IBM MQ -Dateien auf Multiplatforms.

## MQS_KEYSTORE_CONF

Die Umgebungsvariable **MQS_KEYSTORE_CONF** gibt die Position der Keystore-Konfigurationsdatei für IBM MQ Advanced Message Security (AMS) an, wenn sich die Datei nicht an der Standardposition *home_directory*`/.mqs/keystore.conf`befindet.

Weitere Informationen finden Sie unter Keystores und Zertifikate mit AMSverwenden.

Wenn unter Managed File TransferProbleme auftreten, lesen Sie den Artikel Fehlerbehebung, wenn MFT Schlüsselspeichereigenschaften für AMS nicht liest.

## MQS_MQI_XX_ENCODE_CASE_ONE Schlüsseldatei

V 9.3.0 V 9.3.0

Wenn Sie die Umgebungsvariable **MQS_MQI_KEYFILE** festlegen, gibt sie die Position einer ursprünglichen Schlüsseldatei an, die den ursprünglichen Schlüssel für Kennwortschutzoperationen enthält. Wenn die ursprüngliche Schlüsseldatei nicht angegeben wird, wird der standardmäßige ursprüngliche Schlüssel vom IBM MQ -Kennwortschutzsystem verwendet.

Verwenden Sie einen der folgenden Befehle, um **MQS_MQI_KEYFILE**festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQS_MQI_KEYFILE=key file location
```

- Windows Auf Systemen mit Windows:

```
SET MQS_MQI_KEYFILE=key file location
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQS_MQI_KEYFILE) VALUE(key file location)
```

V 9.3.0 V 9.3.0 Weitere Informationen finden Sie unter Angeben eines Anfangsschlüssels für ein IBM MQ MQI client on AIX, Linux, and Windows und Angeben eines Anfangsschlüssels für ein IBM MQ MQI client on IBM i.

## MQS_SSLCRYP_KEYDATEI

V 9.3.0

Die Umgebungsvariable **MQS_SSLCRYP_KEYFILE** ist eine alternative Möglichkeit, den vollständigen Pfad und Namen der Datei anzugeben, die den ursprünglichen Schlüssel enthält, der zum Verschlüsseln des Kennworts in der Konfigurationszeichenfolge der PKCS #11 -Verschlüsselungshardware verwendet wird, anstatt sie mit dem Attribut **SSLCryptoHardwareKeyFile** in der SSL-Zeilengruppe von `qm.ini`anzugeben. **MQS_SSLCRYP_KEYFILE** hat eine höhere Priorität als die Datei `qm.ini` , sodass ihr Wert Vorrang vor allen anderen Werten hat. Weitere Informationen finden Sie unter IBM MQ-Clients mit Verschlüsselungshardware.

## MQS_TRACE_OPTIONS

AIX

Verwenden Sie für das selektive Komponententracing unter AIXdie Umgebungsvariable **MQS_TRACE_OPTIONS** , um die Tracefunktionen für hohe Details und Parameter einzeln zu aktivieren.

**Anmerkung:** Setzen Sie die Umgebungsvariable **MQS_TRACE_OPTIONS** nur, wenn Sie vom IBM Support dazu aufgefordert wurden.

Weitere Informationen finden Sie unter Traceerstellung unter AIX and Linux.

## MQSERVER

Die Umgebungsvariable **MQSERVER** wird zur Definition eines minimalen Kanals verwendet. **MQSERVER** gibt die Position des IBM MQ -Servers und die zu verwendende Übertragungsmethode an.

**Anmerkung:** Sie können mit **MQSERVER** keinen TLS-Kanal oder einen Kanal mit Kanalexits definieren. Weitere Informationen zum Definieren eines TLS-Kanals finden Sie unter Kanäle mit TLS schützen.

Die folgenden Beispiele zeigen, wie **MQSERVER**festgelegt wird:

- Linux AIX Unter AIX and Linux:

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

- Windows Unter Windows:

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

- IBM i Unter IBM i:

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

**Anmerkung:**

- Der Kanalname darf keinen Schrägstrich (/) enthalten, weil dieses Zeichen verwendet wird, um den Kanalnamen, den Transporttyp und den Verbindungsnamen zu trennen. Wenn die **MQSERVER** -Umgebungsvariable verwendet wird, um einen Clientkanal zu definieren, wird eine maximale Nachrichtenlänge (MAXMSGL) von 100 MB verwendet. Daher ist die maximale Nachrichtengröße, die für den Kanal wirksam ist, der im SVRCONN-Kanal auf dem Server angegebene Wert.
- Der Transporttyp kann je nach IBM MQ -Clientplattform `LU62` , `TCP` , `NETBIOS`oder SPXsein.
- Der Verbindungsname muss ein vollständig qualifizierter Netzname sein. Beispiel: `AMACHINE.ACOMPANY.COM(1414)`.
- Der Verbindungsname kann eine durch Kommas getrennte Liste von Verbindungsnamen sein. Die Verbindungsnamen in der Liste werden auf ähnliche Weise für mehrere Verbindungen in einer Clientverbindungstabelle verwendet. Die Liste der Verbindungsnamen kann als Alternative zu Warteschlangenmanagergruppen verwendet werden, um mehrere Verbindungen anzugeben, die der Client versuchen soll. Wenn Sie einen Warteschlangenmanager mit mehreren Instanzen konfigurieren, können Sie eine Verbindungsnamensliste verwenden, um verschiedene Warteschlangenmanagerinstanzen anzugeben.

Wenn Sie die Umgebungsvariable **MQSERVER** verwenden, um den Kanal zwischen Ihrer IBM MQ MQI client -Maschine und einer Servermaschine zu definieren, ist dies der einzige Kanal, der für Ihre Anwendung verfügbar ist, und es wird nicht auf die Definitionstabelle für Clientkanäle (CCDT) verwiesen.

Weitere Informationen finden Sie im Abschnitt Clientverbindungskanal auf dem IBM MQ MQI-Client mit MQSERVER erstellen.

## MQSNOAUT

**Warnung:** Diese Funktionalität wird nicht empfohlen.

Wenn Sie die Umgebungsvariable **MQSNOAUT** auf einen beliebigen Wert setzen, inaktiviert sie den Objektberechtigungsmanager (Object Authority Manager, OAM) und verhindert die Sicherheitsprüfung. Dies kann für eine Testumgebung geeignet sein. Dies schließt sowohl Berechtigungs-als auch Verbindungsauthentifizierungsfunktionen ein. TLS, Kanalauthentifizierungsdatensätze und Sicherheitsexits sind davon nicht betroffen.

Die Umgebungsvariable **MQSNOAUT** wird nur wirksam, wenn ein Warteschlangenmanager erstellt wird.

**Warnung:** Um den OAM zu aktivieren, müssen Sie den Warteschlangenmanager löschen, die Umgebungsvariable löschen und anschließend den Warteschlangenmanager erneut erstellen, ohne **MQSNOAUT**anzugeben.

Weitere Informationen finden Sie unter Sicherheitszugriffsprüfungen auf AIX-, Linux-und Windows-Systemen verhindern.

## MQSPREFIX

Als Alternative zum Ändern des Standardpräfix können Sie die Umgebungsvariable **MQSPREFIX** verwenden, um die **DefaultPrefix** für den Befehl **crtmqm** zu überschreiben.

Weitere Informationen finden Sie im Abschnitt IBM MQ -Dateinamen und in der Zeilengruppe AllQueueManagers der Datei mqs.ini.

## MQSSLCRYP

ALW

Die Umgebungsvariable **MQSSLCRYP** enthält eine Parameterzeichenfolge, mit der Sie die auf dem System vorhandene Verschlüsselungshardware konfigurieren können.

V 9.3.0 Die zulässigen Werte sind dieselben wie für das Feld SSLCryptoHardware in der SSL-Zeilengruppe der Clientkonfigurationsdatei.

Verwenden Sie einen der folgenden Befehle, um **MQSSLCRYP**festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQSSLCRYP=string
```

- Windows Auf Systemen mit Windows:

```
SET MQSSLCRYP=string
```

Weitere Informationen finden Sie in Konfiguration für Verschlüsselungshardware unter AIX, Linux, and Windows und IBM MQ clients , die Verschlüsselungshardware verwenden unter Kennwörter in IBM MQ -Komponentenkonfigurationsdateien schützen.

## MQSSLFIPS

Die Umgebungsvariable **MQSSLFIPS** gibt an, ob bei der Verschlüsselung in IBM MQnur FIPS-zertifizierte Algorithmen verwendet werden sollen. Sie können diese Umgebungsvariable auf YES oder NO setzen. Der Standardwert ist NO. Diese Werte entsprechen den Werten für den Parameter **SSLFIPS** des Befehls **ALTER QMGR** .

Verwenden Sie einen der folgenden Befehle, um **MQSSLFIPS**festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQSSLFIPS=YES|NO
```

- Windows Auf Systemen mit Windows:

```
SET MQSSLFIPS=YES|NO
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

Die Verwendung FIPS-zertifizierter Algorithmen wird durch die Verwendung von Verschlüsselungshardware beeinflusst. Weitere Informationen finden Sie unter Angeben, dass nur FIPS-zertifizierte CipherSpecs zur Ausführungszeit auf dem MQI-Client verwendet werden.

## MQSSLKEYR

Die Umgebungsvariable **MQSSLKEYR** gibt die Position des Schlüsselrepositorys, das das digitale Zertifikat des Benutzersenthält.

V 9.3.0 V 9.3.0 Geben Sie den vollständigen Pfad und den Dateinamen des Schlüsselrepositorys an. Wenn das Dateisuffix nicht angegeben wird, wird angenommen, dass es .kdbist.

Verwenden Sie einen der folgenden Befehle, um **MQSSLKEYR**festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQSSLKEYR=pathname
```

- Windows Auf Systemen mit Windows:

```
SET MQSSLKEYR=pathname
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

Für diese Umgebungsvariable gibt es keinen Standardwert.

Weitere Informationen finden Sie in der Beschreibung des Parameters **SSLKEYR** des Befehls **ALTER QMGR** .

## MQSSLPROXY

Die Umgebungsvariable **MQSSLPROXY** gibt den Hostnamen und die Portnummer des HTTP-Proxy an, der von GSKit für OCSP-Prüfungen verwendet wird.

Verwenden Sie einen der folgenden Befehle, um **MQSSLPROXY**festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQSSLPROXY="string"
```

- Windows Auf Systemen mit Windows:

```
SET MQSSLPROXY= string
```

Die Zeichenfolge, die Sie mit **MQSSLPROXY** angeben, kann entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers sein, der von GSKit für OCSP-Prüfungen verwendet werden soll. Auf die Adresse kann optional eine in Klammern gesetzte Portnummer folgen. Wenn Sie die Portnummer nicht angeben, wird der Standard-HTTP-Port 80 verwendet.

Linux AIX Auf AIX and Linux-Systemen können Sie zum Beispiel einen der folgenden Befehle verwenden:

- 
```
export MQSSLPROXY="proxy.example.com(80)"
```

- 
```
export MQSSLPROXY="127.0.0.1"
```

Weitere Informationen finden Sie unter Mit OCSP (Online Certificate Status Protocol) arbeiten.

## MQSSLRESET

Die Umgebungsvariable **MQSSLRESET** gibt die Anzahl der unverschlüsselten Byte an, die auf einem TLS-Kanal gesendet und empfangen werden, bevor der geheime TLS-Schlüssel neu vereinbart wird. Er kann auf eine ganze Zahl im Bereich von 0 bis 999 999 999 gesetzt werden. Der Standardwert ist 0, was bedeutet, dass geheime Schlüssel nie neu verhandelt werden. Wenn Sie einen TLS-Rücksetzzähler für geheime Schlüssel im Bereich von 1 Byte bis 32 KB angeben, verwenden die TLS-Kanäle eine Zurücksetzungsanzahl von 32 KB für den geheimen Schlüssel. Dieser geheime Rücksetzzähler verhindert, dass überhöhte Schlüsselsätze für kleine TLS-Rücksetzwerte für geheime Schlüssel verwendet werden.

Verwenden Sie einen der folgenden Befehle, um **MQSSLRESET**festzulegen:

- Linux AIX Auf Systemen mit AIX and Linux:

```
export MQSSLRESET=integer
```

- Windows Auf Systemen mit Windows:

```
SET MQSSLRESET=integer
```

- IBM i Unter IBM i:

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

Weitere Informationen finden Sie unter Zurücksetzen von geheimen SSL-und TLS-Schlüsseln.

## MQSUITEB

ALW

Sie können IBM MQ so konfigurieren, dass es in Übereinstimmung mit dem Standard NSA Suite B auf den Plattformen AIX, Linux, and Windows ausgeführt wird.

Die Umgebungsvariable **MQSUITEB** gibt an, ob die Suite B-konforme Verschlüsselung verwendet wird. Wenn Suite B-Verschlüsselung verwendet werden soll, können Sie die Stärke der Verschlüsselung angeben, indem Sie **MQSUITEB** auf einen der folgenden Werte setzen:

- KEINE
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

Sie können mehrere Werte in einer durch Kommas getrennten Liste angeben. Die Verwendung des Werts NONE mit einem anderen Wert ist ungültig.

Weitere Informationen finden Sie unter IBM MQ für Suite B konfigurieren.

## MQTCPTIMEOUT

Die Umgebungsvariable **MQTCPTIMEOUT** gibt an, wie lange IBM MQ auf einen TCP-Verbindungsaufruf wartet.

## ODQ_MSG

Wenn Sie eine Steuerroutine der Warteschlange für nicht zustellbare Nachrichten verwenden, die sich von **runmqdlq**unterscheidet, steht Ihnen die Quelle des Beispiels `amqsdlq`als Basis zur Verfügung. Das Beispiel entspricht der Steuerroutine für nicht zustellbare Nachrichten, die innerhalb des Produkts bereitgestellt wird, verfügt aber über andere Trace- und Fehlermeldungsfunktionen. Verwenden Sie die Umgebungsvariable **ODQ_MSG** , um den Namen der Datei festzulegen, die Fehler-und Informationsnachrichten enthält. Die bereitgestellte Datei heißt `amqsdlq.msg`.

Weitere Informationen finden Sie unter Beispiel für Steuerroutine der Warteschlange für nicht zustellbare Nachrichten.

## ODQ_TRACE

Wenn Sie eine Steuerroutine der Warteschlange für nicht zustellbare Nachrichten verwenden, die sich von **runmqdlq**unterscheidet, steht Ihnen die Quelle des Beispiels `amqsdlq`als Basis zur Verfügung.

Das Beispiel entspricht der Steuerroutine für nicht zustellbare Nachrichten, die innerhalb des Produkts bereitgestellt wird, verfügt aber über andere Trace- und Fehlermeldungsfunktionen. Setzen Sie die Umgebungsvariable **ODQ_TRACE** auf YES oder yes, um die Traceerstellung zu aktivieren.

Weitere Informationen finden Sie unter Beispiel für Steuerroutine der Warteschlange für nicht zustellbare Nachrichten.

## WCF_TRACE_ON

Für den angepassten WCF-Kanal sind zwei unterschiedliche Tracemethoden verfügbar. Diese beiden Tracemethoden werden entweder unabhängig oder zusammen aktiviert. Jedes Verfahren liefert eine eigene Tracedatei, sodass zwei Traceausgabedateien erstellt werden, wenn beide Traceverfahren aktiviert wurden. Es gibt vier mögliche Kombinationen für das Aktivieren und Inaktivieren der beiden Traceverfahren. Neben diesen Kombinationen zur Aktivierung des WCF-Trace kann der XMS .NET -Trace mithilfe der Umgebungsvariablen **WCF_TRACE_ON** aktiviert werden.

Weitere Informationen finden Sie unter Traceerstellung für den angepassten WCF-Kanal für IBM MQ.

## WMQSOAP_HOME

Die Umgebungsvariable **WMQSOAP_HOME** wird verwendet, wenn zusätzliche Konfigurationsschritte ausgeführt werden, nachdem die Hosting-Umgebung des Service .NET SOAP over JMS ordnungsgemäß in IBM MQinstalliert und konfiguriert wurde. Sie ist für lokale Warteschlangenmanager zugänglich.

Weitere Informationen finden Sie unter WCF-Client zu einem .NET -Service, der von IBM MQ sample gehostet wird, und WCF-Client zu einem Axis Java -Service, der von IBM MQ samplegehostet wird.

## XMS_TRACE_ON, XMS_TRACE_FILE_PATH, XMS_TRACE_FORMAT und XMS_TRACE_SPECIFICATION

Wenn Sie IBM MQ classes for XMS .NET Frameworkverwenden, können Sie den Trace aus einer Anwendungskonfigurationsdatei sowie aus den XMS -Umgebungsvariablen konfigurieren. Wenn Sie IBM MQ classes for XMS .NET (Bibliotheken.NET Standard und .NET 6 )verwenden, müssen Sie den Trace über die XMS -Umgebungsvariablen konfigurieren. Der Trace wird normalerweise unter Anleitung von IBM Support verwendet.

Um den Trace für eine XMS .NET -Anwendung zu aktivieren und zu konfigurieren, legen Sie die folgenden Umgebungsvariablen fest, bevor Sie die Anwendung ausführen:

**XMS_TRACE_ON**
Wenn die Umgebungsvariable **XMS_TRACE_ON** festgelegt ist, ist standardmäßig der gesamte Trace aktiviert.

**XMS_TRACE_FILE_PATH**
Die Umgebungsvariable **XMS_TRACE_FILE_PATH** gibt den vollständig qualifizierten Pfadnamen des Verzeichnisses an, in das Trace-und FFDC-Datensätze geschrieben werden, wenn diese Datensätze an eine alternative Position aus dem aktuellen Arbeitsverzeichnis geschrieben werden sollen.

**XMS_TRACE_FORMAT**
Die Umgebungsvariable **XMS_TRACE_FORMAT** gibt das erforderliche Traceformat an, das BASIC oder ADVANCEDsein kann.

**XMS_TRACE_SPECIFICATION**
Die Umgebungsvariable **XMS_TRACE_SPECIFICATION** überschreibt die Traceeinstellungen, die im Traceabschnitt einer Anwendungskonfigurationsdateidefiniert sind. **XMS_TRACE_SPECIFICATION** gilt nur für IBM MQ classes for XMS .NET Framework .

Weitere Informationen finden Sie unter Traceerstellung für XMS .NET -Anwendungen und Traceerstellung für XMS .NET -Anwendungen mit XMS -Umgebungsvariablen.

# Multi IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen, indem Sie die Informationen in den Konfigurationsdateien (`.ini`) bearbeiten. Sie können auch Konfigurationsoptionen für IBM MQ MQI clientsändern.

## Informationen zu diesem Vorgang

Sie können IBM MQ -Konfigurationsdaten auf der Ebene des Knotens oder des Warteschlangenmanagers ändern, indem Sie die Werte ändern, die in einer Gruppe von Konfigurationsattributen (oder Parametern) angegeben sind, die IBM MQregeln.

Eine Konfigurationsdatei (oder eine Zeilengruppendatei) enthält eine oder mehrere Zeilengruppen, bei denen es sich um Gruppen von Zeilen in der `.ini`-Datei handelt, die zusammen eine allgemeine Funktion haben oder einen Teil eines Systems definieren, wie z. B. Protokollfunktionen, Kanalfunktionen und installierbare Services. Sie können IBM MQ -Konfigurationsattribute in den folgenden Konfigurationsdateien ändern:

**IBM MQ -Konfigurationsdatei `mqs.ini`**
Die Datei `mqs.ini` wirkt sich auf den Knoten als Ganzes aus. Für jede IBM MQ -Installation gibt es eine Datei `mqs.ini` .

Da die IBM MQ-Konfigurationsdatei zur Lokalisierung der zu den Warteschlangenmanagern gehörigen Daten benötigt wird, würde das Fehlen bzw. eine falsche Einstellung dieser Datei dazu führen, dass keine oder einige der MQSC-Befehle fehlschlagen. Auch die Verbindung zwischen Anwendungen und einem nicht in der IBM MQ-Konfigurationsdatei definierten Warteschlangenmanager funktioniert nicht.

**Installationskonfigurationsdatei, mqinst.ini**
Linux AIX Auf AIX and Linux -Systemen enthält die Installationskonfigurationsdatei `mqinst.ini`Informationen zu allen IBM MQ -Installationen. Die Datei `mqinst.ini` darf nicht direkt bearbeitet oder referenziert werden, da ihr Format nicht fixiert ist und sich ändern kann. Stattdessen müssen Sie sie mithilfe von Befehlen bearbeiten.

**Konfigurationsdatei des Warteschlangenmanagers `qm.ini`**
Die Datei `qm.ini` wirkt sich auf Änderungen für bestimmte Warteschlangenmanager aus. Für jeden Warteschlangenmanager auf dem Knoten gibt es eine `qm.ini`-Datei.

**IBM MQ MQI client -Konfigurationsdatei, mqclient.ini**
Konfigurationsoptionen für IBM MQ MQI clients werden separat in der Clientkonfigurationsdatei gespeichert, die im Allgemeinen den Namen `mqclient.ini`hat.

**Aktivitätstracekonfigurationsdatei mqat.ini**
Die Datei `mqat.ini` wird verwendet, um das Verhalten des Aktivitätstrace zu konfigurieren.

Möglicherweise müssen Sie eine Konfigurationsdatei bearbeiten, wenn Sie zum Beispiel:

- Sie verlieren eine Konfigurationsdatei. (Wiederherstellen aus der Sicherung, wenn möglich.)
- Sie müssen einen oder mehrere WS-Manager in ein neues Verzeichnis verschieben.
- Sie müssen Ihren Standardwarteschlangenmanager ändern. Dies kann vorkommen, wenn Sie den vorhandenen Warteschlangenmanager versehentlich löschen.
- Dies wird Ihnen vom IBM Support empfohlen.

**Wichtig:** Änderungen, die Sie an einer Konfigurationsdatei vornehmen, werden normalerweise erst wirksam, wenn der Warteschlangenmanager das nächste Mal gestartet wird.

**Hinweise zur Bearbeitung von Konfigurationsdateien:**

- Die Werte der Attribute einer Konfigurationsdatei werden entsprechend den folgenden Prioritäten festgelegt:

  - Parameter, die in der Befehlszeile eingegeben werden, haben Vorrang vor Werten, die in den Konfigurationsdateien definiert sind.
  - Werte, die in den `qm.ini` -Dateien definiert sind, haben Vorrang vor Werten, die in der Datei `mqs.ini` definiert sind.
- Nach der Installation können Sie die Standardwerte in den IBM MQ -Konfigurationsdateien bearbeiten.
- Denken Sie beim Sichern eines Warteschlangenmanagers daran, sowohl seine Konfigurationsdatei (`qm.ini`) als auch die zentrale IBM MQ -Konfigurationsdatei (`mqs.ini`) einzuschließen.
- Wenn Sie einen falschen Wert für ein Konfigurationsdateiattribut festlegen, wirkt sich dies so aus, als würde das Attribut vollständig fehlen. Der Wert wird ignoriert und eine Bedienernachricht wird ausgegeben, um auf das Problem hinzuweisen.
- IBM i Unter IBM isind die `.ini` -Dateien Datenstromdateien, die sich im IFS befinden.
- Es gibt eine Reihe von Syntaxregeln für das Format der Datei `mqat.ini` . Weitere Informationen finden Sie im Abschnitt Verhalten des Aktivitätstrace mit `mqat.ini`konfigurieren.

## Vorgehensweise

1. Bevor Sie eine Konfigurationsdatei bearbeiten, sichern Sie sie, damit Sie eine Kopie haben, auf die Sie bei Bedarf zurückgreifen können.
2. Bearbeiten Sie die Konfigurationsdatei `.ini` auf eine der folgenden Arten:
   - Manuell mit einem Standardtexteditor. Kommentare können in Konfigurationsdateien aufgenommen werden, indem ein "; " oder ein " # " Zeichen vor dem Kommentartext hinzugefügt wird. Wenn Sie ein ";" -oder ein "#" -Zeichen verwenden möchten, ohne dass es einen Kommentar darstellt, können Sie dem Zeichen das Zeichen "\" voranstellen. Das Zeichen wird dann als Teil der Konfigurationsdaten verwendet.
   - Automatisch mit Befehlen, die die Konfiguration von Warteschlangenmanagern auf dem Knoten ändern. Weitere Informationen finden Sie unter Befehlsreferenz.

     Windows Der Windows spezifische Befehl **amqmdain** aktualisiert beispielsweise eine Untergruppe der `qm.ini` -Eigenschaften automatisch. Weitere Informationen finden Sie in amqmdain .
   - Windows Linux Unter Linux (x86 und x86-64) und Windowskönnen Sie eine Untergruppe der `qm.ini` -Eigenschaften mit IBM MQ Exploreraktualisieren. Weitere Informationen finden Sie im Abschnitt IBM MQ mit MQ Explorer konfigurieren .

     **Anmerkung:** Da die Änderung installierbarer Services und ihrer Komponenten erhebliche Auswirkungen hat, sind die installierbaren Services in IBM MQ Explorerschreibgeschützt. Daher müssen Sie alle Änderungen an installierbaren Services vornehmen, indem Sie die Datei `qm.ini` bearbeiten. Weitere Informationen finden Sie unter „Zeilengruppe 'Service' in der Datei 'qm.ini'" auf Seite 150.

**Zugehörige Tasks**
IBM MQ verwalten

# Multi IBM MQ-Konfigurationsdatei, mqs.ini

Die IBM MQ-Konfigurationsdatei `mqs.ini` enthält Informationen, die für alle Warteschlangenmanager auf dem Knoten relevant sind. Sie wird automatisch während der Installation erstellt.

**Anmerkung:** Weitere Informationen dazu, wie und wann Sie die Datei `mqs.ini` bearbeiten und wann Änderungen, die Sie an der Datei vornehmen, wirksam werden, finden Sie unter „IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern" auf Seite 90.

## Verzeichnispositionen

Linux AIX Unter AIX and Linux sind das Datenverzeichnis und das Protokollverzeichnis immer `/var/mqm` und `/var/mqm/log`.

Windows Auf Windows-Systemen werden die Position des Datenverzeichnisses `mqs.ini` und die Position des Protokollverzeichnisses in der Registry gespeichert, da die Position des Verzeichnisses variieren kann. Die Informationen zur Installation der Installation, die in `mqinst.ini` auf AIX and Linux-Systemen enthalten sind, befinden sich ebenfalls in der Registry, da es keine `mqinst.ini`-Datei unter Windows gibt (siehe „Installationskonfigurationsdatei, mqinst.ini” auf Seite 167).

Windows Die Datei `mqs.ini` für Windows -Systeme wird durch den im Schlüssel `HKLM\SOFTWARE\IBM\IBM MQ` angegebenen WorkPath angegeben. Sie enthält:

- Die Namen der WS-Manager
- Der Name des Standard-WS-Managers.
- Die Position der Dateien, die jedem von ihnen zugeordnet sind.

IBM i Unter IBM iwird die Datei `mqs.ini` in `/QIBM/UserData/mqm`gespeichert. Die Datei enthält Folgendes:

- Die Namen der Warteschlangenmanager.
- Der Name des Standardwarteschlangenmanagers.
- Die Position der Dateien, die den einzelnen Warteschlangenmanagern zugeordnet sind.
- Informationen zur Identifizierung von API-Exits (weitere Informationen hierzu finden Sie im Abschnitt API-Exits konfigurieren ).

Insbesondere wird die Datei `mqs.ini` verwendet, um die Daten zu lokalisieren, die den einzelnen Warteschlangenmanagern zugeordnet sind.

## Beispieldatei `mqs.ini` für AIX and Linux

Linux AIX

```
#********************************************************************#
#* Module Name: mqs.ini                                             *#
#* Type       : IBM MQ Machine-wide Configuration File              *#
#* Function   : Define IBM MQ resources for an entire machine       *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This is the installation time default configuration           *#
#*                                                                  *#
#********************************************************************#
AllQueueManagers:
#********************************************************************#
#* The path to the qmgrs directory, below which queue manager data  *#
#* is stored                                                        *#
#********************************************************************#
DefaultPrefix=/var/mqm

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogDefaultPath=/var/mqm/log

QueueManager:
   Name=saturn.queue.manager
   Prefix=/var/mqm
   Directory=saturn!queue!manager
   InstallationName=Installation1

QueueManager:
   Name=pluto.queue.manager
   Prefix=/var/mqm
   Directory=pluto!queue!manager
   InstallationName=Installation2

DefaultQueueManager:
   Name=saturn.queue.manager
```

```
ApiExitTemplate:
   Name=OurPayrollQueueAuditor
   Sequence=2
   Function=EntryPoint
   Module=/usr/ABC/auditor
   Data=123

ApiExitCommon:
   Name=MQPoliceman
   Sequence=1
   Function=EntryPoint
   Module=/usr/MQPolice/tmqp
   Data=CheckEverything
```

## Beispieldatei `mqs.ini` für Windows

Windows

```
#*******************************************************************#
#* Module Name: mqs.ini                                            *#
#* Type       : IBM MQ Machine-wide Configuration File             *#
#* Function   : Define IBM MQ resources for an entire machine      *#
#*******************************************************************#
#* Notes      :                                                    *#
#* 1) This is the installation time default configuration          *#
#*                                                                 *#
#*******************************************************************#
AllQueueManagers:
#*******************************************************************#
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored                                                       *#
#*******************************************************************#
DefaultPrefix=C:\ProgramData\IBM\MQ

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogDefaultPath=C:\ProgramData\IBM\MQ\log

QueueManager:
   Name=saturn.queue.manager
   Prefix=C:\ProgramData\IBM\MQ
   Directory=saturn!queue!manager
   InstallationName=Installation1

QueueManager:
   Name=pluto.queue.manager
   Prefix=C:\ProgramData\IBM\MQ
   Directory=pluto!queue!manager
   InstallationName=Installation2

DefaultQueueManager:
   Name=saturn.queue.manager

ApiExitTemplate:
   Name=OurPayrollQueueAuditor
   Sequence=2
   Function=EntryPoint
   Module=C:\usr\ABC\auditor
   Data=123

ApiExitCommon:
   Name=MQPoliceman
   Sequence=1
   Function=EntryPoint
   Module=C:\usr\MQPolice\tmpq
   Data=CheckEverything
```

## Beispieldatei `mqs.ini` für IBM i

IBM i

```
#********************************************************************#
#* Module Name: mqs.ini                                             *#
#* Type       : IBM MQ Configuration File                           *#
#* Function   : Define IBM MQ resources for the node                *#
#*                                                                  *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This is an example IBM MQ configuration file                  *#
#*                                                                  *#
#********************************************************************#
AllQueueManagers:
#********************************************************************#
#* The path to the qmgrs directory, within which queue manager data *#
#* is stored                                                        *#
#********************************************************************#
DefaultPrefix=/QIBM/UserData/mqm

QueueManager:
Name=saturn.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMSATURN.Q
Directory=saturn!queue!manager

QueueManager:
Name=pluto.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMPLUTO.QU
Directory=pluto!queue!manager

DefaultQueueManager:
Name=saturn.queue.manager
```

**Anmerkungen:**

1. IBM MQ verwendet auf dem Knoten die Standardpositionen für WS-Manager und die Journale.
2. Der Warteschlangenmanager saturn.queue.manager ist der Standardwarteschlangenmanager für den Knoten. Das Verzeichnis für Dateien, die diesem WS-Manager zugeordnet sind, wurde automatisch in einen gültigen Dateinamen für das Dateisystem umgewandelt.
3. Da die IBM MQ-Konfigurationsdatei zum Lokalisieren der Daten verwendet wird, die Warteschlangenmanagern zugeordnet sind, kann eine nicht vorhandene oder falsche Konfigurationsdatei dazu führen, dass einige oder alle IBM MQ-Befehle fehlschlagen. Auch die Verbindung zwischen Anwendungen und einem nicht in der IBM MQ-Konfigurationsdatei definierten Warteschlangenmanager funktioniert nicht.

## `mqs.ini` Zeilengruppen

**Achtung:** Dieser Abschnitt enthält Links zu weiteren Informationen zu den Zeilengruppen in der Datei `mqs.ini` . Jede Zeilengruppe enthält Informationen zu den Parametern in dieser Zeilengruppe.

## Multi Zusammenfassung der Zeilengruppen und Attribute der Datei mqs.ini

Eine Zusammenfassung der Attribute der Zeilengruppen der IBM MQ-Konfigurationsdatei, `mqs.ini`, mit Links zu weiteren Informationen.

*Tabelle 9. Zeilengruppen in der Datei 'mqs.ini'*

| Zeilengruppen und Attribute | Beschreibung der Attribute |
|---|---|
| **AllQueueManagers (Zeilengruppe)** | |
| DefaultPrefix | Der Pfad zum `qmgrs`-Verzeichnis, in dem die Daten des Warteschlangenmanagers gespeichert werden. |
| ALW DefaultEphemeralPräfix | Der Pfad zum Verzeichnis, in dem die temporären Daten des Warteschlangenmanagers gespeichert werden. |

| *Tabelle 9. Zeilengruppen in der Datei 'mqs.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| Multi ConvEBCDICNewline | Informationen zum Konvertieren des EBCDIC-NL-Zeichens in das ASCII-Format durch IBM MQ |
| **Zeilengruppe 'ApiExitCommon' und Zeilengruppe 'ApiExitTemplate'** | |
| Name | Der beschreibende Name des API-Exits, der im Feld 'ExitInfoName' der MQAXP-Struktur übergeben wird. |
| FUNCTION | Der Name des Einstiegspunkts der Funktion in dem Modul, das den API-Exitcode enthält. |
| Modul | Das Modul, das den API-Exitcode enthält. |
| Daten | Die Daten, die an den API-Exit im Feld 'ExitData' der MQAXP-Struktur übergeben werden sollen. |
| Sequenz | Die Reihenfolge, in der dieser API-Exit in Relation zu anderen API-Exits aufgerufen wird. |
| **DefaultQueueManager (Zeilengruppe)** | |
| Name | Der Name des Warteschlangenmanagers, der alle Befehle verarbeitet, für die nicht explizit ein Warteschlangenmanagername angegeben wurde. |
| **Zeilengruppe 'ExitProperties'** | |
| CLWLMode | Gibt an, ob der CLWL-Exit (Cluster Workload) im Modus FAST oder SAFE ausgeführt wird. |
| **LogDefaults, Zeilengruppe** | |
| LogPrimaryFiles | Die Protokolldateien, die beim Erstellen des Warteschlangenmanagers zugeordnet werden. |
| LogSecondaryFiles | Die Protokolldateien, die zugeordnet werden, wenn die Primärdateien erschöpft sind. |
| LogFilePages | Die Anzahl der Seiten in der Protokolldatei. (Die Protokolldateigröße wird in Einheiten von 4-KB-Seiten angegeben.) |
| LogType | Der Typ der Protokollierung, die vom Warteschlangenmanager verwendet werden soll (Umlaufprotokollierung oder lineare Protokollierung). |
| LogBufferPages | Die Größe des Speichers, der den Pufferdatensätzen für das Schreiben zugeordnet ist, wobei die Größe der Puffer in Einheiten von 4-KB-Seiten angegeben wird. |
| LogDefaultPath | Das Verzeichnis, in dem sich die Protokolldateien für einen WS-Manager befinden. |
| LogWriteIntegrity | Die Methode, die die Protokollfunktion verwendet, um Protokollsätze zuverlässig zu schreiben. |
| **Zeilengruppe 'QueueManager'** | |
| Name | Der Name des Warteschlangenmanagers. |
| PREFIX | Gibt an, wo die Warteschlangenmanagerdateien gespeichert sind. |

| *Tabelle 9. Zeilengruppen in der Datei 'mqs.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| Verzeichnis | Der Name des Unterverzeichnisses unter dem Verzeichnis `prefix\QMGRS`, in dem die Warteschlangenmanager-Dateien gespeichert sind. |
| DataPath | Ein expliziter Datenpfad, der beim Erstellen des Warteschlangenmanagers bereitgestellt wurde, überschreibt Prefix und Directory als Pfad zu den WS-Manager-Daten. |
| InstallationName | Der Name der IBM MQ-Installation, die diesem Warteschlangenmanager zugeordnet ist. |
| EphemeralPrefix | Speicherort der temporären Daten des Warteschlangenmanagers. |

## Multi Zeilengruppe 'AllQueueManagers' in der Datei 'mqs.ini'

Die Zeilengruppe AllQueueManagers kann den Pfad zum Verzeichnis `qmgrs` , in dem die einem Warteschlangenmanager zugeordneten Dateien gespeichert sind, den Pfad zur ausführbaren Bibliothek und die Methode für die Konvertierung von Daten im EBCDIC-Format in das ASCII-Format angeben.

Mithilfe der Zeilengruppe AllQueueManagers in der Datei `mqs.ini` können Sie Informationen zu allen Warteschlangenmanagern angeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite IBM MQ Explorer `General` und `Extended` IBM MQ verwenden.

**DefaultPrefix=** ***directory_name***

Dieses Attribut gibt den Pfad zum `qmgrs`-Verzeichnis an, in dem die Daten des Warteschlangenmanagers gespeichert werden.

Wenn Sie das Standardpräfix für den Warteschlangenmanager ändern, replizieren Sie die Verzeichnisstruktur, die zur Installationszeit erstellt wurde. Insbesondere müssen Sie die qmgrs-Struktur erstellen. Stoppen Sie IBM MQ vor dem Ändern des Standardpräfixes und starten Sie IBM MQ erst wieder neu, wenn Sie die Struktur an die neue Position verschoben und den Standardpräfix geändert haben.

**Anmerkung:** ALW Löschen Sie nicht das Verzeichnis `/var/mqm/errors` auf AIX and Linux-Systemen oder das Verzeichnis `\errors` auf Windows-Systemen.

Als Alternative zum Ändern des Standardpräfix können Sie die Umgebungsvariable **MQSPREFIX** verwenden, um die **DefaultPrefix** für den Befehl **crtmqm** zu überschreiben.

Aufgrund der Einschränkungen des Betriebssystems müssen Sie den angegebenen Pfad so kurz halten, dass die Summe der Pfadlänge und aller WS-Manager-Namen maximal 70 Zeichen lang ist.

Multi **DefaultEphemeralPrefix=** ***Verzeichnisname***

Dieses Attribut gibt den Pfad zu dem Verzeichnis an, in dem die temporären Daten des Warteschlangenmanagers (z. B. IPC-Sockets) gespeichert werden; es wird nur verwendet, um beim Erstellen eines Warteschlangenmanagers den **EphemeralPrefix** eines Warteschlangenmanagers zu definieren. Darüber hinaus müssen Sie das Verzeichnis selbst erstellen, wenn Sie den Standardwert ändern. Sie müssen das temporäre Datenverzeichnis mit Berechtigungen erstellen, die es IBM MQ über Gruppenzugriff ermöglichen, in dieses Verzeichnis zu schreiben.

Als Alternative zum Ändern der Datei `mqs.ini` kann die Umgebungsvariable **MQ_EPHEMERAL_PREFIX** verwendet werden, um die **DefaultEphemeralPrefix** für den Befehl **crtmqm** zu überschreiben.

Aufgrund von Beschränkungen seitens des Betriebssystems unterliegt standardmäßige temporäre Präfix folgenden Einschränkungen:

- Linux AIX 12 Zeichen auf AIX and Linux-Plattformen.
- IBM i 24 Zeichen unter IBM i.

MQ Appliance **DefaultEphemeralPrefix** wird in IBM MQ Appliance nicht unterstützt.

Multi **ConvEBCDICNewline = NL_TO_LF | TABLE | ISO**
EBCDIC-Codepages enthalten ein Zeilenvorschubzeichen (NL), das von ASCII-Codepages nicht unterstützt wird (obwohl einige ISO-Varianten von ASCII ein Äquivalent enthalten). Verwenden Sie das Attribut **ConvEBCDICNewline**, um anzugeben, wie IBM MQ das EBCDIC-NL-Zeichen in das ASCII-Format konvertieren soll.

IBM i Unter IBM MQ for IBM i wird CCSID 1253 als ISO CCSID betrachtet, und NL_TO_LF wirkt sich sowohl auf ISO- als auch auf ASCII-Konvertierungen aus.

z/OS Das Attribut **ConvEBCDICNewline** ist unter z/OS nicht verfügbar. Das Verhalten in z/OS entspricht dem Verhalten von `ConvEBCDICNewline=TABLE`. Beachten Sie, dass der Standardwert auf anderen Plattformen unterschiedlich sein kann.

**NL_TO_LF**
Konvertieren Sie das EBCDIC-NL-Zeichen (X'15 ') in das ASCII-Zeilenvorschubzeichen, LF (X'0A'), für alle EBCDIC-zu-ASCII-Konvertierungen.

NL_TO_LF ist der Standardwert.

**TABELLE**
Konvertieren Sie das EBCDIC-NL-Zeichen entsprechend den Konvertierungstabellen, die auf Ihrer Plattform für alle EBCDIC-zu-ASCII-Konvertierungen verwendet werden.

Der Effekt dieses Konvertierungstyps kann von Plattform zu Plattform und von Sprache zu Sprache variieren. Selbst auf der gleichen Plattform kann das Verhalten variieren, wenn Sie andere CCSIDs verwenden.

**ISO**
Konvertieren:

- ISO-CCSIDs unter Verwendung der Methode TABLE
- Alle anderen CCSIDs unter Verwendung der Methode NL_TO_CF

Die möglichen ISO-CCSIDs werden im Abschnitt Tabelle 10 auf Seite 97 aufgeführt.

*Tabelle 10. Liste der möglichen ISO-CCSIDs*

| CCSID | Codetruppe |
|---|---|
| 819 | ISO8859-1 |
| 912 | ISO8859-2 |
| 915 | ISO8859-5 |
| 1089 | ISO8859-6 |
| 813 | ISO8859-7 |
| 916 | ISO8859-8 |
| 920 | ISO8859-9 |
| 1051 | roman8 |

Wenn die ASCII-CCSID keine ISO-Untergruppe ist, hat **ConvEBCDICNewline** standardmäßig den Wert NL_TO_LF.

Ab IBM MQ 9.1.0 Fix Pack 2 und IBM MQ 9.1.2können Sie die **AMQ_CONVEBCDICNEWLINE** -Umgebungsvariable anstelle des Zeilengruppenattributs **ConvEBCDICNewline** verwenden, um beispielsweise die **ConvEBCDICNewline** -Funktionalität auf der Clientseite bereitzustellen, wenn die Datei `mqs.ini` nicht verwendet werden kann. Die Umgebungsvariable nimmt die gleichen Werte (`NL_TO_LF`, `TABLE` oder `ISO`) wie das Attribut **ConvEBCDICNewline** an. Wenn das Zeilengruppenattribut und die Umgebungsvariable festgelegt sind, hat das Zeilengruppenattribut Vorrang.

## Multi Zeilengruppen ApiExitCommon und ApiExitTemplate in der Datei mqs.ini

Die allgemeinen Zeilengruppen ApiExitund ApiExitgeben API-Exitroutinen für alle Warteschlangenmanager an.

Verwenden Sie die allgemeinen Zeilengruppen ApiExitund ApiExitin der Datei `mqs.ini` , um API-Exitroutinen für alle Warteschlangenmanager anzugeben. (Zur Angabe von API-Exitroutinen für einzelne WS-Manager verwenden Sie die Zeilengruppe ApiExitLocal, wie in „Zeilengruppe 'ApiExitLocal' in der Datei 'qm.ini'" auf Seite 118beschrieben.)

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite IBM MQ Explorer `Exits` IBM MQ verwenden.

Windows Unter Windowskönnen Sie auch den Befehl **amqmdain** verwenden, um die Einträge für API-Exits zu ändern.

Weitere Informationen zur Verwendung dieser Attribute finden Sie unter API-Exits konfigurieren.

**Name=Name_des_API-Exits**
Der beschreibende Name des API-Exits, der im Feld 'ExitInfoName' der MQAXP-Struktur übergeben wird.

Dieser Name muss eindeutig sein und darf maximal 48 Zeichen umfassen, wobei es sich um gültige Zeichen für die Namen von IBM MQ-Objekten (z. B. Warteschlangennamen) handeln muss.

**Function=Funktionsname**
Der Name des Einstiegspunkts der Funktion in dem Modul, das den API-Exitcode enthält. Dieser Einstiegspunkt ist die Funktion MQ_INIT_EXIT.

Die Länge dieses Felds ist auf den Wert von MQ_EXIT_NAME_LENGTH beschränkt.

**Module=Modulname**
Das Modul, das den API-Exitcode enthält.

Wenn dieses Feld den vollständigen Pfadnamen enthält, wird es unverändert übernommen. Wenn dieses Feld lediglich den Modulnamen enthält, wird das Modul über das Attribut **ExitsDefaultPath** in der Zeilengruppe `ExitPath` der Datei `qm.ini` lokalisiert.

Auf Plattformen, die separate Threadbibliotheken unterstützen, muss sowohl eine Threadversion als auch eine Version des API-Exitmoduls ohne Threads bereitgestellt werden. Die Threadversion muss das Suffix `_r` aufweisen. Von der Threadversion des IBM MQ-Anwendungsstubs wird dem angegebenen Modulnamen vor dem Laden implizit `_r` angehängt.

Die Länge dieses Felds ist auf die maximale Pfadlänge begrenzt, die von der Plattform unterstützt wird.

**Data=Datenname**
Die Daten, die an den API-Exit im Feld 'ExitData' der MQAXP-Struktur übergeben werden sollen.

Wenn Sie dieses Attribut angeben, werden die führenden und abschließenden Leerzeichen entfernt. Die verbleibende Zeichenfolge wird auf 32 Zeichen abgeschnitten und das Ergebnis wird an den Exit übergeben. Wenn Sie dieses Attribut ausschließen, wird der Standardwert (32 Leerzeichen) an den Exit übergeben.

Die maximale Länge dieses Felds beträgt 32 Zeichen.

**Sequence=Folgenummer**
Die Reihenfolge, in der dieser API-Exit in Relation zu anderen API-Exits aufgerufen wird. Ein Exit mit einer niedrigen Folgenummer wird vor einem Exit mit einer höheren Folgenummer aufgerufen. Die Folgenummern für die Exits müssen nicht fortlaufend vergeben werden. Die Folge '1, 2, 3' führt zu demselben Ergebnis wie die Folge '7, 42, 1096'. Wenn zwei Exits dieselbe Folgenummer aufweisen, entscheidet der Warteschlangenmanager, welcher Exit zuerst aufgerufen wird. Nach dem Ereignis können Sie feststellen, welcher Exit aufgerufen wurde, indem Sie die Uhrzeit oder einen Datenpunkt in 'ExitChainArea' (durch 'ExitChainAreaPtr' in MQAXP angegeben) eingeben oder eine eigene Protokolldatei schreiben.

Dieses Attribut ist ein numerischer Wert ohne Vorzeichen.

## Multi Zeilengruppe 'DefaultQueueManager' in der Datei 'mqs.ini'

Die Zeilengruppe DefaultQueueManager gibt den Standardwarteschlangenmanager für den Knoten an.

Verwenden Sie die Zeilengruppe DefaultQueueManager in der Datei `mqs.ini` , um den Standardwarteschlangenmanager anzugeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie IBM MQ Explorer Eigenschaftenseite von `General` IBM MQ verwenden.

**Name = *default_queue_manager***
Der Standardwarteschlangenmanager verarbeitet alle Befehle, für die kein Warteschlangenmanagername explizit angegeben wurde. Das Attribut **`DefaultQueueManager`** wird automatisch aktualisiert, wenn Sie einen neuen Standardwarteschlangenmanager erstellen. Wenn Sie versehentlich einen neuen Standardwarteschlangenmanager erstellen und dann zum ursprünglichen Warteschlangenmanager zurückkehren möchten, ändern Sie das Attribut **`DefaultQueueManager`** manuell.

## Multi Zeilengruppe 'ExitProperties' in der Datei 'mqs.ini'

Die Zeilengruppe ExitProperties gibt Konfigurationsoptionen an, die von Exitprogrammen des Warteschlangenmanagers verwendet werden.

Mithilfe der Zeilengruppe ExitProperties in der Datei `mqs.ini` können Sie Konfigurationsoptionen angeben, die von Exitprogrammen des Warteschlangenmanagers verwendet werden.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite IBM MQ Explorer `Extended` IBM MQ verwenden.

**CLWLMode = SAFE (Standardwert) | FAST**
Mit dem CLWL-Exit (CLWL-Cluster Workload) können Sie angeben, welche Clusterwarteschlange im Cluster als Antwort auf einen MQI-Aufruf geöffnet werden soll (z. B. `MQOPEN`, `MQPUT`). Der CLWL-Exit wird abhängig vom Wert, den Sie im Attribut **`CLWLMode`** angeben, entweder im FAST-Modus oder im SAFE-Modus ausgeführt. Wenn Sie das Attribut **`CLWLMode`** nicht angeben, wird der Exit für Clusterauslastung im SAFE-Modus ausgeführt.

**SAFE**
Führen Sie den CLWL-Exit in einem separaten Prozess aus dem Warteschlangenmanager aus. Dies ist die Standardeinstellung.

Tritt bei der Ausführung im SAFE-Modus ein Problem mit dem vom Benutzer geschriebenen CLWL-Exit auf, geschieht Folgendes:

- Der CLWL-Serverprozess (amqzlwa0) schlägt fehl.
- Der WS-Manager startet den CLWL-Serverprozess erneut.
- Der Fehler wird Ihnen im Fehlerprotokoll gemeldet. Wenn ein MQI-Aufruf in Bearbeitung ist, erhalten Sie eine Benachrichtigung in Form eines Rückkehrcodes.

Die Integrität des Warteschlangenmanagers bleibt erhalten.

**Anmerkung:** Die Ausführung des CLWL-Exits in einem separaten Prozess kann sich auf die Leistung auswirken.

**FAST**
Führen Sie den Cluster-Exit inline im WS-Manager-Prozess aus.

Wenn Sie diese Option angeben, wird die Leistung verbessert, da die Prozesse, die mit der Ausführung im SAFE-Modus verbunden sind, vermieden werden, dies jedoch zu Lasten der Integrität des Warteschlangenmanagers geht. Sie sollten den CLWL-Exit nur im FAST-Modus ausführen, wenn Sie überzeugt sind, dass es keine Probleme mit Ihrem CLWL-Exit gibt, und Sie sind besonders besorgt über die Leistung.

Tritt ein Problem auf, wenn der CLWL-Exit im FAST-Modus ausgeführt wird, schlägt der Warteschlangenmanager fehl, und Sie laufen Gefahr, dass die Integrität des Warteschlangenmanagers beeinträchtigt wird, der beeinträchtigt wird.

## Multi Zeilengruppe 'LogDefaults' in der Datei 'mqs.ini'

Die Zeilengruppe LogDefaults gibt Informationen zu Protokollstandardwerten für alle WS-Manager an.

Mithilfe der Zeilengruppe LogDefaults in der Datei `mqs.ini` können Sie Informationen zu Protokollstandardwerten für alle Warteschlangenmanager angeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite IBM MQ Explorer `Default log settings` IBM MQ verwenden.

Wenn Sie einen vom Standardwert abweichenden Wert benötigen, müssen Sie diesen Wert explizit in der Zeilengruppe LogDefaults angeben.

Wenn die Zeilengruppe LogDefaults nicht vorhanden ist, werden die IBM MQ -Standardeinstellungen verwendet. Die Protokollattribute werden bei der Erstellung eines Warteschlangenmanagers als Standardwerte verwendet, können jedoch überschrieben werden, wenn Sie die Protokollattribute im Befehl **crtmqm** angeben. Weitere Informationen zu diesem Befehl finden Sie in **crtmqm**.

Nachdem ein Warteschlangenmanager erstellt wurde, werden die Protokollattribute für diesen WS-Manager aus den Einstellungen übernommen, die in „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140beschrieben sind.

**Anmerkung:** Die bereitgestellte Zeilengruppe LogDefaults für eine neue IBM MQ -Installation enthält keine expliziten Werte für die Attribute. Das Fehlen eines Attributs bedeutet, dass der Standardwert für diesen Wert bei der Erstellung eines neuen Warteschlangenmanagers verwendet wird. Die Standardwerte für die Zeilengruppe LogDefaults werden in „Beispieldatei mqs.ini für AIX and Linux" auf Seite 92 und „Beispieldatei mqs.ini für Windows" auf Seite 93gezeigt. Der Wert null für das Attribut `LogBufferPages` bedeutet 512.

Das Standardpräfix, das in „Zeilengruppe 'AllQueueManagers' in der Datei 'mqs.ini'" auf Seite 96angegeben ist, und der Protokollpfad, der für den bestimmten Warteschlangenmanager angegeben ist, der in „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140angegeben ist, ermöglichen es, dass sich der Warteschlangenmanager und sein Protokoll auf verschiedenen physischen Laufwerken befinden. Dies ist die empfohlene Methode, obwohl sie sich standardmäßig auf demselben Laufwerk befinden.

Informationen zum Berechnen der Protokollgrößen finden Sie in „Berechnen der Größe des Protokolls" auf Seite 698.

**Anmerkung:** Die in der folgenden Parameterliste angegebenen Grenzwerte sind Grenzwerte, die von IBM MQ festgelegt werden. Durch die Begrenzung des Betriebssystems kann die maximal mögliche Protokollgröße reduziert werden.

**LogPrimaryFiles = 3 (Standardwert) |*2-254* (Windows) |*2-510* (AIX and Linux)**
Die Protokolldateien, die beim Erstellen des Warteschlangenmanagers zugeordnet werden.

Die minimale Anzahl der primären Protokolldateien, die Sie haben können, ist 2 und das Maximum ist 254 auf Windows oder 510 auf AIX and Linux. Der Standardwert ist 3.

Die Gesamtzahl der primären und sekundären Protokolldateien darf 255 auf Windows oder 511 in AIX and Linux nicht überschreiten und darf nicht kleiner als 3 sein.

Der Wert wird geprüft, wenn der WS-Manager erstellt oder gestartet wird. Sie können sie ändern, nachdem der WS-Manager erstellt wurde. Eine Änderung des Werts ist jedoch erst wirksam, wenn der Warteschlangenmanager erneut gestartet wird und der Effekt möglicherweise nicht sofort ausgeführt wird.

**LogSecondary-Dateien = 2 (Standardwert) |*1-253* (Windows) |*1-509* (AIX and Linux)**
Die Protokolldateien, die zugeordnet werden, wenn die Primärdateien erschöpft sind.

Die minimale Anzahl an sekundären Protokolldateien ist 1 und das Maximum ist 253 bei Windowsoder 509 auf AIX and Linux. Die Standardanzahl ist 2.

Die Gesamtzahl der primären und sekundären Protokolldateien darf 255 auf Windows oder 511 in AIX and Linux nicht überschreiten und darf nicht kleiner als 3 sein.

Der Wert wird geprüft, wenn der Warteschlangenmanager gestartet wird. Sie können diesen Wert ändern, aber Änderungen werden erst wirksam, wenn der Warteschlangenmanager erneut gestartet wird, und selbst dann wird der Effekt möglicherweise nicht sofort wirksam.

**LogFilePages= *number***
Die Protokolldaten werden in einer Reihe von Dateien mit dem Namen "Protokolldateien" festgehalten. Die Protokolldateigröße wird in Einheiten von 4-KB-Seiten angegeben.

Die Standardanzahl der Protokolldateiseiten beträgt 4096, wobei eine Protokolldateigröße von 16 MB angegeben wird.

Unter AIX and Linuxist die Mindestanzahl der Protokolldateiseiten 64, und unter Windowsist die minimale Anzahl von Protokolldateiseiten 32; in beiden Fällen beträgt die maximale Anzahl 65 535.

**Anmerkung:** Die Größe der Protokolldateien, die bei der Erstellung des Warteschlangenmanagers angegeben wurden, kann für einen Warteschlangenmanager nicht geändert werden.

**LogType = CIRCULAR (Standardwert) | LINEAR**
Der Typ des zu verwendenden Protokolls. Der Standardwert ist CIRCULAR.

**CIRCULAR**
Starten Sie die Wiederherstellung nach einem Neustart mit Hilfe des Protokolls, um Transaktionen rückgängig zu machen, die sich in Bearbeitung befanden, als das System gestoppt wurde

Eine genauere Beschreibung des Protokolltyps CIRCULAR (Umlaufprotokollierung) finden Sie im Abschnitt „Typen der Protokollierung” auf Seite 692.

**LINEAR**
Sowohl für die Wiederherstellung nach einem Neustart als auch für die Datenträger-oder Vorwärtswiederherstellung (durch das Erstellen verlorener oder beschädigter Daten durch Wiedergabe des Inhalts des Protokolls).

Eine genauere Beschreibung des Protokolltyps LINEAR (lineare Protokollierung) finden Sie im Abschnitt „Typen der Protokollierung” auf Seite 692.

Wenn Sie den Standardwert ändern möchten, können Sie entweder das Attribut LogType bearbeiten oder die lineare Protokollierung mit dem Befehl **crtmqm** angeben.

Ab IBM MQ 9.1.0 können Sie die Protokollierungsmethode nach der Erstellung eines Warteschlangenmanagers ändern. Weitere Informationen finden Sie unter migmqlog.

**LogBufferPages=0 (Standardwert) |*0-4096***
Die Größe des Speichers, der den Pufferdatensätzen für das Schreiben zugeordnet ist, wobei die Größe der Puffer in Einheiten von 4-KB-Seiten angegeben wird.

Die Mindestanzahl der Pufferseiten beträgt 18 und der Maximalwert 4096. Größere Puffer führen zu einem höheren Durchsatz, insbesondere bei größeren Nachrichten.

Wenn Sie 0 (Standardwert) angeben, wählt der Warteschlangenmanager die Größe 512 (2048 KB) aus.

Wenn Sie eine Zahl im Bereich von 1 bis 17 angeben, nimmt der WS-Manager standardmäßig den Wert 18 (72 KB) an. Wenn Sie eine Zahl im Bereich von 18 und 4096 angeben, verwendet der Warteschlangenmanager die angegebene Zahl, um den zugeordneten Speicher festzulegen.

**LogDefaultPath= *directory_name***
Das Verzeichnis, in dem sich die Protokolldateien für einen WS-Manager befinden. Das Verzeichnis befindet sich auf einer lokalen Einheit, in die der Warteschlangenmanager schreiben kann, und zwar vorzugsweise auf einem anderen Laufwerk aus den Nachrichtenwarteschlangen. Die Angabe eines anderen Laufwerks bietet einen zusätzlichen Schutz im Falle eines Systemausfalls.

Der Standardwert lautet:

- Windows *DefaultPrefix*\log für IBM MQ for Windows , wobei *DefaultPrefix* der Wert ist, der im Attribut DefaultPrefix auf der Eigenschaftenseite von All Queue Managers IBM MQ angegeben ist. Dieser Wert wird zur Installationszeit festgelegt.
- Linux AIX /var/mqm/log für AIX and Linux -Systeme.

Alternativ können Sie den Namen eines Verzeichnisses im Befehl **crtmqm** mit dem Flag **-ld** angeben. Wenn ein Warteschlangenmanager erstellt wird, wird auch ein Verzeichnis unter dem WS-Manager-Verzeichnis erstellt, und dieses Verzeichnis wird verwendet, um die Protokolldateien zu speichern. Der Name dieses Verzeichnisses basiert auf dem Namen des Warteschlangenmanagers. Dadurch wird sichergestellt, dass der Protokolldateipfad eindeutig ist und dass er auch Einschränkungen bezüglich der Länge von Verzeichnisnamen entspricht.

Wenn Sie **-ld** im **crtmqm** -Befehl nicht angeben, wird der Wert des Attributs **LogDefaultPath** in der Datei mqs.ini verwendet.

Der Name des Warteschlangenmanagers wird an den Verzeichnisnamen angehängt, um sicherzustellen, dass mehrere WS-Manager unterschiedliche Protokollverzeichnisse verwenden.

Wenn der Warteschlangenmanager erstellt wird, wird in den Protokollattributen in den Konfigurationsinformationen ein **LogPath** -Wert erstellt, der den vollständigen Verzeichnisnamen für das Protokoll des Warteschlangenmanagers enthält. Dieser Wert wird verwendet, um das Protokoll zu lokalisieren, wenn der Warteschlangenmanager gestartet oder gelöscht wird.

**LogWriteIntegrity =SingleWrite|DoubleWrite|TripleWrite (Standardwert)**
Die Methode, die die Protokollfunktion verwendet, um Protokollsätze zuverlässig zu schreiben.

**TripleWrite (Standardwert)**

Beachten Sie, dass Sie DoubleWrite auswählen können, aber wenn Sie dies tun, interpretiert das System diese Option als TripleWrite .

**SingleWrite**
Sie sollten SingleWrite nur verwenden, wenn das Dateisystem und die Einheit, die als Host für das IBM MQ-Wiederherstellungsprotokoll fungiert, explizit die Atomizität von 4-KB-Schreibvorgängen garantiert.

Das heißt, wenn ein Schreiben einer 4-KB-Seite aus irgendeinem Grund fehlschlägt, sind die einzigen beiden möglichen Status entweder das Vorimage oder das Nachimage. Ein Zwischenzustand sollte nicht möglich sein.

**Anmerkung:** Wenn in Ihrer persistenten Workload ausreichend gemeinsamer Zugriff vorhanden ist, ist bei der Einstellung eines anderen Werts als dem Standardwert TripleWrite ein minimaler potenzieller Vorteil vorhanden.

Weitere Informationen finden Sie unter „LogWriteIntegrity-using SingleWrite oder TripleWrite” auf Seite 144.

## Multi Zeilengruppe 'QueueManager' in der Datei 'mqs.ini'

Die Zeilengruppe QueueManager gibt die Position des Warteschlangenmanagerverzeichnisses an.

Für jeden Warteschlangenmanager gibt es eine Zeilengruppe QueueManager . Die Attribute dieser Zeilengruppe geben den Namen des Warteschlangenmanagers und den Namen des Verzeichnisses mit den Dateien an, die diesem Warteschlangenmanager zugeordnet sind. Der Name des Verzeichnisses basiert auf dem Namen des Warteschlangenmanagers, wird aber umgesetzt, wenn der Name des WS-Managers kein gültiger Dateiname ist. Weitere Informationen zur Namensumsetzung finden Sie unter Informationen zu IBM MQ -Dateinamen.

**Name =** ***queue_manager_name***
Der Name des Warteschlangenmanagers.

**Präfix =** ***prefix***
Gibt an, wo die Warteschlangenmanagerdateien gespeichert sind. Standardmäßig entspricht dieser Wert dem Wert, der im Attribut **DefaultPrefix** der Zeilengruppe Alle Warteschlangenmanager in der Datei `mqs.ini` angegeben ist.

**Verzeichnis =** ***name***
Der Name des Unterverzeichnisses unter dem Verzeichnis *prefix*\QMGRS, in dem die Warteschlangenmanager-Dateien gespeichert sind. Dieser Name basiert auf dem Namen des Warteschlangenmanagers, kann aber umgesetzt werden, wenn ein doppelter Name vorhanden ist oder wenn der Name des WS-Managers kein gültiger Dateiname ist.

**Datenpfad=** ***path***
Ein expliziter Datenpfad, der bei der Erstellung des Warteschlangenmanagers angegeben wurde, überschreibt **Prefix** und **Directory** als Pfad zu den Warteschlangenmanagerdaten.

**InstallationName=** ***name***
Der Name der IBM MQ-Installation, die diesem Warteschlangenmanager zugeordnet ist. Befehle aus dieser Installation müssen bei der Interaktion mit diesem WS-Manager verwendet werden.

IBM i **Bibliothek =** ***name***
Der Name der Bibliothek, in der für diesen Warteschlangenmanager relevante IBM i-Objekte, z. B. Journale und Journalempfänger, gespeichert werden. Dieser Name basiert auf dem Namen des Warteschlangenmanagers, kann aber umgesetzt werden, wenn ein doppelter Name vorhanden ist, oder wenn der Name des WS-Managers kein gültiger Bibliotheksname ist.

**EphemeralPrefix=** ***Name***
Speicherort der temporären Daten des Warteschlangenmanagers.

Standardmäßig ist dieser Wert nicht vorhanden, d. h. die Daten werden unter der Präfixposition gespeichert.

Der Wert wird über den Wert der Umgebungsvariablen **MQ_EPHEMERAL_PREFIX** oder über das Attribut **DefaultEphemeralPrefix** der Zeilengruppe AllQueueManagers in der Datei `mqs.ini` festgelegt, wenn der Warteschlangenmanager erstellt wird.

IBM i Aufgrund von Beschränkungen seitens des Betriebssystems ist das standardmäßige temporäre Präfix unter IBM i auf 24 Zeichen beschränkt.

**Zugehörige Tasks**

„WS-Manager einer Installation zuordnen” auf Seite 499
Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den **crtmqm** -Befehl ausgegeben hat. Unter AIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

## Windows Advanced Configuration and Power Interface (ACPI)

Windows unterstützt den ACPI-Standard (Advanced Configuration and Power Interface). Dies ermöglicht es Windows-Benutzern mit ACPI-fähiger Hardware, Kanäle zu stoppen und erneut zu starten, wenn das System in den Aussetzungsmodus wechselt und anschließend den Betrieb wieder aufnimmt.

Auf der Eigenschaftenseite von ACPI IBM MQ in der IBM MQ Explorer können Sie angeben, wie sich IBM MQ verhalten soll, wenn das System eine Aussetzanforderung empfängt.

Beachten Sie, dass die Einstellungen, die auf der Eigenschaftenseite von ACPI IBM MQ angegeben sind, nur angewendet werden, wenn der Alertmonitor ausgeführt wird. Wenn der Alertmonitor ausgeführt wird, ist das Symbol Alertmonitor in der Taskleiste vorhanden.

**DoDialog= Y | N**
Zeigt den Dialog zum Zeitpunkt der Aussetzungsanforderung an.

**DenySuspend=Y | N**
Verweigert die Aussetzungsanforderung. Dies wird verwendet, wenn DoDialog = N ist, oder wenn DoDialog=Y und ein Dialog nicht angezeigt werden kann, z. B., weil der Notizbuchdeckel geschlossen ist.

**CheckChannelsRunning=Y | N**
Prüft, ob Kanäle aktiv sind. Das Ergebnis kann das Ergebnis der anderen Einstellungen bestimmen.

In der folgenden Tabelle sind die Auswirkungen der einzelnen Kombinationen dieser Parameter aufgeführt:

| DoDialog | DenySuspend | Aktive CheckChannels | Action |
|---|---|---|---|
| N | N | N | Akzeptieren Sie die Aussetzungsanforderung. |
| N | N | Y | Akzeptieren Sie die Aussetzungsanforderung. |
| N | Y | N | Die Aussetzanforderung zurückweisen. |
| N | Y | Y | Wenn Kanäle aktiv sind, wird die Aussetzungsanforderung verweigert. Wenn die Anforderung nicht akzeptiert wird. |
| Y | N | N | Zeigen Sie den Dialog an (siehe Anmerkung ; akzeptieren Sie die Aussetzungsanforderung). Dies ist die Standardeinstellung. |
| Y | N | Y | Wenn keine Kanäle aktiv sind, akzeptieren Sie die Aussetzungsanforderung. Wenn sie den Dialog anzeigen (siehe Anmerkung ), akzeptieren Sie die Anforderung. |
| Y | Y | N | Den Dialog anzeigen ( Hinweis ; die Aussetzungsanforderung verweigern). |
| Y | Y | Y | Wenn keine Kanäle aktiv sind, akzeptieren Sie die Aussetzungsanforderung; wenn sie den Dialog anzeigen ( Hinweis ; verweigern Sie die Anforderung). |

**Anmerkung:** Wenn in den Fällen, in denen der Dialog angezeigt werden soll, der Dialog nicht angezeigt werden kann (z. B. weil die Notizbuch-ID geschlossen ist), wird die Option 'DenySuspend' verwendet, um zu ermitteln, ob die Aussetzungsanforderung akzeptiert oder verweigert wird.

## Multi Warteschlangenmanagerkonfigurationsdateien, qm.ini

Eine WS-Manager-Konfigurationsdatei, `qm.ini`, enthält Informationen, die für einen bestimmten Warteschlangenmanager relevant sind. Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

Für jeden Warteschlangenmanager gibt es eine WS-Manager-Konfigurationsdatei. Die Datei `qm.ini` wird automatisch erstellt, wenn der Warteschlangenmanager, dem sie zugeordnet ist, erstellt wird.

**Anmerkung:** Weitere Informationen dazu, wie und wann eine `qm.ini` -Datei bearbeitet wird und wann Änderungen, die Sie an der Datei vornehmen, wirksam werden, finden Sie unter „IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern" auf Seite 90.

Ab IBM MQ 9.0.4 und IBM MQ 9.0.0 Fix Pack 2 überprüft der Befehl **strmqm** die Syntax der Zeilengruppen CHANNELS und SSL in der Datei `qm.ini`, bevor der Warteschlangenmanager vollständig gestartet wird. Dies macht es wesentlich einfacher zu sehen, was falsch ist, und schnell Korrekturen vorzunehmen, wenn **strmqm** feststellt, dass die Datei `qm.ini` Fehler enthält. Weitere Informationen finden Sie im Abschnitt strmqm .

## Position der `qm.ini`-Dateien

Linux AIX Auf AIX and Linux-Systemen wird eine `qm.ini`-Datei im Stammverzeichnis der Verzeichnisbaumstruktur gespeichert, die vom Warteschlangenmanager belegt wird. Der Pfad und der Name einer Konfigurationsdatei für einen WS-Manager mit dem Namen QMNAME sind beispielsweise:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows Auf Windows-Systemen wird die Position der Datei `qm.ini` durch den Arbeitspfad angegeben, der im Schlüssel `HKLM\SOFTWARE\IBM\WebSphere MQ` angegeben ist. Der Pfad und der Name für eine Konfigurationsdatei für einen Warteschlangenmanager mit dem Namen QMNAME sind beispielsweise wie folgt:

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

IBM i Eine `qm.ini`-Datei wird in der *mqmdata directory*`/`*QMNAME*`/qm.ini` gespeichert, wobei *mqmdata directory* standardmäßig `/QIBM/UserData/mqm` ist und *QMNAME* der Name des Warteschlangenmanagers ist, für den die Initialisierungsdatei gilt.

**Anmerkung:** Sie können die *mqmdata directory* in der `mqs.ini` -Datei ändern.

Der Name des WS-Managers kann bis zu 48 Zeichen lang sein. Dies garantiert jedoch nicht, dass der Name gültig oder eindeutig ist. Daher wird basierend auf dem Namen des Warteschlangenmanagers ein Verzeichnisname generiert. Dieser Prozess wird als *Namensumsetzung* bezeichnet. Eine Beschreibung finden Sie unter IBM MQ -Dateinamen und -Objektnamen unter IBM i.

## `qm.ini` Zeilengruppen

**Achtung:**

- Dieser Abschnitt enthält Links zu weiteren Informationen zu den Zeilengruppen in der Datei `qm.ini` . Jede Zeilengruppe enthält Informationen zu den Parametern in dieser Zeilengruppe, gegebenenfalls einschließlich eines Beispiels.
- Jede Zeilengruppe zeigt die Plattform bzw. Plattformen von IBM MQ for Multiplatforms an, für die diese Zeilengruppe gilt.

## Multi Automatische Konfiguration von 'qm.ini' beim Start

Ab IBM MQ 9.2.0 können Sie Ihren Warteschlangenmanager so konfigurieren, dass der Inhalt einer Datei oder einer Gruppe von Dateien, die `qm.ini`-Überschreibungen enthält, bei jedem Warteschlangenmanager-Start automatisch angewendet wird.

Mithilfe dieser Funktion kann eine Konfiguration geändert und beim nächsten Warteschlangenmanagerneustart automatisch wiederholt werden. Wenn sich die `qm.ini`-Überschreibungen beispielsweise auf einem angehängten Laufwerk befinden, ist es möglich, eine zentrale Konfiguration zu haben, bei der die neueste Version auf jeden Warteschlangenmanager angewendet wird, wenn sie gestartet werden.

Mit dieser Funktion können Sie das Erstellen eines einheitlichen Clusters durch die Verwendung der automatischen Clusterfunktionalität vereinfachen. Ein Beispiel hierfür finden Sie in „Neuen Uniform-Cluster erstellen” auf Seite 459.

**Anmerkung:** Diese Überschreibungen werden nur beim Start des Warteschlangenmanagers angewendet und haben keine Auswirkung auf die Erstellung des Warteschlangenmanagers. Sie können beispielsweise nicht die Anzahl der primären Protokolldateien mit dieser Funktion festlegen.

## Vorbereitungen

Sie können Folgendes verwenden:

1. Eine einzelne Datei und eine Textdatei, die Änderungen an der `qm.ini`-Datei enthält.
2. Eine Gruppe von `qm.ini`-Formatdateien:
   - um ein Verzeichnis anzugeben, in dem die Konfigurationen vorhanden sein sollen
   - Erstellen Sie in diesem Verzeichnis Dateien mit der Erweiterung `.ini`, z. B. `qminisettings.ini`.

Die Datei oder Dateien müssen nur die Zeilengruppe und die Einstellungen **attribute**=*value* für die Elemente enthalten, die sich ändern. Um das Attribut **MaxChannels** in der Zeilengruppe 'Channels' zu aktualisieren, kann die Datei beispielsweise Folgendes enthalten:

```
Channels:
 MaxChannels=1234
```

Beachten Sie, dass alle Zeilen in Überschreibungen der Datei `qm.ini`, die das Präfix # haben, als Kommentar behandelt werden.

## Automatische Konfiguration von qm.ini-Dateiattributen aktivieren

Sie können einen neuen Warteschlangenmanager konfigurieren, indem Sie das Flag **-ii** im Befehl **crtmqm** angeben, das entweder auf eine bestimmte Datei oder ein bestimmtes Verzeichnis zeigt. Der bereitgestellte Wert wird in der Datei `qm.ini` unter der Zeilengruppe **AutoConfig** als Attribut **IniConfig**gespeichert.

Sie können einen vorhandenen Warteschlangenmanager konfigurieren, um die automatische MQSC-Konfiguration zu aktivieren, indem Sie das **AutoConfig** Zeilengruppenattribut **IniConfig**hinzufügen, das auf eine gültige Datei oder ein gültiges Verzeichnis verweist. For example:

```
AutoConfig:
   IniConfig=C:\MQ_Configuration\uniclus.ini
```

## Wie funktioniert die automatische Konfiguration?

Beim Start des Warteschlangenmanagers wird die Konfiguration, die durch das AutoConfig -Zeilengruppenattribut **IniConfig** angegeben wird, validiert, um eine gültige Syntax sicherzustellen, und anschließend in der Datenbaumstruktur des Warteschlangenmanagers im Verzeichnis `autocfg` als einzelne `cached.ini` -Datei gespeichert.

Wenn mehrere Dateien aus einem Verzeichnis verarbeitet werden, geschieht dies in alphabetische Reihenfolge.

Wenn beim ersten Start des Warteschlangenmanagers die Datei oder das Verzeichnis nicht gelesen werden können, wird verhindert, dass der Warteschlangenmanager starten kann, und es wird eine entsprechende Fehlernachricht in die Konsole und in das Fehlerprotokoll des Warteschlangenmanagers geschrieben.

Wenn die Datei oder das Verzeichnis, auf die bzw. auf das verwiesen wird, bei nachfolgenden Neustarts nicht lesbar ist, wird die zuvor zwischengespeicherte Datei verwendet und es wird eine Nachricht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben, in der dies angezeigt wird.

Wenn Sie den Befehl **strmqm** verwenden, werden die Inhalte der Datei `cached.ini` für die Datei `qm.ini` als Überschreibungen angewendet, bevor der Warteschlangenmanager aufgerufen wird.

Dies bedeutet, dass die Einstellungen für einen Standby-Warteschlangenmanager gelesen werden, wenn der Befehl **strmqm** verarbeitet wird, und nicht, wenn der Warteschlangenmanager aktiv wird.

## Wie wird die Ersatzdatei 'qm.ini' erstellt?

Beim ersten Mal, dass die automatische Initialisierungskonfiguration konfiguriert ist und der Warteschlangenmanager gestartet wird, wird eine Kopie der aktuellen `qm.ini`-Datei in das Unterverzeichnis `autoconfig` im Datenverzeichnis des Warteschlangenmanagers als `base_qm.ini`kopiert. Diese Datei wird ab diesem Punkt als Referenzversion betrachtet.

Bei jedem Start des Warteschlangenmanagers, d. h. bei **strmqm** , wird die derzeit aktive Datei `qm.ini` gelöscht und durch eine Kopie von `base_qm.ini`ersetzt. Anschließend wird die Konfiguration aus der `cached.ini`-Datei auf diese Datei angewendet.

Sobald ein Warteschlangenmanager unter der automatischen Konfigurationssteuerung steht, sollten alle Änderungen an der Datei `qm.ini` über die Datei (bzw. die Dateien) vorgenommen werden, auf die über das Attribut **IniConfig** in der Zeilengruppe 'AutoConfig' verwiesen wird.

Da eine vorhandene `qm.ini`-Datei beim Start des Warteschlangenmanagers gelöscht wird, wird nur die Konfiguration in der angegebenen `qm.ini`-Datei mithilfe des Attributs **IniConfig** auf die Referenzversion des Warteschlangenmanagers angewendet.

Wenn eine Zeilengruppe oder ein Attribut über die Konfiguration der automatischen Initialisierung bei früheren Starts des Warteschlangenmanagers geändert wurde, werden diese Änderungen verworfen, es sei denn, sie werden weiterhin in der durch das Attribut **IniConfig** angegebenen Datei (bzw. den durch dieses Attribut angegebenen Dateien) ermittelt.

Aufgrund der Neuerstellung der Datei `qm.ini` beim Start des Warteschlangenmanagers bedeutet dies, dass manuelle Änderungen an der `qm.ini`-Datei verloren gehen. Wenn Sie eine persistente Änderung vornehmen müssen und für diese Änderung nicht das Attribut **IniConfig** verwenden können, haben Sie die folgenden Möglichkeiten:

- Nehmen Sie die Änderung an der `base_qm.ini`-Datei selbst vor.
- Löschen Sie die Datei `base_qm.ini`.

  Wenn Sie diese Datei löschen, wird die `base_qm.ini` beim nächsten Start des Warteschlangenmanagers erneut erstellt, basierend auf dem aktuellen Inhalt der `qm.ini`-Datei. Dadurch werden die aktuellen Änderungen als neue Referenzversion für zukünftige Startvorgänge *permanent gespeichert*.

**Zugehörige Konzepte**

„Zusammenfassung der Zeilengruppen und Attribute der Datei qm.ini” auf Seite 107
Eine Zusammenfassung der Attribute der Zeilengruppen der WS-Manager-Konfigurationsdatei `qmi.ini` mit Links zu weiteren Informationen.

## Multi Zusammenfassung der Zeilengruppen und Attribute der Datei qm.ini

Eine Zusammenfassung der Attribute der Zeilengruppen der WS-Manager-Konfigurationsdatei `qmi.ini` mit Links zu weiteren Informationen.

*Tabelle 11. Zeilengruppen der Datei 'qm.ini'*

| Zeilengruppen und Attribute | Beschreibung der Attribute |
|---|---|
| Windows **Zeilengruppe 'AccessMode'** | |
| Windows Zugriffsgruppe [1] | Eine Windows-Sicherheitsgruppe, deren Mitglieder vollständigen Zugriff auf alle Dateien mit Warteschlangenmanagerdaten erhalten sollen. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| **ApiExitLocal, Zeilengruppe** | |
| Name | Der beschreibende Name des API-Exits, der im Feld 'ExitInfoName' der MQAXP-Struktur übergeben wird. |
| FUNCTION | Der Name des Einstiegspunkts der Funktion in dem Modul, das den API-Exitcode enthält. |
| Modul | Das Modul, das den API-Exitcode enthält. |
| Daten | Die Daten, die an den API-Exit im Feld 'ExitData' der MQAXP-Struktur übergeben werden sollen. |
| Sequenz | Die Reihenfolge, in der dieser API-Exit in Relation zu anderen API-Exits aufgerufen wird. |
| V 9.3.4 Linux AIX **ZeilengruppeAuthToken** | |
| KeyStore | Dateipfad für den Keystore, der die öffentlichen Schlüsselzertifikate oder symmetrischen Schlüssel des vertrauenswürdigen Ausstellers enthält |
| KeyStorePwdFile | Dateipfad für die Datei, die das verschlüsselte Kennwort für den Schlüsselspeicher enthält |
| CertLabel | Die Zertifikatsbezeichnung für ein Zertifikat mit öffentlichem Schlüssel oder einen symmetrischen Schlüssel im Keystore, der für die Validierung von Authentifizierungstoken verwendet wird. |
| UserClaim | Anforderung in dem Token, das Benutzeridentitätsinformationen enthält, die der Warteschlangenmanager für Berechtigungsprüfungen übernehmen kann |
| AllowOSGroups | Dieses Attribut legt fest, ob die Gruppenzugehörigkeit für den übernommenen Benutzer überprüft wird. |
| **Zeilengruppe 'AutoCluster'** | |
| Typ | Der Typ des automatischen Clusters. Die einzige gültige Option ist 'Uniform', die für einen einheitlichen Cluster steht. |
| ClusterName | Der Name des automatischen Clusters. |
| RepositoryName1 | Der Warteschlangenmanagername für das erste vollständige Repository im automatischen Cluster. |
| Repository1Conname | Der Verbindungsname (CONNAME). Dieser Wert gibt an, wie die Verbindung zwischen den Mitgliedern des automatischen Clusters und dem Warteschlangenmanager hergestellt wird. |
| RepositoryName2 | Der Warteschlangenmanagername für das zweite vollständige Repository im automatischen Cluster. |
| Repository2Conname | Der Verbindungsname (CONNAME). Dieser Wert gibt an, wie die Verbindung zwischen den Mitgliedern des automatischen Clusters und dem Warteschlangenmanager hergestellt wird. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| **Zeilengruppe 'AutoConfig'** | |
| MQSCConfig | Entweder ein vollständiger Dateipfad oder Pfad zu einem Verzeichnis, in dem alle `*.mqsc`-Dateien auf jedem Warteschlangenmanager-Start auf den Warteschlangenmanager angewendet werden. |
| IniConfig | Entweder ein vollständiger Dateipfad oder Pfad zu einem Verzeichnis, in dem die gesamte `*.ini`-Datei auf jedem Warteschlangenmanager-Start auf die Datei `qm.ini` angewendet wird. |
| **Zeilengruppe 'Channels'** | |
| MaxChannels | Die maximale Anzahl der zulässigen aktuellen Kanäle. |
| MaxActiveChannels | Die maximale Anzahl der Kanäle, die jeweils aktiv sein können. |
| MaxInitiators | Die maximale Anzahl der Initiatoren. |
| MQIBindType | Die Bindung für Anwendungen. |
| PipeLineLength | Die maximale Anzahl gleichzeitiger Threads, die ein Kanal verwenden wird. |
| AdoptNewMCA | Gibt an, welche Kanaltypen die vorhandene Kanalinstanz stoppen können, damit eine neue Kanalinstanz gestartet werden kann, wenn IBM MQ eine Anforderung zum Starten eines Kanals empfängt, aber feststellt, dass bereits eine Instanz des Kanals ausgeführt wird. |
| AdoptNewMCATimeout | Die Zeit in Sekunden, die die neue Kanalinstanz wartet, bis die alte Kanalinstanz beendet wird. |
| AdoptNewMCACheck | Der Typ der Überprüfung, die erforderlich ist, wenn das Attribut **AdoptNewMCA** aktiviert wird. |
| ChlauthEarlyAdopt | Die Reihenfolge, in der die Authentifizierungsregeln für die Verbindung und den Kanal verarbeitet werden. |
| PasswordProtection | Gibt an, ob die von einer Anwendung angegebenen Berechtigungsnachweise durch MQCSP-Kennwortschutz geschützt werden müssen, wenn der Kanal keine TLS-Verschlüsselung verwendet. |
| IgnoreSeqNumberMismatch | Steuert, wie der Warteschlangenmanager bei einer Folgenummernabweichung während des Kanalstarts vorgeht. |
| **Zeilengruppe 'Connection'** | |
| DefaultBindType | Gibt an, ob Anwendungen und der Warteschlangenmanager, die in separaten Prozessen ausgeführt werden, einige Ressourcen gemeinsam nutzen. |
| **Zeilengruppe 'DiagnosticMessages'** | |
| name | Der Name einer Zeilengruppe. |
| Service | Ein Service, der mit dieser Zeilengruppe aktiviert wird. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| ExcludeMessage | Nachrichten, die nicht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden sollen. |
| SuppressMessage | Nachrichten, die in nur einmal in einem angegebenen Zeitintervall in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden sollen. |
| IBM i SuppressInterval | Das Zeitintervall (in Sekunden), in dem Nachrichten, die in **SuppressMessage** angegeben sind, nur einmal in das Fehlerprotokoll des Warteschlangenmanager geschrieben werden. |
| Severities | Eine durch Kommas getrennte Liste der Bewertungsstufen. |
| FilePath | Der Pfad zu der Position, in die Protokolldateien geschrieben wurden. (Nur unterstützt, wenn das Attribut 'Service' auf 'File' gesetzt ist.) |
| FilePrefix | Das Präfix der Protokolldateien. (Nur unterstützt, wenn das Attribut 'Service' auf 'File' gesetzt ist.) |
| FileSize | Die Größe, in der sich das Protokoll überrollt. (Nur unterstützt, wenn das Attribut 'Service' auf 'File' gesetzt ist.) |
| Format | Das Format der Datei. (Nur unterstützt, wenn das Attribut 'Service' auf 'File' gesetzt ist.) |
| Linux AIX Syslog | Der Syslog-Service, der alle ungefilterten Nachrichten mithilfe der Spezifikation für Diagnosenachrichten im JSON-Format sendet. |
| Linux AIX Ident | Der Wert für 'ident', der den Syslog-Einträgen zugeordnet ist. (Nur unterstützt, wenn das Attribut 'Service' auf 'Syslog' gesetzt ist.) |
| **ExitPath, Zeilengruppe** | |
| ExitsDefaultPath | Der Pfad für Benutzerexitprogramme auf dem Warteschlangenmanagersystem (32 Bit). |
| ExitsDefaultPath64 | Der Pfad für Benutzerexitprogramme auf dem Warteschlangenmanagersystem (64 Bit). |
| **Zeilengruppe 'ExitPropertiesLocal'** | |
| CLWLMode | Gibt an, ob der CLWL-Exit (Cluster Workload) im Modus FAST oder SAFE ausgeführt wird. |
| Linux IBM i AIX **Zeilengruppe 'Filesystem'** | |
| Linux IBM i AIX ValidateAuth | Ermöglicht Benutzern, die nicht Mitglieder der Gruppe `mqm` sind, den Zugriff auf Fehlerverzeichnisse und Dateien. |
| **Zeilengruppe 'Log'** | |
| LogPrimaryFiles | Die Protokolldateien, die beim Erstellen des Warteschlangenmanagers zugeordnet werden. |
| LogSecondaryFiles | Die Protokolldateien, die zugeordnet werden, wenn die Primärdateien erschöpft sind. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| LogFilePages | Die Anzahl der Seiten in der Protokolldatei. (Die Protokolldateigröße wird in Einheiten von 4-KB-Seiten angegeben.) |
| LogType | Der Typ der Protokollierung, die vom Warteschlangenmanager verwendet werden soll (Umlaufprotokollierung oder lineare Protokollierung). |
| LogBufferPages | Die Größe des Speichers, der den Pufferdatensätzen für das Schreiben zugeordnet ist, wobei die Größe der Puffer in Einheiten von 4-KB-Seiten angegeben wird. |
| LogPath | Das Verzeichnis, in dem sich die Protokolldateien für einen WS-Manager befinden. |
| LogWriteIntegrity | Die Methode, die die Protokollfunktion verwendet, um Protokollsätze zuverlässig zu schreiben. |
| LogManagement | Die Methode, mit der Protokollextents entweder manuell oder vom Warteschlangenmanager verwaltet werden. |
| Windows **Zeilengruppe 'LU62'** | |
| Windows TPName | Der Name des TP-Namens, der auf dem fernen Standort gestartet werden soll. |
| Windows Library1 | Der Name der APPC-DLL. |
| Windows Library2 | Wie Library1, wird verwendet, wenn der Code in zwei separaten Bibliotheken gespeichert ist. |
| CP4I **NativeHAInstance stanza** | |
| „Name” auf Seite 145 | Der Instanzname, der bei der Erstellung der WS-Manager-Instanz verwendet wurde. |
| „ReplicationAddress” auf Seite 145 | Der Hostname, IPv4 in der Schreibweise mit Trennzeichen, oder IPv6 Hexadezimalformat-Instanzadresse. |
| CP4I **NativeHALocalInstance stanza** | |
| „LocalName” auf Seite 146 | Der Name der Zeilengruppe 'NativeHALocalInstance', die aus dem Namen der Protokollreplikatinstanz übernommen wurde, der bei der Erstellung des nativen HA-Warteschlangenmanagers angegeben wurde. |
| „KeyRepository” auf Seite 146 | Die Position des Schlüsselrepositorys, das das digitale Zertifikat enthält, das für den Schutz des Protokollreplikationsverkehrs verwendet werden soll. |
| „CertificateLabel” auf Seite 146 | Die Zertifikatsbezeichnung, die das digitale Zertifikat angibt, das für den Schutz des Protokollreplikationsverkehrs verwendet werden soll. |
| „CipherSpec” auf Seite 146 | Die MQ-CipherSpec, die zum Schutz des Protokollreplikationsverkehrs verwendet werden soll. |
| „LocalAddress” auf Seite 146 | Die lokale Netzschnittstellenadresse, die den Protokollreplikationsverkehr akzeptiert. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| „HeartbeatInterval” auf Seite 147 | Das Heartbeatintervall legt fest, wie oft in Millisekunden eine aktive Instanz eines Warteschlangenmanagers mit Native HA ein Netzüberwachungssignal sendet. |
| „HeartbeatTimeout” auf Seite 147 | Das Überwachungssignalzeitlimit legt fest, wie lange eine Replikatinstanz eines Warteschlangenmanagers mit Native HA wartet, bevor sie entscheidet, dass die aktive Instanz nicht mehr reagiert. |
| „RetryInterval” auf Seite 147 | Das Wiederholungsintervall legt fest, wie oft in Millisekunden ein Warteschlangenmanager mit Native HA eine fehlgeschlagene Replikationsverbindung wiederholen soll. |
| Windows **Zeilengruppe 'NETBIOS'** | |
| Windows LocalName | Der Name, unter dem diese Maschine im LAN bekannt ist. |
| Windows AdapterNum | Die Nummer des LAN-Adapters. |
| Windows NumSess | Die Anzahl der Sitzungen, die zugeordnet werden sollen. |
| Windows NumCmds | Die Anzahl der Befehle, die zugeordnet werden sollen. |
| Windows NumNames | Die Anzahl der zuzuordnende Namen. |
| Windows Library1 | Der Name der NetBIOS-DLL. |
| **Zeilengruppe 'QMErrorLog'** | |
| ErrorLogGröße | Gibt die Größe des Fehlerprotokolls des Warteschlangenmanagers an, das in die Sicherung kopiert wird. |
| ExcludeMessage | Gibt Nachrichten an, die nicht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden sollen. |
| SuppressMessage | Gibt Nachrichten an, die nur einmal in einem angegebenen Zeitintervall in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden. |
| SuppressInterval | Gibt das Zeitintervall (in Sekunden) an, in dem Nachrichten, die in SuppressMessage angegeben sind, nur einmal in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden. |
| Linux AIX **Zeilengruppe 'RestrictedMode'** [2] | |
| Linux AIX ApplicationGroup | Der Name der lokalen Übertragungswarteschlange, in die ferne Nachrichten gestellt werden, wenn eine Übertragungswarteschlange nicht explizit für ihr Ziel definiert ist. |
| **Zeilengruppe für die Sicherheit** | |
| ClusterQueueAccessControl | Überprüft die Zugriffssteuerung von Clusterwarteschlangen oder vollständig qualifizierten Warteschlangen, die auf Clusterwarteschlangenmanagern gehostet sind. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| Windows GroupModel | Gibt an, ob der Objektberechtigungsmanager (Object Authority Manager, OAM) globale Gruppen prüft, wenn er die Gruppenzugehörigkeit eines Benutzers unter Windows bestimmt. |
| **Service-Zeilengruppe** | |
| Name | Der Name des erforderlichen Service. |
| EntryPoints | Die Anzahl der Eingangspunkte, die für den Service definiert wurden. |
| Windows SecurityPolicy | Unter Windows die Sicherheitsrichtlinie für jeden Warteschlangenmanager. |
| Linux AIX SecurityPolicy | Gibt unter AIX and Linux an, ob der Warteschlangenmanager die benutzerbasierte oder die gruppenbasierte Berechtigung verwendet.<br>V 9.3.0 Ab IBM MQ 9.3.0 können Sie auch einen nicht aktiven Systembenutzernamen erstellen. |
| SharedBindingsUserId | Für gemeinsam genutzte Bindungen wird angegeben, ob das Feld 'UserIdentifier' in der Struktur 'IdentityContext' aus der Funktion MQZ_AUTHENTICATE_USER die effektive Benutzer-ID oder die reale Benutzer-ID ist. |
| FastpathBindingsUserId | Für Fastpath-Bindungen wird angegeben, ob das Feld 'UserIdentifier' in der Struktur 'IdentityContext' aus der Funktion MQZ_AUTHENTICATE_USER die effektive Benutzer-ID oder die reale Benutzer-ID ist. |
| IsolatedBindingsUserId | Für isolierte Bindungen wird angegeben, ob das Feld 'UserIdentifier' in der Struktur 'IdentityContext' aus der Funktion MQZ_AUTHENTICATE_USER die effektive Benutzer-ID oder die reale Benutzer-ID ist. |
| **Zeilengruppe ' ServiceComponent** | |
| Service | Der Name des erforderlichen Service. |
| Name | Der beschreibende Name der Servicekomponente. |
| Modul | Der Name des Moduls, das den Code für diese Komponente enthält. |
| ComponentDataSize | Die Größe des Komponentendatenbereichs (in Byte), der an die Komponente in jedem Aufruf übergeben wurde. |
| Windows **Zeilengruppe 'SPX'** | |
| Windows Socket | Die SPX-Socket-Nummer in Hexadezimalschreibweise. |
| Windows BoardNum | Die LAN-Adapternummer. |
| Windows KeepAlive | Schalten Sie die KeepAlive-Funktion ein oder aus. |
| Windows Library1 | Der Name der SPX-DLL. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| Windows Library2 | Dasselbe gilt für LibraryName1, das verwendet wird, wenn der Code in zwei separaten Bibliotheken gespeichert wird. |
| Windows ListenerBacklog | Überschreiben Sie die Standardanzahl ausstehender Anforderungen für den SPX-Listener. |
| **SSL-Zeilengruppe** | |
| V 9.3.0 OutboundSNI | Es wird angegeben, ob SNI-fähige Clients beim Einleiten einer TLS-Verbindung als SNI für das ferne System den Namen des IBM MQ-Zielkanals oder den Hostnamen festlegen. |
| AllowOutboundSNI | Es wird angegeben, ob SNI-fähige Clients als SNI den Namen des IBM MQ-Zielkanals für das ferne System festlegen, wenn eine TLS-Verbindung eingeleitet wird.<br>**Achtung:** V 9.3.0 Deprecated Ab IBM MQ 9.3.0 ist diese Eigenschaft veraltet. Verwenden Sie stattdessen **OutboundSNI** . |
| AllowedCipherSpecs | Gibt eine benutzerdefinierte Liste der CipherSpecs an, die für die Verwendung mit IBM MQ-Kanälen auf Multiplatforms angeordnet und aktiviert sind. |
| AllowTLSV13 | Gibt an, ob ein Warteschlangenmanager die TLS 1.3 CipherSpecs verwenden kann. |
| CDPCheckExtensions | Gibt an, ob TLS-Kanäle in diesem Warteschlangenmanager versuchen, CDP-Server zu überprüfen, die in den Zertifikatserweiterungen des CrlDistributionPoint-Zertifikats benannt sind. |
| MinimumRSAKeyGröße | Gibt die Mindestschlüsselgröße an, die RSA-Zertifikate haben müssen, damit sie akzeptiert werden. |
| OCSPAuthentication | Gibt die Aktion an, die ausgeführt werden soll, wenn ein Widerrufsstatus nicht von einem OCSP-Server bestimmt werden kann. |
| OCSPCheckExtensions | Gibt an, ob TLS-Kanäle in diesem Warteschlangenmanager versuchen, OCSP-Server zu überprüfen, die in den Erweiterungen des Zertifikats 'AuthorityInfoAccess' angegeben sind. |
| OCSPZeitlimit | Die Anzahl der Sekunden, die bei der Ausführung einer Widerrufsprüfung auf einen OCSP-Responder gewartet wird. |
| ALW PeerCertChainValidation | Die Einstellung für die IBM Global Security Kit (GSKit)-Zertifikatsprüfung. |
| SSLHTTPProxyName | Der Hostname oder die Netzadresse des HTTP-Proxy-Servers, der von GSKit für OCSP-Prüfungen verwendet wird. |
| SSLHTTPConnectTimeout | Die Anzahl der Sekunde, die beim Ausführen einer Widerrufsprüfung auf die erfolgreiche Herstellung einer Netzverbindung zu einem HTTP-Server gewartet wird. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| **Subpoolzeilengruppe** „3" auf Seite 117 | Diese Zeilengruppe wird von IBM MQ erstellt. Ändern Sie sie nicht. |
| ShortSubpoolName „3" auf Seite 117 | Ein Name, der einem Verzeichnis und einem symbolischen Link entspricht, der im Verzeichnis /var/mqm/sockets erstellt wurde, das IBM MQ für die interne Kommunikation zwischen den aktiven Prozessen verwendet. |
| **TCP-Zeilengruppe** | |
| Port | Die Standardportnummer (in Dezimalschreibweise) für TCP/IP-Sitzungen. |
| Windows Library1 | Der Name der TCP/IP-Sockets-DLL. |
| KeepAlive | Schalten Sie die KeepAlive-Funktion ein oder aus. |
| ListenerBacklog | Überschreiben Sie die Standardanzahl ausstehender Anforderungen für den TCP/IP-Listener. |
| Connect_Timeout | Die Anzahl der Sekunden, bevor ein Versuch unternommen wird, das Socket-Zeitlimit zu verbinden. |
| SndBuffSize | Die Größe des TCP/IP-Sendepuffers in Byte, der vom sendenden Ende der Kanäle verwendet wird. |
| RcvBuffSize | Die Größe des TCP/IP-Empfangspuffers in Byte, der vom empfangenden Kanalende verwendet wird. |
| RcvSndBuffSize | Die Größe des TCP/IP-Sendepuffers in Byte, der vom senderseitigen Ende eines Empfängerkanals verwendet wird. |
| RcvRcvBuffSize | Die Größe des TCP/IP-Empfangspuffers in Byte, der vom empfangenden Ende eines Empfängerkanals verwendet wird. |
| SvrSndBuffSize | Die Größe des TCP/IP-Sendepuffers (in Byte), der vom Serverende eines Clientverbindungs-Serververbindungskanals verwendet wird. |
| SvrRcvBuffSize | Die Größe des TCP/IP-Empfangspuffers (in Byte), der vom Serverende eines Clientverbindungs-Serververbindungskanals verwendet wird. |
| Multi V 9.3.0 SecureCommsOnly | Gibt an, ob eine einfache Textkommunikation zulässig ist, der Standardwert ist oder nicht zulässig ist. |
| **Zeilengruppe 'TuningParameters'** | |
| SuppressDspAuthFail | Gibt an, ob der Warteschlangenmanager die Generierung von Berechtigungsereignissen und das Schreiben von AMQ8077-Fehlernachrichten in das Fehlerprotokoll unterdrückt, wenn eine Berechtigungsprüfung fehlschlägt, wenn die Verbindung nicht über die Berechtigung + dsp für ein Objekt verfügt. |
| ImplSyncOpenOutput | Die minimale Anzahl von Anwendungen, für die die Warteschlange geöffnet ist, bevor ein impliziter Synchronisationspunkt für eine persistente, außerhalb von Synchronisationspunktstellen aktiviert werden kann. |

| *Tabelle 11. Zeilengruppen der Datei 'qm.ini' (Forts.)* | |
|---|---|
| **Zeilengruppen und Attribute** | **Beschreibung der Attribute** |
| UniformClusterName | Der Name des IBM MQ-Clusters, den Sie als Uniform-Cluster verwenden. |
| OAMLdapConnectZeitlimit | Die maximale Zeit (in Sekunden), die der LDAP-Client wartet, um eine TCP-Verbindung zum Server herzustellen. |
| OAMLdapQueryTimeLimit | Die maximale Zeit (in Sekunden), in der der LDAP-Client auf eine Antwort auf eine LDAP-Anforderung vom Server wartet. |
| V 9.3.2 OAMLdapResponseWarning-Time | Wenn eine Verbindung zu einem LDAP-Server länger als die im Parameter **OAMLdapResponseWarningTime** angegebene Anzahl von Sekunden dauerte, wird eine AMQ5544W -Nachricht in das Fehlerprotokoll geschrieben. |
| ExpiryInterval | Gibt die Häufigkeit an, mit der der Warteschlangenmanager die Warteschlangen nach abgelaufenen Nachrichten durchsucht, die noch nicht durch andere Warteschlangenaktivitäten bereinigt wurden. Dies ist ein Zeitintervall in Sekunden. |
| **LivenessHeartBeatLen** | Konfiguriert die Häufigkeit, mit der der Warteschlangenmanager prüft, ob in das Protokoll geschrieben wird, mit einer angemessenen Geschwindigkeit. |
| **ECHeartBeatLen** | Konfiguriert die Häufigkeit der allgemeinen Statusprüfungen des Warteschlangenmanagers. |
| **FileLockHeartBeatLen** | Ändert den Standardwert für die Dateisperrenprüfungen für einen Warteschlangenmanager mit mehreren Instanzen, die der Ausführungscontroller regelmäßig ausführt, um sicherzustellen, dass er weiterhin die exklusive Sperre für die primäre Datei mit mehreren Instanzen hält. |
| **Zeilengruppe 'Variables'** | |
| *Attribut = Wert* | Ein Name und ein zugehöriger Wert, die bei MQSC-Definitionen eingefügt werden. |
| **Zeilengruppe 'XAResourceManager'** | |
| Name | Die Instanz des Ressourcenmanagers. |
| SwitchFile | Der vollständig qualifizierte Name der Ladedatei, die die XA-Switchstruktur des Ressourcenmanagers enthält. |
| XAOpenString | Die Zeichenfolge der Daten, die an den Eingangspunkt xa_open des Ressourcenmanagers übergeben werden sollen. |
| XACloseString | Die Zeichenfolge der Daten, die an den Eingangspunkt xa_close des Ressourcenmanagers übergeben werden sollen. |
| ThreadOfControl | Der Wert, den der Warteschlangenmanager für die Serialisierung verwendet, wenn er den Ressourcenmanager von einem seiner eigenen Multithread-Prozesse aus aufrufen muss. Für Windows ist dieser Wert verbindlich. |

**Anmerkungen:**

1. Die Zeilengruppe AccessMode wird durch die Option **-a [r]** im Befehl **crtmqm** definiert. Ändern Sie die Zeilengruppe 'AccessMode' nicht, nachdem der WS-Manager erstellt wurde.
2. Die Zeilengruppe 'RestrictedMode' wird durch die Option **-g** im Befehl **crtmqm** festgelegt. Ändern Sie diese Zeilengruppe nicht, nachdem der WS-Manager erstellt wurde. Wenn Sie die Option **-g** nicht verwenden, wird die Zeilengruppe nicht in der Datei `qm.ini` erstellt.
3. Die Zeilengruppe 'Subpool' und das Attribut 'ShortSubpoolName' in dieser Zeilengruppe werden automatisch von IBM MQ geschrieben, wenn Sie einen Warteschlangenmanager erstellen. IBM MQ wählt einen Wert für ShortSubpoolName aus. Ändern Sie diesen Wert nicht.

## Windows Zeilengruppe 'AccessMode' in der Datei 'qm.ini'

'AccessMode' gilt nur für Windows-Server. Die Zeilengruppe 'AccessMode' in der Datei 'qm.ini' wird durch die Option `-a [r]` im Befehl **crtmqm** festgelegt. Ändern Sie die Zeilengruppe 'AccessMode' nicht, nachdem der WS-Manager erstellt wurde.

Verwenden Sie die Zugriffsgruppe (`-a [r]`) des Befehls **crtmqm**, um eine Windows-Sicherheitsgruppe anzugeben, deren Mitglieder vollständigen Zugriff auf alle WS-Manager-Datendateien erhalten sollen. Die Gruppe kann abhängig von der verwendeten Syntax entweder eine lokale oder eine globale Gruppe sein. Folgende Syntax ist für den Gruppennamen gültig:

*LocalGroup*
*Domänenname* \ *Name der globalen Gruppe*
*Name der globalen Gruppe* @ *Domänenname*

Sie müssen die zusätzliche Zugriffsgruppe definieren, bevor Sie den Befehl **crtmqm** mit der Option `-a [r]` ausführen.

Wenn Sie die Gruppe mit `-ar` anstelle von `-a` angeben, wird der lokalen mqm-Gruppe kein Zugriff auf die Datendateien des Warteschlangenmanagers gewährt. Verwenden Sie diese Option, wenn das Dateisystem, in dem sich die WS-Manager-Datendateien befinden, keine Zugriffssteuerungseinträge für lokal definierte Gruppen unterstützt.

Die Gruppe ist üblicherweise eine globale Sicherheitsgruppe, die verwendet wird, um Multi-Instanz-Warteschlangenmanagern Zugriff auf einen gemeinsam genutzten Ordner für Warteschlangenmanagerdaten und Protokolle zu ermöglichen. Verwenden Sie die zusätzliche Sicherheitszugriffsgruppe, um Lese- und Schreibberechtigungen für den Ordner festzulegen oder darin enthaltene Warteschlangenmanagerdaten und Protokolldateien freizugeben.

Die zusätzliche Sicherheitszugriffsgruppe ist eine Alternative zur Verwendung der lokalen Gruppe `mqm`, um Berechtigungen für den Ordner festzulegen, der Warteschlangenmanagerdaten und -protokolle enthält. Im Gegensatz zur lokalen Gruppe `mqm` können Sie die zusätzliche Sicherheitszugriffsgruppe als lokale oder globale Gruppe definieren. Es muss eine globale Gruppe sein, um Berechtigungen für die freigegebenen Ordner festlegen zu können, die die Daten und Protokolldateien enthalten, die von Multi-Instanz-Warteschlangenmanagern verwendet werden.

Das Windows-Betriebssystem überprüft die Zugriffsberechtigungen zum Lesen und Schreiben von Warteschlangenmanager-Daten und -Protokolldateien. Es prüft die Berechtigungen der Benutzer-ID, die Warteschlangenmanager-Prozesse ausführt. Welche Benutzer-ID überprüft wird, hängt davon ab, ob Sie den Warteschlangenmanager als Service oder ob Sie ihn interaktiv gestartet haben. Wenn Sie den Warteschlangenmanager als Service gestartet haben, überprüft das Windows-Betriebssystem die Benutzer-ID, die Sie mit dem Assistenten zur **Vorbereitung von** IBM MQ konfiguriert haben. Wenn Sie den Warteschlangenmanager interaktiv gestartet haben, überprüft dasWindows-Betriebssystem die Benutzer-ID, die den Befehl **strmqm** ausgeführt hat.

Die Benutzer-ID muss Mitglied der lokalen `mqm`-Gruppe sein, um den Warteschlangenmanager zu starten. Wenn die Benutzer-ID zu der zusätzlichen Sicherheitszugriffsgruppe gehört, hat der Warteschlangenmanager Lese- und Schreibzugriff auf die Dateien, für die diese Berechtigungen durch Verwendung der Gruppe erteilt wurden.

**Einschränkung:** Eine zusätzliche Sicherheitszugriffsgruppe kann nur im Windows-Betriebssystem angegeben werden. Wenn Sie in anderen Betriebssystemen eine zusätzliche Sicherheitszugriffsgruppe angeben, gibt der Befehl **crtmqm** einen Fehler zurück.

## Beispielzeilengruppe

```
AccessMode:
SecurityGroup=wmq\wmq
```

**Zugehörige Konzepte**

„Nicht gemeinsam genutzte WS-Manager-Daten und -Protokollverzeichnisse und -Dateien unter Windows schützen" auf Seite 582

„Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern" auf Seite 579

**Zugehörige Tasks**

„WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen" auf Seite 551

**Zugehörige Verweise**

crtmqm (WS-Manager erstellen)

## Multi Zeilengruppe 'ApiExitLocal' in der Datei 'qm.ini'

Die lokale Zeilengruppe ApiExitgibt API-Exitroutinen für einen Warteschlangenmanager an.

Ändern Sie für einen Server die lokale Zeilengruppe ApiExitder Datei `qm.ini` , um API-Exitroutinen für einen Warteschlangenmanager anzugeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer `Exits` -Warteschlangenmanagers verwenden.

Ändern Sie für einen Client die lokale Zeilengruppe ApiExitin der Datei `mqclient.ini` , um API-Exitroutinen für einen Warteschlangenmanager anzugeben.

## Übersicht

Die Zeilengruppe `ApiExitLocal` ermöglicht es, nur eine einzige `Module` anzugeben, und es müssen jedoch vier Module bereitgestellt werden:

- 32-Bit-Thread ohne Thread
- 32-Bit-Thread
- 64-Bit-Thread ohne Thread
- 64-Bit-Thread

Beachten Sie, dass IBM MQ die Erweiterung `_r` an den angegebenen Modulnamen anhängt, um die Threadversion des Exits zu kennzeichnen. Für die 32-Bit-und 64-Bit-Varianten stellt IBM MQ jedoch keinen unmittelbar gleichwertigen Mechanismus bereit.

Die Versionen von `amqsaxe0` und `amqsaxe0_r` , die im Lieferumfang von *prefix*`/mqm/samp/bin` enthalten sind, werden für die native Größe des Warteschlangenmanagers auf der Plattform, für die sie erstellt wurden, erstellt (jetzt alle 64-Bit-Versionen) und können nur von Anwendungen verwendet werden, die in derselben nativen Größe ausgeführt werden.

Wenn ein nicht qualifizierter Modulname bereitgestellt wird, sucht IBM MQ in `/var/mqm/exits` für die 32-Bit-Varianten und in `/var/mqm/exits64` für die 64-Bit-Varianten.

`module=amqsaxe` impliziert z. B.:

```
/var/mqm/exits/amqsaxe  - 32 bit unthreaded variant
/var/mqm/exits/amqsaxe_r - 32 bit threaded variant
```

```
/var/mqm/exits64/amqsaxe - 64 bit unthreaded variant
/var/mqm/exits64/amqsaxe_r - 64 bit threaded variant
```

Windows Auf Windows-Systemen können Sie auch den Befehl **amqmdain** verwenden, um die Einträge für API-Exits zu ändern. (Zur Angabe von API-Exit-Routinen für alle Warteschlangenmanager verwenden Sie die Zeilengruppen 'ApiExitCommon' und 'ApiExitTemplate', wie unter „Zeilengruppen ApiExitCommon und ApiExitTemplate in der Datei mqs.ini" auf Seite 98 beschrieben.)

Beachten Sie, dass die Nachricht vom Server nicht umgesetzt werden muss, damit der API-Exit ordnungsgemäß funktioniert. Nachdem der API-Exit die Nachricht verarbeitet hat, muss die Nachricht dann auf dem Client konvertiert werden. Daher ist es erforderlich, dass Sie alle Konvertierungsexits auf dem Client installiert haben.

Weitere Informationen zur Verwendung dieser Attribute finden Sie unter API-Exits konfigurieren.

## Parameter

**Name=Name_des_API-Exits**
Der beschreibende Name des API-Exits, der im Feld 'ExitInfoName' der MQAXP-Struktur übergeben wird.

Dieser Name muss eindeutig sein und darf maximal 48 Zeichen umfassen, wobei es sich um gültige Zeichen für die Namen von IBM MQ-Objekten (z. B. Warteschlangennamen) handeln muss.

**Function=Funktionsname**
Der Name des Einstiegspunkts der Funktion in dem Modul, das den API-Exitcode enthält. Dieser Einstiegspunkt ist die Funktion MQ_INIT_EXIT.

Die Länge dieses Felds ist auf den Wert von MQ_EXIT_NAME_LENGTH beschränkt.

**Module=Modulname**
Das Modul, das den API-Exitcode enthält.

Wenn dieses Feld den vollständigen Pfadnamen enthält, wird es unverändert übernommen. Wenn dieses Feld lediglich den Modulnamen enthält, wird das Modul über das Attribut **ExitsDefaultPath** in der Zeilengruppe `ExitPath` der Datei `qm.ini` lokalisiert.

Auf Plattformen, die separate Threadbibliotheken unterstützen, muss sowohl eine Threadversion als auch eine Version des API-Exitmoduls ohne Threads bereitgestellt werden. Die Threadversion muss das Suffix `_r` aufweisen. Von der Threadversion des IBM MQ-Anwendungsstubs wird dem angegebenen Modulnamen vor dem Laden implizit `_r` angehängt.

Die Länge dieses Felds ist auf die maximale Pfadlänge begrenzt, die von der Plattform unterstützt wird.

**Data=Datenname**
Die Daten, die an den API-Exit im Feld 'ExitData' der MQAXP-Struktur übergeben werden sollen.

Wenn Sie dieses Attribut angeben, werden die führenden und abschließenden Leerzeichen entfernt. Die verbleibende Zeichenfolge wird auf 32 Zeichen abgeschnitten und das Ergebnis wird an den Exit übergeben. Wenn Sie dieses Attribut ausschließen, wird der Standardwert (32 Leerzeichen) an den Exit übergeben.

Die maximale Länge dieses Felds beträgt 32 Zeichen.

**Sequence=Folgenummer**
Die Reihenfolge, in der dieser API-Exit in Relation zu anderen API-Exits aufgerufen wird. Ein Exit mit einer niedrigen Folgenummer wird vor einem Exit mit einer höheren Folgenummer aufgerufen. Die Folgenummern für die Exits müssen nicht fortlaufend vergeben werden. Die Folge '1, 2, 3' führt zu demselben Ergebnis wie die Folge '7, 42, 1096'. Wenn zwei Exits dieselbe Folgenummer aufweisen, entscheidet der Warteschlangenmanager, welcher Exit zuerst aufgerufen wird. Nach dem Ereignis können Sie feststellen, welcher Exit aufgerufen wurde, indem Sie die Uhrzeit oder einen Datenpunkt in 'ExitChainArea' (durch 'ExitChainAreaPtr' in MQAXP angegeben) eingeben oder eine eigene Protokolldatei schreiben.

Dieses Attribut ist ein numerischer Wert ohne Vorzeichen.

### Beispielzeilengruppe

```
ApiExitLocal:
   Name=ClientApplicationAPIchecker
   Sequence=3
   Function=EntryPoint
   Module=/usr/Dev/ClientAppChecker
   Data=9.20.176.20
```

## V 9.3.4 Linux AIX Zeilengruppe AuthToken in der Datei qm.ini

Mithilfe der Zeilengruppe **AuthToken** können Sie den Warteschlangenmanager für die Validierung von Authentifizierungstoken konfigurieren, die von verbundenen Anwendungen bereitgestellt werden.

### Zeilengruppe AuthToken

**KeyStore= Zeichenfolge**
Dateipfad für den Keystore, der die öffentlichen Schlüsselzertifikate und symmetrischen Schlüssel des vertrauenswürdigen Ausstellers enthält. Sie können die Schlüssel einem vorhandenen Keystore hinzufügen oder einen neuen Keystore erstellen. Weitere Informationen finden Sie unter Warteschlangenmanager für das Akzeptieren von Authentifizierungstoken konfigurieren. Der Warteschlangenmanager verwendet die Schlüssel im Keystore, um zu überprüfen, ob das Authentifizierungstoken, das die Anwendung darstellt, vom vertrauenswürdigen Aussteller signiert wurde.

Sie können entweder einen CMS -Keystore mit der Dateierweiterung `.kdb` oder einen PKCS#12 -Keystore mit der Dateierweiterung `.p12`verwenden. Wenn die Keystore-Datei nicht vorhanden ist oder kein Zugriff darauf möglich ist, wird der Fehler `AMQ7076E: Invalid value for attribute in ini file` im Fehlerprotokoll des Warteschlangenmanagers ausgegeben.

Stellen Sie sicher, dass der Keystore-Typ der Dateinamenerweiterung für den Keystore entspricht. IBM MQ erkennt das korrekte Format des Keystores. Inkonsistenzen können jedoch zu anderen administrativen Problemen führen, wenn der Keystoretyp und die Dateinamenerweiterung nicht übereinstimmen.

Die maximale Länge des Schlüsselspeicherdateipfads beträgt 256 Zeichen.

**KeyStorePwdFile= Zeichenfolge**
Dateipfad für die Datei, die das verschlüsselte Kennwort für den Schlüsselspeicher enthält Die Datei muss das verschlüsselte Kennwort als einzelne Textzeile enthalten. Klartextkennwörter werden nicht akzeptiert.

Verschlüsseln Sie das Kennwort mit dem Befehl **runqmcred** , bevor Sie es in der Schlüsselspeicherkennwortdatei speichern. Die Schlüsselspeicherkennwortdatei darf nur das verschlüsselte Kennwort enthalten, das mit dem Befehl **runqmcred** erstellt wurde.

Die maximale Länge des Klartextkennworts vor der Verschlüsselung beträgt 1024 Zeichen.

Dieser Parameter ist optional. Wird sie nicht angegeben, sucht der WS-Manager nach einer Stashdatei mit dem Kennwort in demselben Verzeichnis und mit demselben Namen wie der Keystore, aber mit der Dateierweiterung `.sth`. Wird die Stashdatei nicht gefunden, wird die Konfiguration zurückgewiesen und die Fehlernachricht AMQ7006E wird im Fehlerprotokoll des Warteschlangenmanagers ausgegeben. Weitere Informationen zu den Optionen zum Speichern von Schlüsselspeicherkennwörtern finden Sie unter Schlüsselrepository-Kennwörter verschlüsseln.

Die maximale Länge des Kennwortdateipfads beträgt 256 Zeichen.

**CertLabel= Zeichenfolge**
Die Zertifikatsbezeichnung für ein Zertifikat mit öffentlichem Schlüssel oder einen symmetrischen Schlüssel im Keystore, der für die Validierung von Authentifizierungstoken verwendet wird. Sie können bis zu 32 Zertifikatsbezeichnungen angeben, indem Sie das Attribut **CertLabel** wiederholen.

Wenn Sie dem Warteschlangenmanager-Keystore Zertifikate hinzufügen, geben Sie ihnen aussagekräftige Bezeichnungen. Bei Zertifikatsbezeichnungen muss die Groß-/Kleinschreibung beachtet werden. Sie können alphanumerische Zeichen, Interpunktionszeichen und Leerzeichen enthalten. Wenn

ein ungültiges Zeichen erkannt wird, wird ein Fehler zurückgegeben und eine Fehlernachricht in das IBM MQ -Fehlerprotokoll geschrieben.

Anerkannte Tokenaussteller können mehrere öffentliche Schlüsselzertifikate und symmetrische Schlüssel bereitstellen. Zertifikate mit öffentlichen Schlüsseln haben beispielsweise Gültigkeitszeiträume. Wenn sie kurz vor Ablauf stehen, stellt der Tokenaussteller ein neues Zertifikat mit einem neuen Ablaufdatum bereit. Für eine Zeit können beide Zertifikate gültig sein.

Wenn Anwendungen Token für die Authentifizierung darstellen, wird die Liste der **CertLabels** überprüft, bis ein gültiger Schlüssel gefunden wird, der zum Signieren des Tokens verwendet wurde. Wenn die Übereinstimmung gefunden wird, wird die Tokensignatur validiert.

Wenn **CertLabel** nicht angegeben wird, schlägt die Verbindung von der Anwendung, die das Token darstellt, mit dem Ursachencode `2063 MQRC_SECURITY_ERROR`fehl und die Nachricht `AMQ5786E: Authentication token configuration error` wird in das Fehlerprotokoll des Warteschlangenmanagers geschrieben.

Die maximale Länge der Zertifikatsbezeichnung beträgt 64 Zeichen.

Beispiel:

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
   CertLabel=token
   CertLabel=rsakey
   CertLabel=mark
   ... up to 32 CertLabel fields
```

**UserClaim= Zeichenfolge**

Anforderung innerhalb des Tokens, das die Benutzer-ID enthält, die der Warteschlangenmanager für Berechtigungsprüfungen übernimmt

Dieser Parameter ist optional, wenn der Warteschlangenmanager mit **ADOPTCTX(NO)** konfiguriert ist. Bei Verwendung von **ADOPTCTX(YES)** ist dieser Parameter erforderlich. **ADOPTCTX** ist ein Attribut im Authentifizierungsinformationsobjekt (AUTHINFO), das vom Attribut **CONNAUTH** des Warteschlangenmanagers referenziert wird.

Um eine Identität anzunehmen, muss das Token einen Anspruch mit dem Namen enthalten, der im Attribut **UserClaim** der Zeilengruppe **AuthToken** angegeben ist, und es muss **ADOPTCTX(YES)** verwendet werden.

Wenn Ihr Token beispielsweise einen Anspruch `"AppUser": "MyUserName"`enthält, müssen Sie `UserClaim=AppUser` in der Zeilengruppe AuthToken der Datei `qm.ini` angeben, um die Identität `"MyUserName"` für die Berechtigung anzunehmen.

Die maximale Länge des Attributwerts **UserClaim** beträgt 128 Zeichen.

**Anmerkung:** Ab IBM MQ 9.3.4wird bei Angabe der Zeilengruppe AuthToken der effektive Wert des Attributs **SecurityPolicy** der Zeilengruppe 'Service' auf `UserExternal`gesetzt. Die Tokenauthentifizierung ist nicht verfügbar, wenn **SecurityPolicy** explizit auf `Group` in der Zeilengruppe 'Service' gesetzt ist. Wenn **SecurityPolicy** auf `Group`gesetzt ist, entfernen Sie das Attribut **SecurityPolicy** aus der Zeilengruppe 'Service' und starten Sie den Warteschlangenmanager erneut. Weitere Informationen finden Sie unter SecurityPolicy.

**Anmerkung:** Mit dem Attribut **ADOPTCTX** des Authentifizierungsinformationsobjekts können Sie steuern, ob die Benutzer-ID im Token für Berechtigungsprüfungen übernommen wird. Wenn Sie den Warteschlangenmanager erstellen, wird dieses Attribut auf **ADOPTCTX(YES)** gesetzt. Dieser Wert bewirkt, dass die Benutzer-ID aus dem Token übernommen wird. Die Benutzer-ID muss die Anforderungen für Benutzer-IDs in Authentifizierungstoken erfüllen. Weitere Informationen finden Sie unter Benutzer-IDs in Authentifizierungstoken. Wenn die Tokenbenutzeranforderung eine Benutzer-ID enthält, die die Anforderungen nicht erfüllt, wird die Verbindung mit dem Ursachencode **2035 MQRC_NOT_AUTHORIZED**zurückgewiesen. Wenn **ADOPTCTX(NO)** festgelegt ist, wird das Token nur für die Authentifizierung verwendet und ein anderer Benutzer muss für die Autorisierung verwendet werden.

**AllowOSGroups=NO (Standardwert) |YES**
Der Standardwert ist NO. Bestimmt, ob eine Identität, die aus einem Token übernommen wird, als Betriebssystembenutzer behandelt wird und ob die Gruppenzugehörigkeiten des übereinstimmenden Betriebssystembenutzers während der Autorisierung berücksichtigt werden.

**AllowOSGroups= NO | N**
Berechtigungsprüfungen basieren nur auf dem Namen des Benutzers, der aus dem Token übernommen wird.

**AllowOSGroups= JA | J**
Berechtigungsprüfungen basieren auf dem Namen des Benutzers und die Gruppen, zu denen sie möglicherweise gehören, werden ebenfalls überprüft.

## Beispielzeilengruppe-nur Authentifizierung

Ihre Zeilengruppe **AuthToken** kann nur mit den beiden mindestens erforderlichen Parametern gültig sein:

- **KeyStore** -Dateipfad und
- **CertLabel** -Name.

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb
   CertLabel=token
   ... up to 32 CertLabel fields
```

Wenn Sie nur die beiden Mindestparameter angegeben haben, gilt Folgendes:

- Eine Stashdatei `key.sth` muss mit dem verschlüsselten Keystore-Kennwort vorhanden sein, damit die Keystore-Kennwortdatei nicht erforderlich ist.
- Das Token enthält keinen Benutzernamen, der zur Autorisierung an IBM MQ übergeben werden muss. Die Anwendung kann eine Verbindung herstellen und authentifiziert werden, aber es muss ein anderer Mechanismus vorhanden sein, um der Anwendung die Berechtigung zu erteilen, Arbeit auszuführen, nachdem sie verbunden wurde.

Je nach Konfiguration für Ihren Warteschlangenmanager kann der Benutzername, der für die Berechtigung verwendet wird, der Benutzername sein, der im Kanal über MCA-Regeln definiert ist, oder der Benutzername, unter dem die Client-App ausgeführt wurde, kann auf Ihrem Server vorhanden sein und zu Gruppen mit Berechtigungen gehören. Beachten Sie Folgendes, wenn Sie Tokens verwenden:

- Ihr Warteschlangenmanager wird in den Modus **UserExternal** versetzt, d. h., Benutzer, die auf dem Betriebssystem, auf dem der Warteschlangenmanager ausgeführt wird, nicht vorhanden sind, können für die Authentifizierung verwendet werden.
- Auch wenn Sie die Option **AllowOSGroups** nicht in Ihre Zeilengruppe **AuthToken** `qm.ini` einschließen, wird der Standardwert auf Nogesetzt. Wenn Sie **UserClaim** einschließen, aber **AllowOSGroups**=Janicht angeben, wird der Tokenbenutzer, der für die Autorisierung übernommen wird, nicht auf Gruppen überprüft, zu denen er möglicherweise auf dem Betriebssystem gehört, auf dem der Warteschlangenmanager ausgeführt wird.

## Beispielzeilengruppe-Authentifizierung und Berechtigung

Sie können alle **AuthToken** -Parameter definieren:

- **KeyStore** -Dateipfad,
- **KeyStorePwdFile** -Dateipfad,
- **CertLabel** -Name,
- **UserClaim** -Name und
- **AllowOSGroups**-Option gelöscht werden.

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qmgrs/QMJWT/ssl/key.kdb
```

```
    KeyStorePwdFile=/var/mqm/qmgrs/QMJWT/ssl/key.pw
    CertLabel=token
    CertLabel=rsakey
    CertLabel=mark
    ... up to 32 CertLabel fields
    UserClaim=AppUser
    AllowOSGroups=Y
```

Wenn Sie alle verfügbaren Parameter eingeschlossen haben, gilt Folgendes:

- Verschlüsseln Sie das Kennwort für den Keystore mit dem Befehl **runqmcred** . Speichern Sie sie in einer Datei und schließen Sie dann den Dateipfad in die Zeilengruppe **AuthToken** ein.
- Der Benutzername im Benutzeranspruch des Authentifizierungstokens wird sowohl für die Authentifizierung als auch für die Berechtigung verwendet.
  - Der Tokenbenutzer kann als Benutzer auf dem Betriebssystem vorhanden sein, auf dem der Warteschlangenmanager ausgeführt wird.
  - Sie haben ein Authentifizierungsinformationsobjekt definiert, um die Benutzerprüfung zu aktivieren.
  - Sie richten Kanalauthentifizierungsdatensätze ein, um einen Benutzer mit Berechtigung zur Interaktion mit IBM MQ -Objekten basierend auf den Kanalauthentifizierungs-oder MCA-Regeln zu übernehmen.

Ihre Strategie für die Authentifizierung und Autorisierung von Tokenbenutzern hängt von Ihren Anforderungen und von der Konfiguration Ihrer IBM MQ -Warteschlangenmanager ab. Weitere Informationen finden Sie unter Mit Authentifizierungstoken arbeiten.

**Zugehörige Konzepte**

Mit Tokens arbeiten

**Zugehörige Tasks**

Warteschlangenmanager für die Annahme von **AuthTokens** konfigurieren

Authentifizierungstoken in einer Anwendung verwenden

## Multi Zeilengruppe 'AutoCluster' in der Datei qm.ini

Die Zeilengruppe AutoCluster wird verwendet, wenn der WS-Manager zu ermitteln beginnt, ob der Cluster Mitglied eines automatischen Clusters ist, und kann die vollständigen Repositorys des Clusters identifizieren.

Die folgenden Attribute sind für die Zeilengruppe 'AutoCluster' obligatorisch:

**Type=*Uniform***
Gibt den Typ des automatischen Clusters an. Die einzige gültige Option ist *Uniform*, was für einen Uniform-Cluster steht.

**ClusterName=*<String>***
Der Name des Clusters, d. h. der Name des automatischen Clusters.

Die folgenden Attribute sind für die Zeilengruppe 'AutoCluster' optional, müssen aber paarweise angegeben werden:

**RepositoryName1 =*< Zeichenfolge>***
Dies ist der Warteschlangenmanagername für das erste vollständige Repository im automatischen Cluster. Es kann der Name dieses Warteschlangenmanagers oder ein anderer Name sein.

**Repository1Conname=*< Verbindungsnamenszeichenfolge>***
Dies ist der Verbindungsname (CONNAME). Dieser Wert gibt an, wie die Verbindung zwischen den Mitgliedern des automatischen Clusters und diesem Warteschlangenmanager hergestellt wird.

**Repository2Name=*< Zeichenfolge>***
Dies ist der Warteschlangenmanagername für das zweite vollständige Repository im automatischen Cluster. Es kann der Name dieses Warteschlangenmanagers oder ein anderer Name sein.

**Repository2Conname=*< Verbindungsnamenszeichenfolge>***
Dies ist der Verbindungsname (CONNAME). Dieser Wert gibt an, wie die Verbindung zwischen den Mitgliedern des automatischen Clusters und diesem Warteschlangenmanager hergestellt wird.

### Beispielzeilengruppe

```
AutoCluster:
        Repository1Name=QM1
        Repository2Name=QM2
        Repository1Conname=127.0.0.1(1414)
        Repository2Conname=127.0.0.1(1415)
        ClusterName=UNIFORMCLUSTER1
        Type=Uniform
```

**Zugehörige Konzepte**

„Automatischer Ausgleich von Anwendungen” auf Seite 444
Die Verteilung und Verfügbarkeit von Anwendungen durch den automatischen Anwendungsausgleich wird erheblich verbessert, indem ein IBM MQ-Uniform-Cluster aktiviert wird, mit dem die Anwendungsverteilung im gesamten Cluster genau verwaltet wird, anstatt auf eine beliebige Festlegung oder manuelle Fixierung von Anwendungen auf bestimmte Warteschlangenmanager angewiesen zu sein.

**Zugehörige Tasks**

„Neuen Uniform-Cluster erstellen” auf Seite 459
Hier finden Sie Informationen, wie ein neuer Uniform-Cluster erstellt wird.

**Zugehörige Verweise**

„Automatische Clusterkonfiguration verwenden” auf Seite 463
Sie konfigurieren IBM MQ, um die automatische Konfiguration zu aktivieren, indem Sie die `qm.ini`-Konfigurationsinformationen ändern.

## Multi Zeilengruppe 'AutoConfig' in der Datei qm.ini

Die Attribute der Zeilengruppe 'AutoConfig' werden häufig im Rahmen der Einrichtung von Uniform-Clustern verwendet.

**Anmerkung:** Die Zeilengruppe 'AutoCluster' kann nur für Uniform-Cluster verwendet werden.

**MQSCConfig=*<Path>***
Der Pfad ist entweder ein vollständiger Dateipfad oder Pfad zu einem Verzeichnis, in dem alle `*.mqsc`-Dateien bei jedem Warteschlangenmanager-Start auf den Warteschlangenmanager angewendet werden.

Weitere Informationen finden Sie im Abschnitt Automatische Konfiguration aus einem MQSC-Script beim Start.

**IniConfig=*<Path>***
Der Pfad ist entweder ein vollständiger Dateipfad oder Pfad zu einem Verzeichnis, in dem alle `*.ini`-Dateien auf jedem Warteschlangenmanager-Start auf die `qm.ini`-Datei angewendet werden.

Weitere Informationen finden Sie unter „Automatische Konfiguration von 'qm.ini' beim Start” auf Seite 105.

V 9.3.0 **ConfigTimeout**
Der Wert gibt die Zeit (in Sekunden) an, die der Warteschlangenmanager auf den Abschluss der automatischen Konfiguration wartet. Nach dieser Zeit wird der Warteschlangenmanager gestartet und steht Anwendungen für Verbindungen zur Verfügung.

Das Standardverhalten ist, dass es kein Zeitlimit gibt. Dies bedeutet, dass der Warteschlangenmanager Anwendungen erst für Verbindungen zur Verfügung steht, wenn alle Befehle für die automatische Konfiguration ausgeführt wurden.

Dieses Attribut sollte nicht konfiguriert werden, einfach weil die Konfiguration eine lange Zeit benötigt, da Anwendungen möglicherweise eine Verbindung herstellen können, bevor die für sie gültige Konfiguration abgeschlossen wurde, z. B. die Erstellung von Warteschlangen, die von der Anwendung benötigt werden.

### Beispielzeilengruppe

```
AutoConfig:
        MQSCConfig=/tmp/auto.mqsc
        IniConfig=/tmp/auto.ini
        ConfigTimeout=120
```

**Zugehörige Konzepte**

„Automatischer Ausgleich von Anwendungen” auf Seite 444
Die Verteilung und Verfügbarkeit von Anwendungen durch den automatischen Anwendungsausgleich wird erheblich verbessert, indem ein IBM MQ-Uniform-Cluster aktiviert wird, mit dem die Anwendungsverteilung im gesamten Cluster genau verwaltet wird, anstatt auf eine beliebige Festlegung oder manuelle Fixierung von Anwendungen auf bestimmte Warteschlangenmanager angewiesen zu sein.

**Zugehörige Tasks**

„Neuen Uniform-Cluster erstellen” auf Seite 459
Hier finden Sie Informationen, wie ein neuer Uniform-Cluster erstellt wird.

**Zugehörige Verweise**

„Automatische Clusterkonfiguration verwenden” auf Seite 463
Sie konfigurieren IBM MQ, um die automatische Konfiguration zu aktivieren, indem Sie die `qm.ini`-Konfigurationsinformationen ändern.

## Multi Zeilengruppe 'Channels' in der Datei 'qm.ini'

Die Attribute der Zeilengruppe 'Channels' bestimmen die Konfiguration eines Kanals.

z/OS Diese Informationen gelten nicht für IBM MQ for z/OS.

Mithilfe der Zeilengruppe `CHANNELS` in der Datei `qm.ini` können Sie Informationen zu Kanälen angeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer `Channels` -Warteschlangenmanagers verwenden.

**MaxChannels = 100 (Standardwert) |*number***
Die maximale Anzahl zulässiger *aktueller* Kanäle.

Der Standardwert ist 100.

Gegebenenfalls können Sie für **`MaxChannels`** einen anderen Wert festlegen, um die maximale Anzahl der aktuellen Kanäle zu begrenzen. Bei IBM MQ Appliance beträgt der Standardwert 999 999 999 und sollte nicht geändert werden.

**MaxActiveChannels= *MaxChannels_value***
Die maximale Anzahl der Kanäle, die zu einem beliebigen Zeitpunkt *aktiv* sein dürfen. Der Standardwert ist der Wert, der für das Attribut **`MaxChannels`** angegeben ist.

**MaxInitiators = 3 (Standardwert) |*number***
Die maximale Anzahl der Initiatoren. Der Standardwert und der Maximalwert sind 3.

**MQIBindType = FASTPATH | STANDARD**
Die Bindung für Anwendungen:

**FASTPATH**
Kanäle werden unter Verwendung von `MQCONNX  FASTPATH` verbunden; es gibt keinen Agentenprozess.

**STANDARD**
Kanäle verbinden sich mit STANDARD.

**PipeLineLength = 1 | *number***
Die maximale Anzahl gleichzeitiger Threads, die ein Kanal verwenden wird. Der Standardwert ist 1. Alle Werte größer als 1 werden wie der Wert 2 behandelt.

Wenn Sie Pipelining verwenden, konfigurieren Sie die Warteschlangenmanager an beiden Enden des Kanals, um einen Wert für **PipeLineLength** größer als 1 zu verwenden.

**Anmerkung:** Pipelining ist nur für TCP/IP-Kanäle wirksam.

Weitere Informationen finden Sie unter Unterstützung mehrerer Threads-Pipelining .

**AdoptNewMCA = NO (Standardwert) | SVR | SDR | RCVR | CLUSRCVR | ALL | FASTPATH**
Wenn IBM MQ eine Anforderung zum Starten eines Kanals empfängt, aber feststellt, dass bereits eine Instanz des Kanals aktiv ist, muss in einigen Fällen die vorhandene Kanalinstanz gestoppt werden, bevor die neue Instanz gestartet werden kann. Mit dem Attribut **AdoptNewMCA** können Sie steuern, welche Arten von Kanälen auf diese Weise beendet werden können.

Wenn Sie das Attribut **AdoptNewMCA** für einen bestimmten Kanaltyp angeben, der neue Kanal jedoch nicht gestartet werden kann, weil bereits eine übereinstimmende Kanalinstanz ausgeführt wird, gilt Folgendes:

1. Der neue Kanal versucht, die vorherige zu stoppen, indem er sie zum Beenden anfordert.
2. Wenn der vorherige Kanalserver auf diese Anforderung nicht reagiert, wenn das Warteintervall von **AdoptNewMCATimeout** abläuft, wird der Thread oder der Prozess für den vorherigen Kanalserver beendet.
3. Wenn der vorherige Kanalserver nach Schritt 2 nicht beendet wurde und das Warteintervall von **AdoptNewMCATimeout** zum zweiten Mal abläuft, beendet IBM MQ den Kanal mit einem CHANNEL IN USE-Fehler.

Die Funktion **AdoptNewMCA** gilt für Server-, Sender-, Empfänger-und Clusterempfängerkanäle. Im Fall eines Sende-oder Serverkanals kann im empfangenden Warteschlangenmanager nur eine Instanz eines Kanals mit einem bestimmten Namen ausgeführt werden. Im Fall eines Empfangs-oder Clusterempfängerkanals werden möglicherweise mehrere Instanzen eines Kanals mit einem bestimmten Namen im empfangenden Warteschlangenmanager ausgeführt, aber es kann jeweils nur eine Instanz von einem bestimmten fernen Warteschlangenmanager ausgeführt werden.

**Anmerkung: AdoptNewMCA** wird auf Requester- oder Serververbindungskanälen nicht unterstützt.

Geben Sie einen oder mehrere durch Kommas oder Leerzeichen voneinander getrennte Werte aus der folgenden Liste an:

**NEIN**
Das Feature AdoptNewMCA ist nicht erforderlich. Dies ist die Standardeinstellung.

**SVR**
Verwalten Sie Serverkanäle.

**SDR**
Adoptions-Senderkanäle.

**RCVR**
Empfaenger-Channels.

**CLUSRCVR**
Clusterempfängerkanäle verwalten.

**ALLE**
Alle Kanaltypen mit Ausnahme von FASTPATH-Kanälen werden hinzugefügt.

**FASTPATH**
Geben Sie den Kanal an, wenn es sich um einen FASTPATH-Kanal handelt. Dies geschieht nur, wenn der entsprechende Kanaltyp ebenfalls angegeben wird, z. B. AdoptNewMCA=RCVR,SVR,FASTPATH.

**Achtung!:** Das Attribut "AdoptNewMCA" verhält sich möglicherweise in unvorhersehbarer Weise mit FASTPATH-Kanälen. Gehen Sie mit großer Vorsicht vor, wenn Sie das Attribut AdoptNewMCA für FASTPATH-Kanäle aktivieren.

**AdoptNewMCATimeout= 60 (Standardwert) | 1-3600**
Die Zeit in Sekunden, die die neue Kanalinstanz wartet, bis die alte Kanalinstanz beendet wird. Geben Sie einen Wert im Bereich von 1 bis 3600 an. Der Standardwert ist 60.

**AdoptNewMCACheck = QM | ADRESSE | NAME | ALL**
Der Typ der Überprüfung, die erforderlich ist, wenn das Attribut AdoptNewMCA aktiviert wird. Wenn möglich, führen Sie eine vollständige Überprüfung durch, um die Kanäle vor dem Herunterfahren, versehentlich oder böswillig zu schützen. Überprüfen Sie mindestens, ob die Kanalnamen übereinstimmen.

Geben Sie einen oder mehrere der folgenden Werte an, die im Fall von *QM*, *NAME* oder *ALL* durch Kommas oder Leerzeichen getrennt sind:

**QM**
Überprüfen Sie, ob die Namen der WS-Manager übereinstimmen.

Beachten Sie, dass der Name des Warteschlangenmanagers selbst und nicht die QMID übereinstimmt.

**ADDRESS**
Überprüfen Sie die IP-Adresse der DFV-Quelle. Zum Beispiel die TCP/IP-Adresse.

**Anmerkung:** Durch Komma getrennte CONNAME-Werte gelten für Zieladressen und sind daher für diese Option nicht relevant.

Wenn ein Warteschlangenmanager mit mehreren Instanzen von `hosta` zu `hostb` nicht ausgeführt werden kann, verwenden alle abgehenden Kanäle dieses Warteschlangenmanagers die Quellen-IP-Adresse von `hostb`. Wenn sich dieser Wert nicht von `hosta` unterscheidet, kann AdoptNewMCACheck = *ADDRESS* nicht übereinstimmen.

Sie können SSL oder TLS mit gegenseitiger Authentifizierung verwenden, um zu verhindern, dass ein Angreifer einen vorhandenen aktiven Kanal aufbricht. Alternativ können Sie eine HACMP-Typlösung mit IP-Übernahme anstelle von Warteschlangenmanagern mit mehreren Instanzen verwenden oder eine Netzlastausgleichsfunktion verwenden, um die Quellen-IP-Adresse zu maskieren.

**NAME**
Überprüfen Sie, ob die Kanalnamen übereinstimmen.

**ALLE**
Suchen Sie nach übereinstimmenden WS-Managernamen, der DFV-Adresse und nach übereinstimmenden Kanalnamen.

Der Standardwert ist `AdoptNewMCACheck=NAME,ADDRESS,QM`.

**ChlauthEarlyAdopt = Y (Standardwert) | N**
Die Reihenfolge, in der die Authentifizierungsregeln für die Verbindung und die Regeln für die Kanalauthentifizierung verarbeitet werden, ist ein wichtiger Faktor bei der Bestimmung des Sicherheitskontexts für IBM MQ-Clientanwendungsverbindungen.

**Achtung:** Wenn dass Attribut **`ChlauthEarlyAdopt`** nicht in der Datei 'qm.ini' vorhanden ist, ist `N` der Standardwert. Allerdings werden ab IBM MQ 9.0.4 alle Warteschlangenmanager, die mit **`ChlauthEarlyAdopt`**=Y erstellt wurden, automatisch der Datei 'qm.ini' hinzugefügt.

**`ChlauthEarlyAdopt`** übernimmt nur Benutzer-IDs, die einem Warteschlangenmanager zur Verbindungsauthentifizierung bereitgestellt wurden, wenn ADOPTCTX(YES) im Objekt AUTHINFO für die Verbindungsauthentifizierung im Warteschlangenmanager festgelegt ist.

Gültige Werte für **`ChlauthEarlyAdopt`** sind die folgenden Werte:

**Y**
Der Kanal validiert und übernimmt die Berechtigungsnachweise für die Benutzer-ID und das Kennwort, die von einer Anwendung bereitgestellt wurden, die die Authentifizierung über die Warteschlangenmanagerverbindung verwendet, bevor Kanalauthentifizierungsregeln angewendet werden. In diesem Betriebsmodus stimmen die Kanalauthentifizierungsregeln mit der Benutzer-ID überein, die sich aus den Verbindungsauthentifizierungsprüfungen ergibt.

**N**
Der Kanal verzögert die Authentifizierung der Benutzer-ID und des Kennworts für die Verbindungsauthentifizierung, die von einer Anwendung bereitgestellt wurden, bis die Kanalauthentifizierungsregeln angewendet wurden. Beachten Sie, dass in diesem Betriebsmodus die Kanalauthentifizierungsblockung und die Zuordnungsregeln die Ergebnisse der Benutzer-ID und der Kennwortprüfung nicht berücksichtigen können.

Beispiel: Das Standardauthentifizierungsinformationsobjekt wird auf **ADOPTCTX(YES)** gesetzt, und der Benutzer `fred` ist angemeldet. Die folgenden beiden CHLAUTH-Regeln sind konfiguriert:

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

Der folgende Befehl wird ausgegeben, mit dem Zweck, den Befehl als den angenommenen Sicherheitskontext des Benutzers bob zu authentifizieren:

```
runmqsc -c -u bob QMGR
```

In der Tat verwendet der Warteschlangenmanager den Sicherheitskontext von `fred`, nicht bob, und die Verbindung schlägt fehl.

Um den Sicherheitskontext von bob verwenden zu können, muss **ChlauthEarlyAdopt** auf Y gesetzt werden.

**PasswordProtection = Kompatibel (Standardwert) |always|optional|warn**

Ab IBM MQ 8.0können Authentifizierungsnachweise, die von IBM MQ client -Anwendungen angegeben werden, wenn sie eine Verbindung zu einem Warteschlangenmanager herstellen, mithilfe der IBM MQ MQCSP-Kennwortschutzfunktion geschützt werden, wenn die Verbindung keine TLS-Verschlüsselung verwendet.

Der MQCSP-Kennwortschutz ist für Test-und Entwicklungszwecke nützlich, da die Verwendung des MQCSP-Kennwortschutzes einfacher ist, als die TLS-Verschlüsselung zu konfigurieren, aber nicht als sicher.

Weitere Informationen zum Schutz von Berechtigungsnachweisen in der MQCSP-Struktur und zu den Werten, die für dieses Attribut festgelegt werden können, enthält der Abschnitt MQCSP-Kennwortschutz.

**IgnoreSeqNumberMismatch = NO (Standardwert) | YES**
Die Nachrichtenkanalagenten (MCAs) an den beiden Enden eines Kanals protokollieren die Anzahl der Nachrichten, die über den Kanal gesendet werden, um die Synchronisation aufrechtzuerhalten. Die Synchronisation kann verloren gehen, wenn beispielsweise die Kanaldefinition an einem Ende gelöscht und anschließend neu erstellt wird. Unter diesen Umständen kann ein RESET CHANNEL erforderlich sein, um zu bestätigen, dass die Synchronisationsdaten verloren gegangen sind, und dem Kanal die Fortsetzung des Startvorgangs zu ermöglichen.

Das Attribut **IgnoreSeqNumberMismatch** muss im Warteschlangenmanager des Empfängers festgelegt sein.

Effektiv führt dieses Attribut einen Befehl zum Zurücksetzen des Kanals auf dem Empfängerkanal aus.

Dieses Attribut steuert über folgende Werte, wie der Warteschlangenmanager bei einer Folgenummernabweichung während des Kanalstarts vorgeht:

**NEIN**
Kanalfolgenummern werden während der Kanalresynchronisation überprüft. Wenn sich die beiden MCAs nicht auf dieselbe Folgenummer verständigen, wird die Fehlernachricht AMQ9526 gemeldet und der Start des Kanals schlägt fehl.

**JA**
Kanalfolgenummern werden während der Kanalresynchronisation überprüft. Wenn sich die beiden MCAs nicht auf dieselbe Folgenummer verständigen, wird jedoch die Warnnachricht AMQ9703 gemeldet und der Start des Kanals wird fortgesetzt. Dieser Attributwert sollte unter normalen

Umständen nicht benötigt werden. Wenn bekannt ist, dass die Synchronisation verloren gegangen ist, beispielsweise bei einem Disaster-Recovery, dann bewirkt diese Option, dass nicht jede Folgenummernabweichung manuell bestätigt werden muss. Die Angabe dieses Werts hat eine ähnliche Wirkung wie die automatische Ausgabe eines **RESET CHANNEL** als Antwort auf jede Folgenummernabweichung durch einen Administrator.

**ChlauthIgnoreUserCase = N (Standardwert) | Y**

Ermöglicht, dass ein Warteschlangenmanager einen Abgleich des Benutezrnamens innerhalb von CHLAUTH-Regeln unabhängig von der Groß-/Kleinschreibung vornimmt. Mit dieser Option ist Folgendes möglich:

- CLNTUSER in CHLAUTH TYPE(USERMAP)-Regeln können ohne Berücksichtigung der Groß- und Kleinschreibung abgeglichen werden
- USERLIST in CHLAUTH TYPE(BLOCKUSER)-Regeln können ohne Berücksichtigung der Groß- und Kleinschreibung abgeglichen werden

Gültige Werte für **ChlauthIgnoreUserCase** sind die folgenden Werte:

**N**
Die Kanalauthentifizierungsregel versucht, die Clientbenutzer-ID unter Berücksichtigung der Groß-/Kleinschreibung abzugleichen; beispielsweise stimmt die Regel, in der CLNTUSER('Fred') angegeben wird, nicht mit 'fred' oder 'FRED' überein, sondern nur mit der Benutzer-ID 'Fred'. Dies ist der Standardwert.

**Y**
Die Kanalauthentifizierungsregel versucht, die Clientbenutzer-ID ohne Berücksichtigung der Groß-/Kleinschreibung abzugleichen; beispielsweise stimmt eine Kanalauthentifizierungsregel mit TYPE(USERMAP) oder TYPE(USERBLOCK), in der CLNTUSER('Fred') angegeben wird, mit jeder Variante der Groß-/Kleinschreibung überein, d. h. die Benutzer-IDs 'Fred', 'FRED' und 'fred' stimmen alle überein.

Beachten Sie, dass beim Ignorieren der Groß-/Kleinschreibung beim Abgleich von Kanalauthentifizierungsregeln möglicherweise eine Übereinstimmung mit mehreren Regeln möglich ist. In diesem Fall ist die Regel, die übereinstimmt, nicht definiert. Wenn beispielsweise der Benutzer 'fred' mit den folgenden Regeln eine Verbindung zu einem Warteschlangenmanager über den CLIENT-Kanal herstellt, kann eine Zuordnung zu 'mquser1' oder 'mquser2' vorgenommen werden:

```
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('fred') USERSRC(MAP) MCAUSER('mquser1')
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('FRED') USERSRC(MAP) MCAUSER('mquser2')
```

Um alle Unsicherheiten bei der Verwendung von `ChlauthIgnoreUserCase=Y` auszuschließen, vermeiden Sie die Definition von sich überschneidenden CHLAUTH-Regeln, die bei der Verwendung eines Abgleichs ohne Berücksichtigung der Groß-/Kleinschreibung zu unterschiedlichem Verhalten führen würde.

**ChlauthIssueWarn = y**

Legen Sie dieses Attribut fest, wenn die Nachricht AMQ9787 generiert werden soll, wenn Sie das Attribut WARN = YES im **SET CHLAUTH** -Befehl festlegen.

## Beispielzeilengruppe

```
Channels:
   MaxChannels=200
   MaxActiveChannels=100
   MQIBindType=STANDARD
   PipelineLength=2
```

**Zugehörige Konzepte**

„Kanalstatus” auf Seite 242
Ein Kanal kann zu einem beliebigen Zeitpunkt in einem von vielen Status sein. Einige Staaten haben auch Unterzustände. Aus einem bestimmten Zustand kann ein Kanal in andere Zustände übergehen.

## Multi Zeilengruppe 'Connection' in der Datei 'qm.ini'

Die Zeilengruppe 'Connection' definiert den Standardbindungstyp.

Verwenden Sie die Zeilengruppe 'Connection' in der Datei `qm.ini` , um den Standardbindungstyp anzugeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer `Extended` -Warteschlangenmanagers verwenden.

**Anmerkung:** Sie müssen eine Connection-Zeilengruppe erstellen, wenn Sie eine benötigen.

**DefaultBindTyp = SHARED (Standardwert) |ISOLIERT**

Wenn **DefaultBindType** auf ISOLATED gesetzt ist, werden Anwendungen und der Warteschlangenmanager in separaten Prozessen ausgeführt, und es werden keine Ressourcen von ihnen gemeinsam genutzt.

Wenn **DefaultBindType** auf SHARED gesetzt ist, werden Anwendungen und der Warteschlangenmanager in separaten Prozessen ausgeführt, aber einige Ressourcen werden von ihnen gemeinsam genutzt.

Die Standardeinstellung ist SHARED.

**Achtung: DefaultBindType** gilt für alle MQCONN-Aufrufe und die, die MQCONNX mit MQCNO_STANDARD_BINDING verwenden.

Das Ändern von **DefaultBindType** kann dazu führen, dass die Leistung einiger Anwendungen abnimmt.

### Beispielzeilengruppe

```
Connection:
  DefaultBindType=SHARED
```

## Multi Diagnosenachrichtenprotokollierung

Die Diagnosenachrichtenprotokolle von IBM MQ sind ein Mechanismus, der es verschiedenen Komponenten des IBM MQ-Systems ermöglicht, Diagnosenachrichten, die sich auf die IBM MQ-Konfiguration und die Laufzeitstatusänderungen und -probleme beziehen, zu melden.

Diese Protokolle werden manchmal als IBM MQ *Fehlerprotokolle* bezeichnen, sie haben aber schon immer nicht nur Fehlernachrichten, sondern auch IBM MQ-Informations- und -Warnnachrichten enthalten. Die drei primären Komponenten von IBM MQ, die ihre Meldungen in diese Protokolle schreiben, sind:

- Warteschlangenmanager
- IBM MQ-Clients
- Der Rest des IBM MQ-Systems

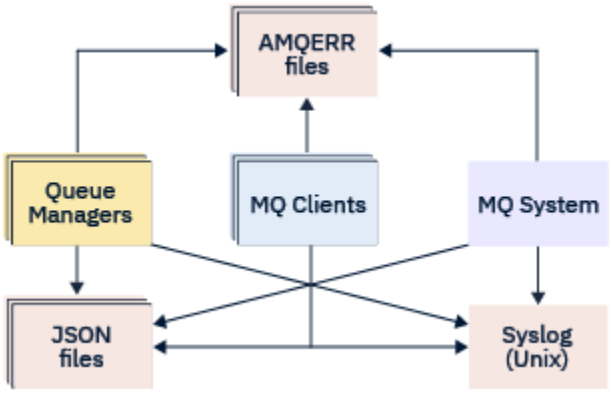
IBM MQ unterstützt die Berichterstellung von Diagnosenachrichten durch eine Reihe unterschiedlicher Methoden, die als *Diagnosenachrichtenservices* bezeichnet werden. Dies ermöglicht einen angepassten Ansatz zum Aufzeichnen und Verarbeiten dieser Informationen:

- AMQERRnn, Protokolldateien
- JSON-formatierte Protokolldateien
- AIX Syslog im JSON-Format

Die JSON-Ausgabe von IBM MQ wird als einzelne JSON-Objekte formatiert, so dass jede einzelne Zeile des JSON-Protokolls oder Syslog-Datensatzes ein gültiges JSON-Objekt darstellt. Das Protokoll als Ganzes ist nicht als einzelnes JSON-Objekt eingebunden.

In der folgenden Abbildung wird dargestellt, dass Warteschlangenmanager, IBM MQ-Clients und das IBM MQ-System *alle* mithilfe der beschriebenen Methoden Diagnosenachrichten melden können.

*Abbildung 5. So können verschiedene Teile von IBM MQ Diagnosenachrichten melden*




## So werden IBM MQ-Diagnoseprotokolle konfiguriert:

Diagnoseprotokolle werden mithilfe von Zeilengruppen in der `qm.ini`-Datei definiert und angepasst, insbesondere für die IBM MQ-Komponente, für die sie erforderlich sind. Jeder eindeutige Protokollendpunkt wird unter seiner eigenen Zeilengruppenüberschrift in der INI-Datei zusammen mit allen in ihr definierten Anpassungen definiert. Zu den Anpassungen gehören:

- Die Größe der Protokolldateien, auf die umgebrochen werden soll, bevor ein Rollover durchgeführt wird. Dies gilt nicht für Syslog.
- Alle Filter, die auf der Wertigkeit der Protokollnachrichten basieren, und
- Alle spezifischen Nachrichtencodes, die unterdrückt werden sollen.

IBM MQ kann so konfiguriert werden, dass alle drei Typen von Protokollendpunkten geschrieben werden können, sodass bestimmte Protokollzeilengruppen bestimmte Rollen erfüllen können. In ähnlicher Weise können mehrere Dateiservices definiert werden. For example:

- Das JSON-Format erleichtert die Syntaxanalyse durch automatisierte Tools in lokalen Umgebungen und Cloud-Umgebungen.
- Die Syslog-Ausgabe ermöglicht es IBM MQ-Komponenten, Diagnoseinformationen in eine gemeinsame Betriebssystemprotokollposition in Übereinstimmung mit anderen Produkten auf dem System zu integrieren.
- Protokollieren von Endpunkten, die auf der Basis der Wertigkeit gefiltert wurden, sodass bestimmte Protokolldateien z. B. nur schwer wiegender Fehler im System aufgezeichnet werden können.

Unabhängig von der Art der Konfiguration der Diagnoseprotokollierung werden die traditionellen Diagnosedateien, die im Systemprotokollverzeichnis von IBM MQ (`/var/mqm/errors/AMQERRnn.log`) und in einem bestimmten Protokollverzeichnis des Warteschlangenmanagers (`/var/mqm/qmgrs/<qmgr_name>/errors/AMQERRnn.log`) gespeichert sind, immer zusätzlich zu jeder anderen verwendeten Protokollierungskonfiguration geschrieben.

Nur für WS-Manager kann die optionale Konfiguration dieser obligatorischen Protokolle durch die Angabe von Attributen der „Zeilengruppen für Diagnosenachrichtenservice” auf Seite 133 ausgeführt werden.

## Unterschiedliche Zeilengruppenbereiche

Die zusätzlichen Zeilengruppen können auf verschiedene Bereiche von IBM MQ angewendet werden.

**Qmgr (qm.ini)**
Gilt für die Protokollnachricht, die vom Warteschlangenmanager generiert wurde.

**System (mqs.ini)**
Gilt für Protokollnachrichten, die vom System generiert werden. Diese Option ist nicht spezifisch für einen Warteschlangenmanager, es sei denn, ein Warteschlangenmanager kann nicht auf seine eigenen Protokolle zugreifen oder diese nicht in seine eigenen Protokolle schreiben.

**Schablonen (mqs.ini)**
Eine oder mehrere Zeilengruppen als Schablonen, die bei der Erstellung eines Warteschlangenmanagers in `qm.ini` kopiert werden.

**Client (mqclient.ini)**
Gilt für Clientoperationen, z. B. **`runmqsc`** im Clientmodus für einen fernen Warteschlangenmanager.

## Konvertieren zwischen JSON-formatierten und traditionell formatierten Protokollen

Der Befehl mqrc wurde erweitert, um eine Reihe von Konvertierungen zwischen JSON-und traditionell formatierten Protokollen und zwischen verschiedenen Sprachen zu ermöglichen.

**Zugehörige Verweise**

„Zeilengruppen für Diagnosenachrichtenservice” auf Seite 133
Die Optionen für den Diagnosenachrichtenservice ermöglichen die Anpassung Ihrer IBM MQ-Diagnoseprotokollierung, so dass die Protokollausgabe an verschiedene Protokollendpunkte aus verschiedenen Komponenten von IBM MQ übertragen werden kann.

„QMErrorLog-Zeilengruppe” auf Seite 132
Sie verwenden die Warteschlangenmanagerfehlerprotokollzeilengruppe QMErrorLog in der `qm.ini`-Datei, um die Operation und den Inhalt der IBM MQ-Fehlerprotokolle anzupassen.

„Diagnosenachrichtenservices” auf Seite 136
Die folgenden Diagnosenachrichtenservices und ihre servicespezifischen Attribute, die in den Zeilengruppen DiagnosticSystemMessages, DiagnosticMessages und DiagnosticMessagesTemplate der Konfigurationsdateien angegeben sind, können definiert werden:

### Multi *QMErrorLog-Zeilengruppe*

Sie verwenden die Warteschlangenmanagerfehlerprotokollzeilengruppe QMErrorLog in der `qm.ini`-Datei, um die Operation und den Inhalt der IBM MQ-Fehlerprotokolle anzupassen.

Der Service QMErrorLog ist der traditionelle IBM MQ-Protokollierungsservice, der für die Ausgabe von Diagnosenachrichten in Bezug auf den Warteschlangenmanager verwendet wird. Der QMErrorLog-Service wird kontinuierlich ausgeführt und kann nicht inaktiviert werden, kann aber in einem gewissen Umfang angepasst werden.

Sie können die Zeilengruppe QMErrorLog in der Datei `qm.ini` verwenden, um bestimmte Nachrichten vom Schreiben in das Fehlerprotokoll des WS-Managers auszuschließen. Sie können auch für ein definiertes Zeitintervall vorgeben, dass bestimmte Nachrichten nicht in das Fehlerprotokoll geschrieben werden sollen.

Windows Linux Statt die `qm.ini`-Datei direkt zu bearbeiten, können Sie alternativ die Seite 'Erweiterte Warteschlangenmanagereigenschaften' in IBM MQ Explorer verwenden, um Nachrichten mit den Attributen **Ausgeschlossene Nachrichten**, **Unterdrückte Nachrichten** und **Intervall für unterdrückte Nachrichten** auszuschließen und zu unterdrücken.

**Achtung:**

- Windows Mit dem IBM MQ Explorer können Sie die Änderungen nur dann vornehmen, wenn Sie auf der Windows-Plattform mit einem lokalen Warteschlangenmanager arbeiten.
- Die Zeilengruppe 'QMErrorLog' gilt nicht für die IBM MQ-Systemkonfigurationsdatei `mqs.ini` oder die Clientkonfigurationsdatei, die im Allgemeinen `mqclient.ini` genannt wird.

Die folgenden Attribute können in die Zeilengruppe QMErrorLog aufgenommen werden:

**ErrorLogSize= *maximale_Größe***
Gibt die Größe des Fehlerprotokolls des Warteschlangenmanagers an, das in die Sicherung kopiert wird *maxsize* muss im Bereich von 32768 bis 2147483648 Byte liegen. Wenn **ErrorLogSize** nicht angegeben wird, wird der Standardwert von 33554432 Byte (32 MB) verwendet.

Sie können dieses Attribut verwenden, um die maximale Größe zurück auf das vorherige Maximum von 2 MB zu reduzieren, falls erforderlich.

Sie können die Größe des Protokolls mit der Umgebungsvariablen **MQMAXERRORLOGSIZE** festlegen.

**ExcludeMessage= *msgIds***
Gibt Nachrichten an, die nicht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden sollen.

Weitere Informationen finden Sie unter ExcludeMessage in „Zeilengruppen für Diagnosenachrichtenservice" auf Seite 133.

**SuppressMessage= *msgIds***
Gibt Nachrichten an, die nur einmal in einem angegebenen Zeitintervall in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden. Wenn dieselbe Nachrichten-ID sowohl in `SuppressMessage` als auch in `ExcludeMessage` angegeben ist, wird die Nachricht ausgeschlossen.

Diese Option ist für Diagnosenachrichtenservices, die in `mqclient.ini` definiert sind, nicht anwendbar. Weitere Informationen finden Sie unter Unterdrückungsnachricht in „Zeilengruppen für Diagnosenachrichtenservice" auf Seite 133.

**SuppressInterval= *length***
Gibt das Zeitintervall (in Sekunden) an, in dem Nachrichten, die in `SuppressMessage` angegeben sind, nur einmal in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden. *length* muss im Bereich von 1 bis 86400 Sekunden liegen. Wenn `SuppressInterval` nicht angegeben ist, wird der Standardwert von 30 Sekunden verwendet.

## Beispielzeilengruppe

```
QMErrorLog:
   ErrorLogSize=262144
   ExcludeMessage=7234
   SuppressMessage=9001,9002,9202
   SuppressInterval=30
```

### Zugehörige Konzepte

„Warteschlangenmanagerkonfigurationsdateien, qm.ini" auf Seite 104
Eine WS-Manager-Konfigurationsdatei, `qm.ini`, enthält Informationen, die für einen bestimmten Warteschlangenmanager relevant sind. Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

### Zugehörige Verweise

„Zeilengruppen für Diagnosenachrichtenservice" auf Seite 133
Die Optionen für den Diagnosenachrichtenservice ermöglichen die Anpassung Ihrer IBM MQ-Diagnoseprotokollierung, so dass die Protokollausgabe an verschiedene Protokollendpunkte aus verschiedenen Komponenten von IBM MQ übertragen werden kann.

## Multi *Zeilengruppen für Diagnosenachrichtenservice*

Die Optionen für den Diagnosenachrichtenservice ermöglichen die Anpassung Ihrer IBM MQ-Diagnoseprotokollierung, so dass die Protokollausgabe an verschiedene Protokollendpunkte aus verschiedenen Komponenten von IBM MQ übertragen werden kann.

Sie aktivieren zusätzliche Diagnosenachrichtenservices, indem Sie eine Zeilengruppe mit einem der folgenden Namen verwenden:

- **DiagnosticSystemMessages**

Definiert die Services, die verwendet werden, wenn eine Diagnosenachricht generiert wird, die in das Systemfehlerprotokoll geht. Gültig in den `mqs.ini`- oder `mqclient.ini`-Dateien.

Clientanwendungen verwenden eine Zeilengruppe **DiagnosticSystemMessages** in der Datei mqclient.ini und in der Zeilengruppe mqs.ini steuert die Zeilengruppe **DiagnosticSystemMessages** Nachrichten für eine Serveranwendung, die keinen Warteschlangenmanagerkontext hat.

Es ist möglich, einen WS-Manager und Anwendungen zu konfigurieren, die zusätzlich alle Nachrichten in den Syslog-Service schreiben.

- **DiagnosticMessages**

  Definiert die Services, die verwendet werden, wenn eine Diagnosenachricht generiert wird, die in das Fehlerprotokoll des Warteschlangenmanagers eingeht. Nur in der Datei `qm.ini` gültig.

- **DiagnosticMessagesTemplate**

  Eine Zeilengruppe, die aus der Datei `mqs.ini` in die Datei **DiagnosticMessages** in der Datei `qm.ini` kopiert wird, wenn ein Warteschlangenmanager erstellt wird

Verwenden Sie den Befehl mqrc , um Diagnosenachrichten anzuzeigen.

## Attribute der Zeilengruppen

**Achtung:** `Service` und der Name einer Zeilengruppe sind obligatorisch.

**name= <Zeilengruppenname>**
Der Name einer Zeilengruppe. Der Wert muss in einer ini-Datei eindeutig sein.

**Service= *Servicetyp***

Dieses Attribut definiert einen Service, bei dem der Name des Service nicht von der Groß-/Kleinschreibung abhängig ist, die von dieser Zeilengruppe aktiviert wird.

Wenn Sie `syslog` beispielsweise als zusätzlichen Service aktivieren möchten, geben Sie Folgendes ein:

```
Service=syslog
```

Siehe „Diagnosenachrichtenservices” auf Seite 136 und die zugehörigen spezifischen Attribute, die für die Zeilengruppen für den Diagnosenachrichtenservice zur Verfügung stehen.

Sie können den Zeilengruppen die folgenden optionalen Attribute hinzufügen:

- ExcludeMessage
- SuppressMessage
- SuppressInterval
- „Severities” auf Seite 136

**ExcludeMessage= *msgIds***
Gibt Nachrichten an, die nicht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden sollen. Wenn Ihr IBM MQ-System stark ausgelastet ist und viele Kanäle gestoppt und gestartet werden, wird eine große Anzahl an Informationsnachrichten an die z/OS-Konsole und an das Hardcopy-Protokoll gesendet. Die IBM MQ-IMS-Bridge und der Puffermanager können auch eine große Anzahl an Informationsnachrichten erzeugen, daher können Sie durch Ausschließen von Nachrichten bei Bedarf verhindern, dass Sie eine unnötig große Anzahl von Nachrichten empfangen. *msgIds* enthält eine durch Kommas getrennte Liste mit Nachrichten-IDs, die aus den folgenden Nachrichten bestehen:

5211-Maximale Länge des Eigenschaftsnamens überschritten.
5973-Verteilte Publish/Subscribe-Subskription unterdrückt
5974-Verteilte Publish/Subscribe-Veröffentlichung unterdrückt
6254-Das System konnte die gemeinsam genutzte Bibliothek nicht dynamisch laden.

IBM i 7163 - Nachricht 'Job gestartet (nur IBM i)
7234-Anzahl der geladenen Nachrichten
8245-Entität weist nicht genügend Berechtigung zum Anzeigen des Objekts auf
9001-Kanalprogramm normal beendet
9002-Kanalprogramm gestartet
9202-Ferner Host nicht verfügbar
9208-Fehler beim Empfangen vom Host
9209-Verbindung geschlossen
9228-Kanalantworter kann nicht gestartet werden
9489-Maximale Anzahl der SVRCONN-Instanzen überschritten
9490-SVRCONN max-Instanzen pro Clientgrenzwert überschritten
9508-Verbindung zum WS-Manager kann nicht hergestellt werden
9524-Ferner WS-Manager nicht verfügbar
9528-Benutzer hat das Schließen des Kanals angefordert
9545-Intervall für Trennen der Verbindung abgelaufen
9558-Ferner Kanal ist nicht verfügbar
9637-Kanal fehlt ein Zertifikat
9776-Kanal wurde von Benutzer-ID blockiert
9777-Kanal wurde durch NOACCESS-Zuordnung blockiert
9782-Verbindung wurde durch Adresse blockiert
9999-Kanalprogramm abnormal beendet

**SuppressMessage= *msgIds***

Gibt Nachrichten an, die nur einmal in einem angegebenen Zeitintervall in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden. Wenn Ihr IBM MQ-System stark ausgelastet ist und viele Kanäle gestoppt und gestartet werden, wird eine große Anzahl an Informationsnachrichten an die z/OS-Konsole und an das Hardcopy-Protokoll gesendet. Die IBM MQ - IMS-Bridge und der Puffermanager können auch eine große Anzahl an Informationsnachrichten erzeugen. Durch Unterdrücken von Nachrichten wird bei Bedarf verhindert, dass Sie eine Reihe von sich wiederholenden Nachrichten empfangen können. Das Zeitintervall wird durch `SuppressInterval` angegeben. *msgIds* enthält eine durch Kommas getrennte Liste mit Nachrichten-IDs aus den folgenden:

5211-Maximale Länge des Eigenschaftsnamens überschritten.
5973-Verteilte Publish/Subscribe-Subskription unterdrückt
5974-Verteilte Publish/Subscribe-Veröffentlichung unterdrückt
6254-Das System konnte die gemeinsam genutzte Bibliothek nicht dynamisch laden.
IBM i 7163 - Nachricht 'Job gestartet (nur IBM i)
7234-Anzahl der geladenen Nachrichten
8245-Entität weist nicht genügend Berechtigung zum Anzeigen des Objekts auf
9001-Kanalprogramm normal beendet
9002-Kanalprogramm gestartet
9202-Ferner Host nicht verfügbar
9208-Fehler beim Empfangen vom Host
9209-Verbindung geschlossen
9228-Kanalantworter kann nicht gestartet werden
9489-Maximale Anzahl der SVRCONN-Instanzen überschritten
9490-SVRCONN max-Instanzen pro Clientgrenzwert überschritten
9508-Verbindung zum WS-Manager kann nicht hergestellt werden
9524-Ferner WS-Manager nicht verfügbar
9528-Benutzer hat das Schließen des Kanals angefordert
9545-Intervall für Trennen der Verbindung abgelaufen
9558-Ferner Kanal ist nicht verfügbar
9637-Kanal fehlt ein Zertifikat

9776-Kanal wurde von Benutzer-ID blockiert
9777-Kanal wurde durch NOACCESS-Zuordnung blockiert
9782-Verbindung wurde durch Adresse blockiert
9999-Kanalprogramm abnormal beendet

Wenn dieselbe Nachrichten-ID sowohl in `SuppressMessage` als auch in `ExcludeMessage` angegeben ist, wird die Nachricht ausgeschlossen.

Diese Option ist für Diagnosenachrichtenservices, die in `MQ client.ini` definiert sind, nicht anwendbar.

**SuppressInterval=** ***length***
Gibt das Zeitintervall in Sekunden an, in dem Nachrichten, die in **SuppressMessage** angegeben sind, nur einmal in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden. *length* muss im Bereich von 1 bis 86400 Sekunden liegen. Wenn Sie **SuppressInterval** nicht angeben, wird der Standardwert von 30 Sekunden verwendet.

**Severities**
Eine durch Kommas getrennte Liste mit Wertigkeitsstufen, bei der der Name der Wertigkeitsstufe nicht zwischen Groß-und Groß-/Kleinschreibung beachtet werden muss. Zulässige Werte sind:

- I (oder Information oder 0)
- W (oder Warnung oder 10)
- E (oder Fehler oder 20 und 30)
- S (oder Stop oder 40)
- T (oder System oder 50)

**Anmerkungen:**

1. Der Standardwert ist `all`
2. Es werden nur Nachrichten in den ausgewählten Bewertungsstufen dem Service angezeigt.

   Alternativ dazu können Sie das Pluszeichen (+) verwenden, das die angegebene Fehlerstufe und alle höheren Ebenen anzeigt. So können Sie beispielsweise alle Fehler anzeigen:

```
Severities=E+
```

**Zugehörige Verweise**
„QMErrorLog-Zeilengruppe” auf Seite 132
Sie verwenden die Warteschlangenmanagerfehlerprotokollzeilengruppe QMErrorLog in der `qm.ini`-Datei, um die Operation und den Inhalt der IBM MQ-Fehlerprotokolle anzupassen.

„Diagnosenachrichtenservices” auf Seite 136
Die folgenden Diagnosenachrichtenservices und ihre servicespezifischen Attribute, die in den Zeilengruppen DiagnosticSystemMessages, DiagnosticMessages und DiagnosticMessagesTemplate der Konfigurationsdateien angegeben sind, können definiert werden:

*Diagnosenachrichtenservices*
Die folgenden Diagnosenachrichtenservices und ihre servicespezifischen Attribute, die in den Zeilengruppen DiagnosticSystemMessages, DiagnosticMessages und DiagnosticMessagesTemplate der Konfigurationsdateien angegeben sind, können definiert werden:

Die folgenden Diagnosenachrichtenservices sind definiert:

**Datei**
Dieser Service sendet alle ungefilterten Nachrichten an eine Datei auf ähnliche Weise an den Service QMErrorLog. Je nach angegebener **Format** wird entweder das vorhandene Textformat oder das angegebene JSON-Format verwendet. Standardmäßig gibt es drei Dateien mit den Namen `AMQERR01.LOG`, `AMQERR02.LOG`und `AMQERR03.LOG` oder `AMQERR01.json`, `AMQERR02.json`und `AMQERR03.json`, abhängig von der Eigenschaft **Format** , und diese Rollover basieren auf der konfigurierten Größe.

Die folgenden Attribute werden nur in einer Datei-Zeilengruppe unterstützt:

**FilePath**
Der Pfad zu der Position, in die Protokolldateien geschrieben wurden. Der Standardwert ist dieselbe Position wie die AMQERR01.log-Dateien, d. h. System- oder Warteschlangenmanager. Der Pfad muss absolut sein, er kann jedoch austauschbare Einfügungen enthalten. For example:

**+ MQ_Q_MGR_DATA_PATH +**
Der vollständige Pfad zum übergeordneten Verzeichnis des Nachrichtenverzeichnisses des Warteschlangenmanagerdiagnoseprogramms. Die Standardwerte lauten wie folgt:

- Linux AIX Auf AIX and Linux-Plattformen: `/var/mqm/qmgrs/<QM_name>`
- Windows unter Windows, `C:\Program Data\IBM\MQ\qmgrs\<QM_name>`

**+ MQ_DATA_PATH +**
Der vollständige Pfad zum übergeordneten Element des Nachrichtenverzeichnisses für Systemdiagnosen. Die Standardwerte lauten wie folgt:

- Linux AIX Auf AIX and Linux-Plattformen:`/var/mqm`
- Windows Unter Windows: `C:\Program Data\IBM\MQ`

Sie müssen diesen Pfad mit den entsprechenden Berechtigungen erstellen, wenn er das vorhandene Fehlerverzeichnis nicht verwendet.

**FilePrefix**
Das Präfix der Protokolldateien. Der Standardwert ist AMQERR.

**FileSize**
Die Größe, in der sich das Protokoll überrollt. Der Standardwert ist 32MB, wie bei der Eigenschaft **ErrorLogSize** der „QMErrorLog-Zeilengruppe" auf Seite 132, die semantisch identisch ist.

**Anmerkung:** Die Eigenschaft **ErrorLogSize** gilt nur für den Standardfehlerprotokollservice, nicht für angepasste Diagnoseservices.

Sie können die Größe des Protokolls mit der Umgebungsvariablen **MQMAXERRORLOGSIZE** festlegen.

**Format**
Das Format der Datei. Der Wert kann entweder *text* (für zusätzliche QMErrorLog-Stilservices) oder *json* sein, der Standardwert ist.

Das Suffix der Datei ist entweder `.LOG` oder `.json` abhängig von der Einstellung dieses Attributs.

Bearbeiten Sie beispielsweise die `qm.ini`-Datei des Warteschlangenmanagers, und fügen Sie die folgende Zeilengruppe hinzu:

```
DiagnosticMessages:
   Service = File
   Name = JSONLogs
   Format = json
   FilePrefix = AMQERR
```

Nach dem Neustart hat der Warteschlangenmanager `AMQERR0x.json`-Dateien in seinem Verzeichnis ERRORS.

Sie können mehrere Dateiservices definieren. Dies ermöglicht die Konfiguration wie in den folgenden Beispielen gezeigt, in denen Nachrichten unterschiedlicher Tags über verschiedene Protokollgruppen aufgeteilt werden:

```
DiagnosticMessages:
  Name=ErrorsToFile
  Service=File
  Severities=E+
  FilePrefix=OnlyErrors
```

```
DiagnosticMessages:
  Name=NonErrorstoFile
  Service=File
  Severities=1 W
  FilePrefix=Information
```

Linux AIX **Syslog**

Der Syslog-Service ist unter Windows oder IBM i nicht verfügbar.

Sie können nur einen Syslog-Service definieren, und der Syslog-Service sendet alle ungefilterten Nachrichten mithilfe der Spezifikation für Diagnosenachrichten im JSON-Format an das Systemprotokoll. Die Informationen werden dem Systemprotokoll in der in der Tabelle angegebenen Reihenfolge hinzugefügt, beginnend mit dem Parameter msgID und den Einfügungen.

Der Schweregrad der Nachricht wird auf folgende Weise auf die Systemprotokollstufe (syslog) abgebildet:

*Tabelle 12. Zuordnung der Nachrichtenbewertung zur Syslog-Ebene*

| **Bewertung** | **Ebene** |
|---|---|
| 0 | LOG_INFO |
| 10 | LOG_WARNUNG |
| 20 | LOG_ERR |
| 30 | LOG_ERR |
| 40 | LOG_ALERT |
| 50 | LOG_ALERT |

Das folgende Attribut wird nur in einer syslog-Zeilengruppe unterstützt:

**Ident**
Definiert den Wert **ident**, der den Syslog-Einträgen zugeordnet ist. Der Standardwert ist *ibm-mq*.

Das folgende Beispiel zeigt Fehlernachrichten, die an Syslog gesendet werden:

```
DiagnosticMessages:
  Name=ErrorsToSyslog
  Ident=mq
  Service=Syslog
  Severities=E+
```

Weitere Informationen zu generischen Zeilengruppenattributen finden Sie unter „Zeilengruppen für Diagnosenachrichtenservice” auf Seite 133.

**Anmerkungen:**

1. Nur für den Dateiservice können mehrere Zeilengruppen mit jeweils einem anderen Namen vorhanden sein. Nur die Definition, die den endgültigen Namen in der Sequenz verwendet, wird wirksam.
2. Änderungen am Wert einer Zeilengruppe treten erst in Kraft, wenn der WS-Manager erneut gestartet wird.

## Multi Zeilengruppe 'ExitPath' in der Datei 'qm.ini'

Die Zeilengruppe ExitPath gibt den Pfad für Benutzerexitprogramme auf dem Warteschlangenmanagersystem an.

Mithilfe der Zeilengruppe `ExitPath` in der Datei `qm.ini` können Sie den Pfad für Benutzerexitprogramme auf dem Warteschlangenmanagersystem angeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer `Exits` -Warteschlangenmanagers verwenden.

**ExitsDefaultPath= *string***
Das Attribut "ExitsDefaultPath" gibt die Position von an:

- 32-Bit-Kanalexits für Clients
- 32-Bit-Kanalexits und Datenkonvertierungsexits für Server
- Nicht qualifizierte XA-Switchloaddateien

**ExitsDefaultPath64 = *string***
Das Attribut "ExitsDefaultPath64" gibt die Position von an:

- 64-Bit-Kanalexits für Clients
- 64-Bit-Kanalexits und Datenkonvertierungsexits für Server
- Nicht qualifizierte XA-Switchloaddateien

### Beispielzeilengruppe

```
ExitPath:
   ExitsDefaultPath=/var/mqm/exits
   ExitsDefaultPath64=/var/mqm/exits64
```

## Multi Zeilengruppe 'ExitPropertiesLocal' in der Datei 'qm.ini'

Die lokale Zeilengruppe ExitPropertiesgibt Informationen zu Exiteigenschaften auf einem Warteschlangenmanager an.

Mithilfe der Zeilengruppe ExitPropertiesLocal in der Datei `qm.ini` können Sie Informationen zu Exiteigenschaften auf einem Warteschlangenmanager angeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer -Clusterwarteschlangenmanagers verwenden.

Windows Unter Windows können Sie diese Informationen auch mit dem Befehl **amqmdain** angeben.

Diese Einstellung wird standardmäßig aus dem Attribut **CLWLMode** in der Zeilengruppe 'ExitProperties' der maschinenweiten Konfiguration übernommen (siehe „Zeilengruppe 'ExitProperties' in der Datei 'mqs.ini'" auf Seite 99 ). Ändern Sie diese Einstellung nur, wenn Sie diesen Warteschlangenmanager auf eine andere Weise konfigurieren möchten. Dieser Wert kann für einzelne Warteschlangenmanager mit dem Attribut "Clusterworkloadmodus" auf der Eigenschaftenseite "Clusterwarteschlangenmanager" überschrieben werden.

Mithilfe der Zeilengruppe ExitProperties in der Datei `mqs.ini` können Sie Konfigurationsoptionen angeben, die von Exitprogrammen des Warteschlangenmanagers verwendet werden.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite IBM MQ Explorer `Extended` IBM MQ verwenden.

**CLWLMode = SAFE (Standardwert) | FAST**
Mit dem CLWL-Exit (CLWL-Cluster Workload) können Sie angeben, welche Clusterwarteschlange im Cluster als Antwort auf einen MQI-Aufruf geöffnet werden soll (z. B. `MQOPEN`, `MQPUT`). Der CLWL-Exit wird abhängig vom Wert, den Sie im Attribut **CLWLMode** angeben, entweder im FAST-Modus oder im SAFE-Modus ausgeführt. Wenn Sie das Attribut **CLWLMode** nicht angeben, wird der Exit für Clusterauslastung im SAFE-Modus ausgeführt.

**SAFE**
Führen Sie den CLWL-Exit in einem separaten Prozess aus dem Warteschlangenmanager aus. Dies ist die Standardeinstellung.

Tritt bei der Ausführung im SAFE-Modus ein Problem mit dem vom Benutzer geschriebenen CLWL-Exit auf, geschieht Folgendes:

- Der CLWL-Serverprozess (amqzlwa0) schlägt fehl.
- Der WS-Manager startet den CLWL-Serverprozess erneut.
- Der Fehler wird Ihnen im Fehlerprotokoll gemeldet. Wenn ein MQI-Aufruf in Bearbeitung ist, erhalten Sie eine Benachrichtigung in Form eines Rückkehrcodes.

Die Integrität des Warteschlangenmanagers bleibt erhalten.

**Anmerkung:** Die Ausführung des CLWL-Exits in einem separaten Prozess kann sich auf die Leistung auswirken.

**FAST**
Führen Sie den Cluster-Exit inline im WS-Manager-Prozess aus.

Wenn Sie diese Option angeben, wird die Leistung verbessert, da die Prozesse, die mit der Ausführung im SAFE-Modus verbunden sind, vermieden werden, dies jedoch zu Lasten der Integrität des Warteschlangenmanagers geht. Sie sollten den CLWL-Exit nur im FAST-Modus ausführen, wenn Sie überzeugt sind, dass es keine Probleme mit Ihrem CLWL-Exit gibt, und Sie sind besonders besorgt über die Leistung.

Tritt ein Problem auf, wenn der CLWL-Exit im FAST-Modus ausgeführt wird, schlägt der Warteschlangenmanager fehl, und Sie laufen Gefahr, dass die Integrität des Warteschlangenmanagers beeinträchtigt wird, der beeinträchtigt wird.

### Beispielzeilengruppe

```
ExitPropertiesLocal:
      CLWLMode=SAFE
```

## Linux IBM i AIX Zeilengruppe 'Filesystem' in der Datei 'qm.ini'

Die Zeilengruppe 'Filesystem' gibt an, ob die in den Fehlerprotokollen des Warteschlangenmanagers festgelegten Berechtigungen unverändert bleiben oder auf ihre Standardwerte zurückgesetzt werden sollen.

Die Standardberechtigungen, die in den Fehlerprotokolldateien festgelegt sind, werden in den meisten Fällen nützlich sein. Daher ist es für die meisten IBM MQ-Administratoren nicht erforderlich, sie zu ändern.

Es kann jedoch sein, dass Ihr IBM MQ -Administrator die Berechtigungen für seine Fehlerprotokolldateien ändern will. In diesem Fall sollte er die Zeilengruppenoption `Filesystem` auf **ValidateAuth**=*No*setzen, was bewirkt, dass der Warteschlangenmanager die Berechtigungen danach unverändert lässt.

Das Standardverhalten (ohne **ValidateAuth**=*Nein*) besteht darin, dass der Warteschlangenmanager die Dateiberechtigungen der Warteschlangenmanager-Fehlerprotokolle überprüft und sie an ihre Standardwerte zurückändert. Diese Prüfung kann jederzeit erfolgen, auch während einer End- oder Startoperation eines Warteschlangenmanagers.

### Beispielzeilengruppe

```
Filesystem:
      ValidateAuth=No
```

## Multi Zeilengruppe 'Log' in der Datei 'qm.ini'

Die Zeilengruppe 'Log' gibt Informationen zur Protokollierung auf einem Warteschlangenmanager an.

Verwenden Sie die Zeilengruppe 'Log' in der Datei `qm.ini` , um Informationen zur Protokollierung auf einem Warteschlangenmanager anzugeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer **-Protokollwarteschlangenmanagers** verwenden.

Standardmäßig werden diese Einstellungen von den Einstellungen übernommen, die für die Standardprotokolleinstellungen für den Warteschlangenmanager angegeben sind (siehe „Zeilengruppe 'LogDefaults' in der Datei 'mqs.ini'" auf Seite 100). Ändern Sie diese Einstellungen nur, wenn Sie diesen Warteschlangenmanager auf eine andere Weise konfigurieren möchten.

Weitere Informationen zur Berechnung von Protokollgrößen finden Sie in „Berechnen der Größe des Protokolls" auf Seite 698.

**Anmerkung:** Die in der folgenden Parameterliste angegebenen Grenzwerte werden von IBM MQfestgelegt. Durch die Begrenzung des Betriebssystems kann die maximal mögliche Protokollgröße reduziert werden.

**LogPrimaryFiles = 3 (Standardwert) |*2-254* ( Windows ) |*2-510* ( AIX and Linux -Systeme)**
Die Protokolldateien, die beim Erstellen des Warteschlangenmanagers zugeordnet werden.

Die minimale Anzahl der von Ihnen verwendeten primären Protokolldateien beträgt 2 und die maximale Anzahl 254 unter Windows bzw. 510 auf AIX and Linux-Systemen. Der Standardwert ist 3.

Die Gesamtzahl der primären und sekundären Protokolldateien darf nicht größer sein als 255 (unter Windows) bzw. 511 (auf AIX and Linux-Systemen) und nicht kleiner als 3.

Der Wert wird geprüft, wenn der WS-Manager erstellt oder gestartet wird. Sie können sie ändern, nachdem der WS-Manager erstellt wurde. Eine Änderung des Werts ist jedoch erst wirksam, wenn der Warteschlangenmanager erneut gestartet wird und der Effekt möglicherweise nicht sofort ausgeführt wird.

**LogSecondary-Dateien = 2 (Standardwert) |*1-253* ( Windows ) |*1-509* ( AIX and Linux -Systeme)**
Die Protokolldateien, die zugeordnet werden, wenn die Primärdateien erschöpft sind.

Die minimale Anzahl der sekundären Protokolldateien beträgt 1 und die maximale Anzahl beträgt 253 unter Windows bzw. 509 auf AIX and Linux-Systemen. Die Standardanzahl ist 2.

Die Gesamtzahl der primären und sekundären Protokolldateien darf nicht größer sein als 255 (unter Windows) bzw. 511 (auf AIX and Linux-Systemen) und nicht kleiner als 3.

Der Wert wird geprüft, wenn der Warteschlangenmanager gestartet wird. Sie können diesen Wert ändern, aber Änderungen werden erst wirksam, wenn der Warteschlangenmanager erneut gestartet wird, und selbst dann wird der Effekt möglicherweise nicht sofort wirksam.

**LogFilePages= *number***
Die Protokolldaten werden in einer Reihe von Dateien mit dem Namen "Protokolldateien" festgehalten. Die Protokolldateigröße wird in Einheiten von 4-KB-Seiten angegeben.

Die Standardanzahl der Protokolldateiseiten beträgt 4096, wobei eine Protokolldateigröße von 16 MB angegeben wird.

Auf AIX and Linux-Systemen beträgt die Mindestanzahl der Protokolldateiseiten 64, und bei Windows beträgt die Mindestanzahl der Protokolldateiseiten 32; in beiden Fällen beträgt die maximale Anzahl 65 535.

**Anmerkung:** Die Größe der Protokolldateien, die während der Erstellung des Warteschlangenmanagers angegeben werden, kann für Warteschlangenmanager nicht geändert werden.

**LogType = CIRCULAR (Standardwert) | LINEAR| REPLICATED**
Der Typ der Protokollierung, der vom Warteschlangenmanager verwendet werden soll. Der Standardwert ist CIRCULAR. Weitere Informationen zum Erstellen eines Warteschlangenmanagers mit dem erforderlichen Protokollierungstyp finden Sie in der Beschreibung des Attributs **LogType** im Abschnitt „Zeilengruppe 'LogDefaults' in der Datei 'mqs.ini'" auf Seite 100.

**CIRCULAR**
Starten Sie die Wiederherstellung nach einem Neustart, indem Sie das Protokoll verwenden, um Transaktionen rückgängig zu machen, die in Bearbeitung waren, als das System gestoppt wurde.

Eine genauere Beschreibung des Protokolltyps CIRCULAR (Umlaufprotokollierung) finden Sie im Abschnitt „Typen der Protokollierung” auf Seite 692.

**LINEAR**
Sowohl für die Wiederherstellung nach einem Neustart als auch für die Datenträger-oder Vorwärtswiederherstellung (durch das Erstellen verlorener oder beschädigter Daten durch Wiedergabe des Inhalts des Protokolls).

Eine genauere Beschreibung des Protokolltyps LINEAR (lineare Protokollierung) finden Sie im Abschnitt „Typen der Protokollierung” auf Seite 692.

CP4I **REPLICATED**
Wird von einer nativen HA-Gruppe verwendet, um Protokolldaten von der aktiven Instanz auf den Replikatinstanzen zu vervielfältigen.

Eine genauere Beschreibung des Protokolltyps REPLICATED (Replizierung) finden Sie im Abschnitt „Typen der Protokollierung” auf Seite 692.

**Anmerkung:** Die **LogType** eines Warteschlangenmanagers kann nicht geändert werden, indem dieses Attribut in der Datei `qm.ini` geändert wird. Um den **LogType** eines Warteschlangenmanagers zu ändern, müssen Sie den Befehl **migmqlog** verwenden.

**LogBufferPages=0 (Standardwert) |*0-4096***
Die Größe des Speichers, der den Pufferdatensätzen für das Schreiben zugeordnet ist, wobei die Größe der Puffer in Einheiten von 4-KB-Seiten angegeben wird.

Die Mindestanzahl der Pufferseiten beträgt 18 und der Maximalwert 4096. Größere Puffer führen zu einem höheren Durchsatz, insbesondere bei größeren Nachrichten.

Wenn Sie 0 (die Standardeinstellung) angeben, wählt der Warteschlangenmanager die Größe aus.

Wenn Sie eine Zahl im Bereich von 1 bis 17 angeben, nimmt der WS-Manager standardmäßig den Wert 18 (72 KB) an. Wenn Sie eine Zahl im Bereich von 18 bis 4096 angeben, verwendet der Warteschlangenmanager die angegebene Zahl, um die zugeordnete Speicherkapazität festzulegen.

Der Wert wird geprüft, wenn der Warteschlangenmanager gestartet wird. Der Wert kann innerhalb der angegebenen Grenzwerte erhöht oder verringert werden. Eine Änderung des Werts ist jedoch erst wirksam, wenn der Warteschlangenmanager das nächste Mal gestartet wird.

**LogPath= *directory_name***
Das Verzeichnis, in dem sich die Protokolldateien für einen WS-Manager befinden. Dieser muss auf einer lokalen Einheit vorhanden sein, in die der Warteschlangenmanager schreiben kann, und zwar vorzugsweise auf einem anderen Laufwerk aus den Nachrichtenwarteschlangen. Die Angabe eines anderen Laufwerks bietet einen zusätzlichen Schutz im Falle eines Systemausfalls.

Der Standardwert lautet:

- Windows `C:\ProgramData\IBM\MQ\log` in Windows.
- Linux AIX `/var/mqm/log` auf AIX and Linux-Systemen.

Sie können den Namen eines Verzeichnisses mit dem Befehl **crtmqm** mit dem Flag **-ld** angeben. Wenn ein Warteschlangenmanager erstellt wird, wird auch ein Verzeichnis unter dem WS-Manager-Verzeichnis erstellt, und dieses Verzeichnis wird verwendet, um die Protokolldateien zu speichern. Der Name dieses Verzeichnisses basiert auf dem Namen des Warteschlangenmanagers. Dadurch wird sichergestellt, dass der Protokolldateipfad eindeutig ist und dass er auch Einschränkungen bezüglich der Länge von Verzeichnisnamen entspricht.

Wenn Sie `-ld` im Befehl **crtmqm** nicht angeben, wird der Wert des Attributs **LogDefaultPath** verwendet.

Linux AIX Auf AIX and Linux -Systemen müssen die Benutzer-ID 'mqm' und die Gruppe 'mqm' über vollständige Berechtigungen für die Protokolldateien verfügen. Wenn Sie die Position dieser Dateien ändern, müssen Sie diese Berechtigungen selbst zuweisen. Dies ist nicht

erforderlich, wenn sich die Protokolldateien an den Standardpositionen befinden, die im Lieferumfang des Produkts enthalten sind.

**LogWriteIntegrity =SingleWrite|DoubleWrite|TripleWrite (Standardwert)**
Die Methode, die die Protokollfunktion verwendet, um Protokollsätze zuverlässig zu schreiben.

**TripleWrite (Standardwert)**

Beachten Sie, dass Sie DoubleWrite auswählen können, aber wenn Sie dies tun, interpretiert das System diese Option als TripleWrite .

**SingleWrite**
Sie sollten SingleWrite nur verwenden, wenn das Dateisystem und die Einheit, die als Host für das IBM MQ-Wiederherstellungsprotokoll fungiert, explizit die Atomizität von 4-KB-Schreibvorgängen garantiert.

Das heißt, wenn ein Schreiben einer 4-KB-Seite aus irgendeinem Grund fehlschlägt, sind die einzigen beiden möglichen Status entweder das Vorimage oder das Nachimage. Ein Zwischenzustand sollte nicht möglich sein.

**Anmerkung:** Wenn in Ihrer persistenten Workload ausreichend gemeinsamer Zugriff vorhanden ist, ist bei der Einstellung eines anderen Werts als dem Standardwert TripleWrite ein minimaler potenzieller Vorteil vorhanden.

Weitere Informationen finden Sie unter „LogWriteIntegrity-using SingleWrite oder TripleWrite” auf Seite 144.

**LogManagement = Manuell (Standard) | Automatisch | Archivieren**
Die Methode, mit der Protokollextents entweder manuell oder vom Warteschlangenmanager verwaltet werden. Der Standardwert ist Manuell.

Das Attribut gilt nur, wenn **LogType** auf LINEARgesetzt ist.

Wenn Sie den Wert für **LogManagement** ändern, wird die Änderung erst nach einem Neustart des Warteschlangenmanagers wirksam.

Wenn ein nicht erkannter Wert für das Attribut gefunden wird, wird der WS-Manager erst gestartet, wenn der Wert korrigiert wird.

Die Eigenschaft **LogManagement** ist unter IBM i nicht gültig.

**Manuell (Standard)**
Sie verwalten die Protokollspeicherbereiche manuell. Die Angabe dieser Option bedeutet, dass der Warteschlangenmanager keine Protokollspeicherbereiche wiederverwenden oder löschen kann, selbst wenn sie nicht mehr für die Wiederherstellung benötigt werden.

**Automatisch**
Protokollerweiterungen werden automatisch vom Warteschlangenmanager verwaltet. Die Angabe dieser Option bedeutet, dass der Warteschlangenmanager Protokollspeicherbereiche wiederverwenden oder löschen kann, sobald sie nicht mehr für die Wiederherstellung benötigt werden. Für die Archivierung wird keine Vergütung berücksichtigt.

**Archiv**
Protokollspeicherbereiche werden vom Warteschlangenmanager verwaltet, aber Sie müssen den Warteschlangenmanager benachrichtigen, wenn die Archivierung der einzelnen Protokollspeicherbereiche abgeschlossen ist.

Die Angabe dieser Option bedeutet, dass der Warteschlangenmanager keinen Protokollspeicherbereich wiederverwenden oder löschen kann, sobald der Warteschlangenmanager benachrichtigt wurde, dass ein Speicherbereich, der nicht mehr für die Wiederherstellung erforderlich ist, archiviert wurde.

Diese Benachrichtigung wird mit dem MQSC-Befehl **RESET QMGR** oder dem PCF-Befehl Reset Queue Manager ausgeführt.

### Beispielzeilengruppe

```
Log:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=4096
   LogType=CIRCULAR
   LogBufferPages=0
   LogPath=/var/mqm/log/saturn!queue!manager/
```

**Anmerkung:** Der Wert 0 für **LogBufferPages** ergibt einen Wert von 512.

### Multi *LogWriteIntegrity-using SingleWrite oder TripleWrite*

Durch das Festlegen der Option **LogWriteIntegrity** in der Zeilengruppe 'Log' der Datei `qm.ini` wird der Algorithmus bestimmt, der von der Protokollfunktion in IBM MQ verwendet wird, um Protokolleinträge in das Wiederherstellungsprotokoll zu schreiben. Die Standardeinstellung ist `TripleWrite` und diese Einstellung ist in fast jedem möglichen Szenario sicher.

Die Einstellung von **LogWriteIntegrity** wirkt sich nur aus, wenn eine partielle Protokollseite geschrieben werden soll. Bei einem Warteschlangenmanager mit einer angemessenen Anzahl gleichzeitig ablaufender Aktivitäten tritt dieses Szenario selten auf.

### SingleWrite

`SingleWrite` wählt einen Algorithmus aus, der in sehr ungewöhnlichen Umständen eine bessere Leistung als die Standardeinstellung `TripleWrite` erzielt. Die Einstellung `SingleWrite` ist nur dann sicher, wenn die zugrunde liegende Speicherplattform absolut garantieren kann, dass 4KB Seiten, die synchron in das IBM MQ -Wiederherstellungsprotokoll geschrieben werden, atomar geschrieben werden.

Sie sollten die Einstellung `SingleWrite` nur verwenden, wenn das Dateisystem oder die Einheit, das bzw. die das IBM MQ -Wiederherstellungsprotokoll hostet, explizit die Atomizität von 4KB -Schreibvorgängen garantiert. Das heißt, wenn ein Schreiben einer 4-KB-Seite aus einem beliebigen Grund fehlschlägt, sind die einzigen beiden möglichen Status entweder das Vorimage oder das Nachimage und es sollte kein Zwischenstatus möglich sein. In allen anderen Fällen sollten Sie `TripleWrite`verwenden.

Auf einem System mit ausreichend gemeinsamen Zugriff schreibt der Warteschlangenmanager nur vollständige Seiten mit Protokolldaten, und wenn ein hoher Prozentsatz vollständiger Seiten erreicht ist, gibt es zwischen `SingleWrite` und `TripleWrite` keine wesentlichen Leistungsunterschiede.

Wenn auf einem System kaum gemeinsamer Zugriff vorhanden ist, kann es zu einem wesentlichen Leistungsvorteil für `SingleWrite` kommen. Die bevorzugte Lösung ist normalerweise, den gemeinsamen Zugriff zu erhöhen, statt `SingleWrite` zu verwenden.

Beachten Sie, dass es schwierig sein kann, die Atomizität von 4-KB-Schreibvorgängen zuverlässig zu bestimmen, und Änderungen an der zugrunde liegenden Software oder Hardware können eine solche Anmerkung nicht bestätigen.

Wenn Sie sich nicht sicher sind, ob Ihre Speicherinfrastruktur jetzt und zu jedem beliebigen Zeitpunkt in der Zukunft in allen Fällen die erforderlichen Garantien erfüllen kann, sollten Sie `TripleWrite` verwenden.

## Windows Zeilengruppe 'LU62' in der Datei 'qm.ini' (nur Windows)

Die Zeilengruppe LU62 gibt die Protokollkonfigurationsparameter für SNA LU 6.2 an. Diese Parameter überschreiben die Standardattribute für Kanäle.

Mithilfe der Zeilengruppe LU62 in der Datei `qm.ini` können Sie Konfigurationsparameter für das Protokoll SNA LU 6.2 angeben. Sie überschreiben die Standardattribute für Kanäle.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer LU6.2 -Warteschlangenmanagers verwenden.

**TP-Name**
Der Name des TP-Namens, der auf dem fernen Standort gestartet werden soll.

**Library1= *DLLName 1***
Der Name der APPC-DLL.

Der Standardwert ist WCPIC32.

**Library2= *DLLName2***
Wie Library1, wird verwendet, wenn der Code in zwei separaten Bibliotheken gespeichert ist.

Der Standardwert ist WCPIC32.

## CP4I Zeilengruppe NativeHAInstance in der Datei qm.ini

Für IBM MQ in Containern gibt die Zeilengruppe NativeHAInstance an, wie die drei Knoten in einer nativen Hochverfügbarkeitskonfiguration miteinander kommunizieren können.

**Anmerkung:** Diese Informationen gelten nur für Containerumgebungen. Weitere Informationen finden Sie unter Native HA mit IBM MQ Operator konfigurieren oder Native HA-Gruppe erstellen, wenn eigene Container erstellt werden.

Sie fügen drei NativeHAInstance -Zeilengruppen hinzu, eine für jede Warteschlangenmanagerinstanz in der nativen HA-Gruppe (einschließlich der lokalen Instanz). Fügen Sie die folgenden Attribute hinzu:

**Name**
Geben Sie den Instanznamen an, den Sie beim Erstellen der Warteschlangenmanagerinstanz verwendet haben.

**ReplicationAddress**
Geben Sie den Hostnamen, die IPv4 -Schreibweise mit Trennzeichen oder die Adresse im IPv6 -Hexadezimalformat der Instanz an. Sie können die Adresse als Hostnamen, als IPv4 -Schreibweise mit Trennzeichen oder als IPv6 -Adresse im Hexadezimalformat angeben. Die Replikationsadresse muss von jeder Instanz in der Gruppe auflösbar und weiterleitbar sein. Die für die Protokollreplikation zu verwendende Portnummer muss in eckigen Klammern angegeben werden. Beispiel:

```
ReplicationAddress=host1.example.com(4444)
```

### Beispielzeilengruppe

Das folgende Beispiel zeigt die Zeilengruppe NativeHAInstance , die in der Datei qm.ini verwendet wird, um die drei Knoten einer nativen HA-Konfiguration anzugeben.

```
NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

**Zugehörige Konzepte**

„Zeilengruppe NativeHALocalInstance in der Datei qm.ini” auf Seite 145
Für IBM MQ in Containern steuert die Zeilengruppe NativeHALocalInstance die Operation einer nativen HA-Konfiguration.

## CP4I Zeilengruppe NativeHALocalInstance in der Datei qm.ini

Für IBM MQ in Containern steuert die Zeilengruppe NativeHALocalInstance die Operation einer nativen HA-Konfiguration.

**Anmerkung:** Diese Informationen gelten nur für Containerumgebungen. Weitere Informationen finden Sie unter Native HA mit IBM MQ Operator konfigurieren oder Native HA-Gruppe erstellen, wenn eigene Container erstellt werden.

Die Zeilengruppe NativeHALocalInstance wird automatisch zur Datei qm.ini auf jedem Knoten hinzugefügt, wenn Sie eine native Hochverfügbarkeitskonfiguration erstellen. Anschließend können Sie die Datei qm.ini bearbeiten und die Attribute in der Zeilengruppe NativeHALocalInstance anpassen.

**LocalName**
Der Name der Zeilengruppe NativeHALocalInstance , der dem Namen der Protokollreplikatinstanz entnommen wird, der bei der Erstellung des nativen HA-Warteschlangenmanagers angegeben wurde

Sie können optional die folgenden Attribute zur Zeilengruppe NativeHALocalInstance hinzufügen:

**KeyRepository**
V 9.3.0 V 9.3.0 Der vollständige Pfad und der Dateiname des Schlüsselrepositorys, das das digitale Zertifikat enthält, das zum Schutz des Protokollreplikationsdatenverkehrs verwendet wird. Wird die Dateierweiterung nicht angegeben, wird angenommen, dass es sich um .kdbhandelt.

Wenn das Zeilengruppenattribut KeyRepository nicht angegeben wird, werden Protokollreplikationsdaten zwischen Instanzen in Klartext ausgetauscht.

V 9.3.2 **KeyRepositoryPassword**
Das Schlüsselrepository ist mit einem Kennwort geschützt, da es sensible Informationen enthält. Damit IBM MQ auf den Inhalt des Schlüsselrepositorys zugreifen kann, muss das Kennwort des Schlüsselrepositorys abgerufen werden können. Wenn das Kennwort nicht in einer Stashdatei des Schlüsselrepositorys gespeichert ist, können Sie das Kennwort im Attribut KeyRepositoryPassword angeben. For example:

```
KeyRepositoryPassword=passw0rd
```

**Achtung:** Wenn Sie das Kennwort mithilfe dieses Attributs angeben, verschlüsseln Sie das Kennwort mit dem IBM MQ -Kennwortschutzsystem. Weitere Informationen finden Sie unter „Schlüsselrepository-Kennwort verschlüsseln” auf Seite 147.

V 9.3.2 **InitialKeyFile**
Geben Sie dieses Attribut an, wenn das mit dem Attribut KeyRepositoryPassword angegebene Kennwort des Schlüsselrepositorys mit einem bestimmten Anfangsschlüssel verschlüsselt wird. Der Name der Datei, die den ursprünglichen Schlüssel enthält, kann mit dem Parameter **-sf** angegeben werden, wenn der Befehl **runmqicred** zum Verschlüsseln des Kennworts für das Schlüsselrepository verwendet wird.

Setzen Sie den Wert dieses Attributs auf den Namen der Datei, die den Anfangsschlüssel für die Verschlüsselung des Kennworts enthält. Angenommen, eine Datei mit dem Namen mykey.key enthält den ursprünglichen Schlüssel:

```
InitialKeyFile=/mykey.key
```

Weitere Informationen finden Sie unter „Schlüsselrepository-Kennwort verschlüsseln” auf Seite 147.

**CertificateLabel**
Die Zertifikatsbezeichnung, die das digitale Zertifikat angibt, das für den Schutz des Protokollreplikationsverkehrs verwendet werden soll. Wenn KeyRepository angegeben wird, aber CertificateLabel weggelassen wird, wird der Standardwert ibmwebspheremq*queue_manager* verwendet.

**CipherSpec**
Die zum Schutz des Protokollreplikationsdatenverkehrs zu verwendende CipherSpec . Wird dieses Zeilengruppenattribut angegeben, muss auch KeyRepository angegeben werden. Wenn KeyRepository angegeben wird, aber CipherSpec weggelassen wird, wird der Standardwert ANY verwendet.

**LocalAddress**
Die lokale Netzschnittstellenadresse, die den Protokollreplikationsverkehr akzeptiert. Wenn dieses Zeilengruppenattribut angegeben wird, gibt es die lokale Netzschnittstelle und/oder den Port im Format "[addr] [(port)]" an. Die Netzadresse kann als Hostname, als IPv4 -Schreibweise mit Trennzeichen oder als IPv6 -Hexadezimalformat angegeben werden. Wenn dieses Attribut weggelassen

wird, versucht der Warteschlangenmanager, eine Bindung zu allen Netzschnittstellen herzustellen. Er verwendet den Port, der in der Zeilengruppe `ReplicationAddress` in der Zeilengruppe `NativeHAInstances` angegeben ist und dem Namen der lokalen Instanz entspricht.

**HeartbeatInterval**
Das Heartbeatintervall legt fest, wie oft in Millisekunden eine aktive Instanz eines Warteschlangenmanagers mit Native HA ein Netzüberwachungssignal sendet. Der gültige Bereich für den Heartbeatintervallwert liegt zwischen 500 (0,5 Sekunden) und 60000 (1 Minute). Ein Wert außerhalb dieses Bereichs führt dazu, dass der Warteschlangenmanager nicht gestartet wird. Wird dieses Attribut nicht angegeben, wird der Standardwert 5000 (5 Sekunden) verwendet. Es muss für alle Instanzen dasselbe Heartbeatintervall festgelegt werden.

**HeartbeatTimeout**
Das Überwachungssignalzeitlimit legt fest, wie lange eine Replikatinstanz eines Warteschlangenmanagers mit Native HA wartet, bevor sie entscheidet, dass die aktive Instanz nicht mehr reagiert. Der gültige Bereich für den Wert dieses Zeitlimits liegt zwischen 500 (0,5 Sekunden) und 120000 (2 Minuten). Der Wert des Überwachungssignalzeitlimits muss größer-gleich dem Wert des Heartbeatintervalls sein.

Ein ungültiger Wert führt dazu, dass der Warteschlangenmanager nicht gestartet wird. Wird dieses Attribut nicht angegeben, wartet ein Replikat 2 x `HeartbeatInterval`, bevor es den Prozess startet, um eine neue aktive Instanz zu wählen. Es muss für alle Instanzen dasselbe Überwachungssignalzeitlimit festgelegt werden.

**RetryInterval**

Das Wiederholungsintervall legt fest, wie oft in Millisekunden ein Warteschlangenmanager mit Native HA eine fehlgeschlagene Replikationsverbindung wiederholen soll. Der gültige Bereich für das Wiederholungsintervall liegt zwischen 500 (0,5 Sekunden) und 120000 (2 Minuten). Wenn dieses Attribut weggelassen wird, wartet ein Replikat 2 x `HeartbeatInterval`, bevor es eine fehlgeschlagene Replikationsverbindung wiederholt.

**SSLFipsRequired**

Gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden, wenn Verschlüsselung beim Senden von Protokollreplikationsdatenverkehr verwendet wird. Setzen Sie den Wert auf `Yes`oder `No`.

**EncryptionPolicySuiteB**

Gibt an, ob der Protokollreplikationsdatenverkehr Suite-B-konforme Verschlüsselung verwendet und welche Stärke verwendet wird. Legen Sie einen der folgenden Werte fest:

**NONE**
Die Suite-B-kompatible Verschlüsselung wird nicht verwendet. Dies ist die Standardeinstellung.

**128_BIT,192_BIT**
Setzt die Sicherheitsstärke sowohl auf 128-Bit-als auch auf 192-Bit-Stufen.

**128_BIT**
Setzt die Sicherheitsstärke auf 128-Bit-Ebene.

**192_BIT**
Setzt die Sicherheitsstärke auf 192-Bit-Ebene.

## Schlüsselrepository-Kennwort verschlüsseln

V 9.3.2

Das Kennwort für das Schlüsselrepository kann mit dem IBM MQ -Kennwortschutzsystem oder einer Stashdatei für das Schlüsselrepository geschützt werden. Weitere Informationen zu diesen beiden Methoden finden Sie unter Schlüsselrepository-Kennwörter verschlüsseln.

Wenn das Repository-Kennwort mit dem Attribut `KeyRepositoryPassword` in der Zeilengruppe `NativeHALocalInstance` angegeben wird, verschlüsseln Sie das Kennwort mit dem Kennwortschutzsystem IBM MQ . Verschlüsseln Sie das Kennwort mit dem Befehl **runmqicred** . Der Befehl gibt das verschlüsselte Kennwort zurück, das im Attribut `KeyRepositoryPassword` angegeben werden kann.

Verwenden Sie einen eindeutigen Anfangsschlüssel, um das Kennwort sicher zu verschlüsseln. Der Name der Datei, die den ursprünglichen Schlüssel enthält, kann mit dem Parameter **-sf** im **runmqicred** -Befehl angegeben werden. Wenn Sie keinen eindeutigen Schlüssel angeben, wird der Standardschlüssel verwendet.

Wenn Sie das Kennwort für das Schlüsselrepository mit einem eindeutigen Anfangsschlüssel verschlüsseln, müssen Sie auch denselben Anfangsschlüssel angeben, indem Sie das Attribut `InitialKeyFile` in der Zeilengruppe `NativeHALocalInstance` verwenden.

### Beispielzeilengruppe

Das folgende Beispiel zeigt die Zeilengruppe `NativeHALocalInstance` , die in der Datei `qm.ini` verwendet wird, um den lokalen Namen eines Knotens anzugeben.

```
NativeHALocalInstance:
  LocalName=node-1
```

**Zugehörige Konzepte**

„Zeilengruppe NativeHAInstance in der Datei qm.ini” auf Seite 145
Für IBM MQ in Containern gibt die Zeilengruppe `NativeHAInstance` an, wie die drei Knoten in einer nativen Hochverfügbarkeitskonfiguration miteinander kommunizieren können.

**Zugehörige Verweise**

runmqicred ( IBM MQ -Clientkennwörter schützen)

## Windows Zeilengruppe 'NETBIOS' in der Datei 'qm.ini' (nur Windows)

Die NETBIOS-Zeilengruppe in der Datei `qm.ini` gibt NetBIOS -Protokollkonfigurationsparameter an. Diese Parameter überschreiben die Standardattribute für Kanäle.

Verwenden Sie die NETBIOS-Zeilengruppe in der Datei `qm.ini` , um NetBIOS -Protokollkonfigurationsparameter anzugeben. Sie überschreiben die Standardattribute für Kanäle.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer Netbios-Warteschlangenmanagers verwenden.

**LocalName=** ***name***
Der Name, unter dem diese Maschine im LAN bekannt ist.

**AdapterNum = 0 (Standardwert) |** ***Adapternummer***
Die Nummer des LAN-Adapters. Der Standardwert ist Adapter 0.

**NumSess = 1 (Standardwert) |** ***Anzahl_Sitzungen***
Die Anzahl der Sitzungen, die zugeordnet werden sollen. Der Standardwert ist 1.

**NumCmds = 1 (Standardwert) |** ***Anzahl_Befehle***
Die Anzahl der Befehle, die zugeordnet werden sollen. Der Standardwert ist 1.

**NumNames = 1 (Standardwert) |** ***number_of_names***
Die Anzahl der zuzuordnende Namen. Der Standardwert ist 1.

**Library1=** ***DLLName1***
Der Name der NetBIOS-DLL.

Der Standardwert ist NETAPI32.

**Zugehörige Konzepte**

„Lokalen NetBIOS-Namen für IBM MQ definieren” auf Seite 280
Der lokale NetBIOS-Name, der von IBM MQ-Kanalprozessen verwendet wird, kann auf drei Arten angegeben werden.

## Linux AIX Zeilengruppe 'RestrictedMode' in der Datei 'qm.ini'

Die Zeilengruppe RestrictedMode gibt den Namen der Gruppe an, die Mitglieder enthält, die berechtigt sind, MQI-Anwendungen auszuführen, alle IPCC-Ressourcen zu aktualisieren und den Inhalt einiger Warteschlangenmanagerverzeichnisse zu ändern. Diese Zeilengruppe gilt nur für AIX and Linux -Systeme.

Die Zeilengruppe 'RestrictedMode' wird durch die Option **-g** im Befehl **crtmqm** festgelegt. Wenn Sie die Option **-g** nicht verwenden, wird die Zeilengruppe nicht in der Datei `qm.ini` erstellt.

Es gibt einige Verzeichnisse im Datenverzeichnis eines Warteschlangenmanagers, in denen IBM MQ-Anwendungen Dateien erstellen, während sie mit dem Warteschlangenmanager verbunden sind. Damit Anwendungen Dateien in diesen Verzeichnissen erstellen können, wird ihnen der World-Write-Zugriff gewährt:

- `/var/mqm/sockets/`*`QMgrName`*`/@ipcc/ssem/`*`hostname`*`/`
- `/var/mqm/sockets/`*`QMgrName`*`/@app/ssem/`*`hostname`*`/`
- `/var/mqm/sockets/`*`QMgrName`*`/zsocketapp/`*`hostname`*`/`

Dabei steht *QMGRNAME* für den Namen des Warteschlangenmanagers und *hostname* für den Hostnamen.

Auf einigen Systemen ist es nicht akzeptabel, allen Benutzern Schreibzugriff auf diese Verzeichnisse zu erteilen. Beispiel: Die Benutzer, die keinen Zugriff auf den Warteschlangenmanager benötigen. Der eingeschränkte Modus ändert die Berechtigungen der Verzeichnisse, in denen WS-Manager-Daten gespeichert werden. Auf die Verzeichnisse kann dann nur von den Mitgliedern der angegebenen Anwendungsgruppe zugegriffen werden. Die Berechtigungen für den gemeinsam genutzten Speicher von System V IPC, der für die Kommunikation mit dem Warteschlangenmanager verwendet wird, werden ebenfalls auf die gleiche Weise geändert.

Die Anwendungsgruppe ist der Name der Gruppe mit Mitgliedern, die über die Berechtigung zum Führen der folgenden Elemente verfügen:

- MQI-Anwendungen ausführen
- Alle IPCC-Ressourcen aktualisieren
- Den Inhalt einiger WS-Manager-Verzeichnisse ändern

Gehen Sie wie folgt vor, um den eingeschränkten Modus für einen WS-Manager

- Der Ersteller des Warteschlangenmanagers muss sich in der Gruppe `mqm` und in der Anwendungsgruppe befinden.
- Die `mqm` -Benutzer-ID muss in der Anwendungsgruppe enthalten sein.
- Alle Benutzer, die den Warteschlangenmanager verwalten möchten, müssen sich in der Gruppe `mqm` und in der Anwendungsgruppe befinden.
- Alle Benutzer, die IBM MQ-Anwendungen ausführen wollen, müssen sich in der Anwendungsgruppe befinden.

Jeder MQCONN-oder MQCONNX-Aufruf, der von einem Benutzer ausgegeben wird, der nicht in der Anwendungsgruppe enthalten ist, schlägt mit dem Ursachencode MQRC_Q_MGR_NOT_AVAILABLE fehl.

**Wichtig:** Auf vielen Betriebssystemen muss sich der betreffende Benutzer abmelden und erneut anmelden, damit das Hinzufügen eines Benutzers zu einer Gruppe erkannt wird.

Der eingeschränkte Modus wird mit dem IBM MQ-Berechtigungsservice betrieben. Daher müssen Sie Benutzern auch die Berechtigung erteilen, eine Verbindung mit IBM MQ herzustellen und auf die Ressourcen zuzugreifen. Dies geschieht mit dem IBM MQ-Berechtigungsservice.

ALW Weitere Informationen zum Konfigurieren des IBM MQ-Berechtigungsservice finden Sie im Abschnitt Sicherheit auf AIX, Linux, and Windows-Systemen einrichten.

Verwenden Sie nur den eingeschränkten Modus von IBM MQ, wenn die vom Berechtigungsservice bereitgestellte Steuerung keine ausreichende Isolation von Warteschlangenmanagerressourcen bietet.

**Zugehörige Verweise**

**crtmqm** (Warteschlangenmanager erstellen)

## Multi Zeilengruppe 'Security' in der Datei 'qm.ini'

Die Zeilengruppe 'Security' gibt Optionen für den Object Authority Manager (OAM) an.

**ClusterQueueAccessControl = RQMName | Xmitq**
Legen Sie dieses Attribut fest, um die Zugriffssteuerung von Clusterwarteschlangen oder vollständig qualifizierten Warteschlangen zu überprüfen, die auf Cluster-WS-Managern gehostet sind.

**RQMName**
Die Profile, die auf die Zugriffssteuerung von fern gehosteten Warteschlangen überprüft werden, sind benannte Warteschlangen oder benannte WS-Manager-Profile.

**XMITQ**
Die Profile, die auf die Zugriffssteuerung von fern gehosteten Warteschlangen überprüft werden, werden in SYSTEM.CLUSTER.TRANSMIT.QUEUE aufgelöst.

Xmitq ist der Standardwert.

Windows **GroupModel=GlobalGroups**
Dieses Attribut legt fest, ob der OAM globale Gruppen prüft, wenn er die Gruppenzugehörigkeit eines Benutzers in Windowsbestimmt.

Standardmäßig werden globale Gruppen nicht überprüft.

**GlobalGroups**
Der OAM überprüft globale Gruppen.

Wenn GlobalGroups festgelegt ist, akzeptieren die Berechtigungsbefehle **setmqaut**, **dspmqaut**und **dmpmqaut** globale Gruppennamen (siehe Parameter **setmqaut** -g ).

**Anmerkung:** Wenn Sie ClusterQueueAcessControl=RQMName festlegen und eine angepasste Implementierung des Berechtigungsservice mit weniger als MQZAS_VERSION_6 ergibt, wird der Warteschlangenmanager nicht gestartet. Geben Sie in dieser Instanz entweder ClusterQueueAcessControl=Xmitq ein oder führen Sie ein Upgrade des angepassten Berechtigungsservice auf MQZAS_VERSION_6 oder höher aus.

### Beispielzeilengruppe

```
Security:
      ClusterQueueAccessControl=Xmitq
      GroupModel=GlobalGroups
```

## Multi Zeilengruppe 'Service' in der Datei 'qm.ini'

Die Zeilengruppe 'Service' wird verwendet, um Änderungen an installierbaren Services vorzunehmen. Diese Zeilengruppe enthält den Namen des Service und die Anzahl der für den Service definierten Eingangspunkte.

**Anmerkung:** Windows Linux Es hat erhebliche Auswirkungen auf die Änderung installierbarer Services und ihrer Komponenten. Aus diesem Grund sind die installierbaren Services in IBM MQ Explorerschreibgeschützt.

Für jede Komponente in einem Service müssen Sie auch den Namen und den Pfad des Moduls angeben, das den Code für diese Komponente enthält. Verwenden Sie dazu die Zeilengruppe ServiceComponent .

Die beiden Zeilengruppen **Service** und **ServiceComponent** können ebenso wie die darin enthaltenen Zeilengruppenschlüssel in beliebiger Reihenfolge aufgeführt werden. Für jede dieser Zeilengruppen müs-

sen alle Zeilengruppenschlüssel vorhanden sein. Falls ein Zeilengruppenschlüssel mehrfach vorkommt, wird der letzte verwendet.

Beim Start verarbeitet der Warteschlangenmanager der Reihe nach alle Servicekomponenteneinträge der Konfigurationsdatei. Anschließend lädt er das angegebene Komponentenmodul, wobei er den Eingangspunkt für die Initialisierung der Komponente aufruft und ihm eine Konfigurationskennung übergibt.

**Name = AuthorizationService (Standardwert) |NameService**
Der Name des erforderlichen Service.

**AuthorizationService**
Für IBM MQwird die Komponente **AuthorizationService** als Objektberechtigungsmanager oder OAM bezeichnet. Die Zeilengruppe **Service** und die zugehörige Zeilengruppe **ServiceComponent** werden automatisch hinzugefügt, wenn der WS-Manager erstellt wird, können aber durch die Umgebungsvariable *MQSNOAUT* überschrieben werden. Fügen Sie andere **ServiceComponent** -Zeilengruppen manuell hinzu.

Linux AIX Die folgenden Zeilengruppen in der Datei `qm.ini` definieren zwei Berechtigungsservicekomponenten unter IBM MQ for AIX. *MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

```
Service:
   Name=AuthorizationService
   EntryPoints=13

ServiceComponent:
   Service=AuthorizationService
   Name=MQSeries.UNIX.auth.service
Module=MQ_INSTALLATION_PATH/lib/amqzfu
   ComponentDataSize=0

ServiceComponent:
   Service=AuthorizationService
   Name=user.defined.authorization.service
   Module=/usr/bin/udas01
   ComponentDataSize=96
```

Linux AIX Die Zeilengruppe ServiceComponent `MQSeries.UNIX.auth.service` definiert die Standardberechtigungsservicekomponente, den OAM. Wenn Sie diese Zeilengruppe entfernen und den Warteschlangenmanager erneut starten, wird der Objektberechtigungsmanager inaktiviert und es werden keine Berechtigungsprüfungen vorgenommen.

Windows Sie können das Attribut **SecurityPolicy** auch mithilfe der IBM MQ -Services hinzufügen. Das Attribut **SecurityPolicy** gilt nur, wenn der in der Zeilengruppe 'Service' angegebene Service der Berechtigungsservice ist, d. h. der Standard-OAM. Mit dem Attribut **SecurityPolicy** können Sie für jeden Warteschlangenmanager die Sicherheitsrichtlinie angeben. Folgende Werte sind möglich:

**Standard**
Geben Sie `Default` an, wenn die Standardsicherheitsrichtlinie gelten soll. Wenn dem OAM für eine bestimmte Benutzer-ID keine Windows-Sicherheitskennung (NT SID) übergeben wird, wird die passende SID in den relevanten Sicherheitsdatenbanken gesucht.

**NTSIDsRequired**
Verlangt bei Sicherheitsprüfungen, dass dem OAM eine NT SID übergeben wird.

Windows Die Zeilengruppe ServiceComponent `MQSeries.WindowsNT.auth.service` definiert die Standardberechtigungsservicekomponente, den OAM. Wenn Sie diese Zeilengruppe entfernen und den Warteschlangenmanager erneut starten, wird der Objektberechtigungsmanager inaktiviert und es werden keine Berechtigungsprüfungen vorgenommen.

**NameService**
Standardmäßig wird kein Namensservice bereitgestellt. Wenn Sie einen Namensservice benötigen, müssen Sie die Zeilengruppe NameService manuell hinzufügen.

Linux AIX Die folgenden AIX and Linux `qm.ini` -Dateizeilengruppen für den Namensservice geben eine Namensservicekomponente an, die vom (fiktiven) Unternehmen ABC bereitgestellt wird.

```
# Stanza for name service
 Service:
    Name=NameService
    EntryPoints=5

# Stanza for name service component, provided by ABC
 ServiceComponent:
    Service=NameService
    Name=ABC.Name.Service
    Module=/usr/lib/abcname
    ComponentDataSize=1024
```

**Anmerkung:** Windows Auf Windows -Systemen werden Informationen zur Zeilengruppe NameService in der Registry gespeichert.

**EntryPoints=** ***number-of-entries***
Die Anzahl der Eingangspunkte, die für den Service definiert wurden.

Dazu gehören die Initialisierungs-und Beendigungseingangspunkte

Windows **SecurityPolicy= Standard | NTSIDsRequired**
Auf Windows-Systemen gilt das Attribut **SecurityPolicy** nur, wenn der angegebene Service der Standardberechtigungsservice, d. h. der OAM ist. Mit dem Attribut **SecurityPolicy** können Sie für jeden Warteschlangenmanager die Sicherheitsrichtlinie angeben.

Folgende Werte sind möglich:

**Standard**
Verwenden Sie die Standardsicherheitsrichtlinie, die wirksam werden soll. Wenn dem OAM für eine bestimmte Benutzer-ID keine Windows-Sicherheitskennung (NT SID) übergeben wird, wird die passende SID in den relevanten Sicherheitsdatenbanken gesucht.

**NTSIDsRequired**
Übergeben Sie eine NT-SID an den OAM, wenn Sie Sicherheitsprüfungen durchführen.

Weitere Informationen finden Sie unter Windows -Sicherheitskennungen (SIDs).

Siehe auch Zeilengruppen für Berechtigungsservice konfigurieren: Windows-Systeme.

Linux AIX **SecurityPolicy=Benutzer|Gruppe|UserExternal|default**
Auf AIX and Linux -Systemen gibt der Wert an, ob der WS-Manager benutzer-oder gruppenbasierte Berechtigung verwendet. Bei Werten wird die Groß-/Kleinschreibung nicht beachtet.

Dieser kann einen der folgenden Werte annehmen:

**Gruppe**
Der Warteschlangenmanager verwendet die gruppenbasierte Berechtigung. Einer Gruppe wird die Berechtigung für den Zugriff auf eine Ressource erteilt.

Ein Benutzer erhält die Zusammenfassung aller Berechtigungen, die jeder Gruppe erteilt werden, zu der er gehört.

Benutzer-IDs und Gruppen müssen für das lokale Betriebssystem definiert werden.

**Benutzer**
Der Warteschlangenmanager verwendet die benutzerbasierte Berechtigung. Die Berechtigung für den Zugriff auf eine Ressource kann einer Gruppe oder einer bestimmten Benutzer-ID erteilt werden.

Ein Benutzer erhält die Zusammenfassung der folgenden Berechtigungen:

- Berechtigungen, die dem jeweiligen Benutzer erteilt werden
- Berechtigungen, die jeder Gruppe erteilt werden, zu der der Benutzer gehört.

Benutzer-IDs und Gruppen müssen für das lokale Betriebssystem definiert werden.

V 9.3.0 **UserExternal**
Der Warteschlangenmanager verwendet die benutzerbasierte Berechtigung. Berechtigungen können jedoch Benutzer-IDs erteilt werden, die dem lokalen Betriebssystem nicht bekannt sind.

Die Berechtigung für den Zugriff auf eine Ressource kann einer Gruppe oder einer bestimmten Benutzer-ID erteilt werden.

Ein Benutzer erhält die Zusammenfassung der folgenden Berechtigungen:

- Berechtigungen, die dem jeweiligen Benutzer erteilt werden
- Berechtigungen, die jeder Gruppe erteilt werden, zu der der Benutzer gehört.

Wenn ein Benutzer dem lokalen Betriebssystem nicht bekannt ist, wird davon ausgegangen, dass er nur zur Gruppe 'nobody' gehört. Weitere Informationen zu Gruppen finden Sie im Abschnitt Prinzipals und Gruppen unter AIX, Linux, and Windows. Die Benutzer-ID muss bis zu 12 Zeichen lang sein und den Regeln für die Benennung von IBM MQ -Objektenentsprechen.

Sie können vorhandene Warteschlangenmanager so ändern, dass diese zusätzliche Option verwendet wird, ohne eine aktuelle Konfiguration zu verlieren.

V 9.3.4 Dies ist der Standardwert, wenn die Zeilengruppe AuthToken angegeben wird.

**Standard**
Der Warteschlangenmanager verwendet die gruppenbasierte Berechtigung. Das Verhalten entspricht dem für die Option `group` .

Dies ist der Standardwert, wenn die Zeilengruppe AuthToken nicht angegeben wird.

Starten Sie den Warteschlangenmanager erneut, damit Änderungen am Attributwert wirksam werden.

**Anmerkung:** V 9.3.4 Linux AIX Ab IBM MQ 9.3.4wird bei Angabe der Zeilengruppe AuthToken der effektive Wert des Attributs **SecurityPolicy** der Zeilengruppe 'Service' auf `UserExternal`gesetzt. Die Tokenauthentifizierung ist nicht verfügbar, wenn **SecurityPolicy** explizit auf `Group` in der Zeilengruppe 'Service' gesetzt ist. Wenn **SecurityPolicy** auf `Group`gesetzt ist, entfernen Sie das Attribut **SecurityPolicy** aus der Zeilengruppe 'Service' und starten Sie den Warteschlangenmanager erneut. Weitere Informationen finden Sie im Abschnitt „Zeilengruppe AuthToken in der Datei qm.ini” auf Seite 120.

**SharedBindingsUserId= *user-type***
Das Attribut **SharedBindingsUserId** gilt nur, wenn der angegebene Service der Standardberechtigungsservice ist, d. h. der OAM. Das Attribut **SharedBindingsUserId** wird nur für gemeinsam genutzte Bindungen verwendet. Mit diesem Wert können Sie angeben, ob das Feld *UserIdentifier* in der *IdentityContext*-Struktur aus der Funktion MQZ_AUTHENTICATE_USER die effektive Benutzer-ID oder die reale Benutzer-ID ist.

Informationen zur Funktion MQZ_AUTHENTICATE_USER finden Sie im Abschnitt MQZ_AUTHENTICATE_USER-Authentifizierungs-Benutzer .

Folgende Werte sind möglich:

**Standard**
Der Wert für das Feld *UserIdentifier* wird als reale Benutzer-ID festgelegt.

**Real**
Der Wert für das Feld *UserIdentifier* wird als reale Benutzer-ID festgelegt.

**Effektiv**
Der Wert des Feldes *UserIdentifier* wird als effektive Benutzer-ID festgelegt.

**FastpathBindingsUserId= *user-type***
Das Attribut **FastpathBindingsUserId** gilt nur, wenn der angegebene Service der Standardberechtigungsservice ist, d. h. der OAM. Das Attribut **FastpathBindingsUserId** wird nur mit Bezug auf Fastpath-Bindungen verwendet. Mit diesem Wert können Sie angeben, ob das Feld *UserIdentifier* in der *IdentityContext*-Struktur aus der Funktion MQZ_AUTHENTICATE_USER die effektive Benutzer-ID oder die reale Benutzer-ID ist.

Informationen zur Funktion MQZ_AUTHENTICATE_USER finden Sie im Abschnitt MQZ_AUTHENTICATE_USER-Authentifizierungs-Benutzer .

Folgende Werte sind möglich:

**Standard**
Der Wert für das Feld *UserIdentifier* wird als reale Benutzer-ID festgelegt.

**Real**
Der Wert für das Feld *UserIdentifier* wird als reale Benutzer-ID festgelegt.

**Effektiv**
Der Wert des Feldes *UserIdentifier* wird als effektive Benutzer-ID festgelegt.

**IsolatedBindingsUserId= *user-type***
Das Attribut **IsolatedBindingsUserId** gilt nur, wenn der angegebene Service der Standardberechtigungsservice ist, d. h. der OAM. Das Attribut **IsolatedBindingsUserId** wird nur für isolierte Bindungen verwendet. Mit diesem Wert können Sie angeben, ob das Feld *UserIdentifier* in der *IdentityContext*-Struktur aus der Funktion MQZ_AUTHENTICATE_USER die effektive Benutzer-ID oder die reale Benutzer-ID ist.

Informationen zur Funktion MQZ_AUTHENTICATE_USER finden Sie im Abschnitt MQZ_AUTHENTICATE_USER-Authentifizierungs-Benutzer .

Folgende Werte sind möglich:

**Standard**
Der Wert des Feldes *UserIdentifier* wird als effektive Benutzer-ID festgelegt.

**Real**
Der Wert für das Feld *UserIdentifier* wird als reale Benutzer-ID festgelegt.

**Effektiv**
Der Wert des Feldes *UserIdentifier* wird als effektive Benutzer-ID festgelegt.

Weitere Informationen zu installierbaren Services und Komponenten finden Sie im Abschnitt Installierbare Services und Komponenten für AIX, Linux, and Windows.

Weitere Informationen zu Sicherheitsservices im Allgemeinen finden Sie im Abschnitt Sicherheit auf AIX and Linux-Systemen konfigurieren.

## Beispielzeilengruppe

```
Service:
   Name=AuthorizationService
   EntryPoints=14
```

**Zugehörige Konzepte**
Installierbare Services und Komponenten für AIX, Linuxund Windows
**Zugehörige Verweise**
Installierbare Services und Komponenten unter IBM i
Referenzinformationen zu installierbaren Services

## Multi Zeilengruppe ServiceComponent in der Datei qm.ini

Die Zeilengruppe ServiceComponent gibt Informationen für die Servicekomponente an. Sie müssen Servicekomponentinformationen angeben, wenn Sie einen neuen installierbaren Service hinzufügen. Die Berechtigungsservice-Zeilengruppe ist standardmäßig vorhanden, und die zugeordnete Komponente (OAM) ist aktiv.

Die beiden Zeilengruppen **Service** und **ServiceComponent** können ebenso wie die darin enthaltenen Zeilengruppenschlüssel in beliebiger Reihenfolge aufgeführt werden. Für jede dieser Zeilengruppen müs-

sen alle Zeilengruppenschlüssel vorhanden sein. Falls ein Zeilengruppenschlüssel mehrfach vorkommt, wird der letzte verwendet.

Beim Start verarbeitet der Warteschlangenmanager der Reihe nach alle Servicekomponenteneinträge der Konfigurationsdatei. Anschließend lädt er das angegebene Komponentenmodul, wobei er den Eingangspunkt für die Initialisierung der Komponente aufruft und ihm eine Konfigurationskennung übergibt.

**Service = *service_name***
Der Name des erforderlichen Service. Dieser Wert muss mit dem Wert übereinstimmen, der im Attribut Name der Servicekonfigurationsinformationen angegeben ist.

**Name = *component_name***
Der beschreibende Name der Servicekomponente. Dieser Wert muss eindeutig sein und darf nur Zeichen enthalten, die für die Namen von IBM MQ-Objekten gültig sind (z. B. Warteschlangennamen). Dieser Name tritt in Bedienernachrichten auf, die durch den Service generiert werden. Es wird empfohlen, diesen Namen mit einer Unternehmensmarke oder einer ähnlichen Unterscheidungszeichenfolge zu beginnen.

**Modul = *module_name***
Der Name des Moduls, das den Code für diese Komponente enthält. Dies muss ein vollständiger Pfadname sein.

**ComponentDataSize= *size***
Die Größe des Komponentendatenbereichs (in Byte), der an die Komponente in jedem Aufruf übergeben wurde. Geben Sie Null an, wenn keine Komponentendaten erforderlich sind.

## Beispielzeilengruppe

```
ServiceComponent:
   Service=AuthorizationService
   Name=MQSeries.UNIX.auth.service
   Module=amqzfu
   ComponentDataSize=0
```

Weitere Beispiele mit einer Zeilengruppe AuthorizationService und den zugehörigen Zeilengruppen ServiceComponent sowie einer Zeilengruppe NameService und der zugehörigen Zeilengruppe ServiceComponent finden Sie in „Zeilengruppe 'Service' in der Datei 'qm.ini'" auf Seite 150.

**Zugehörige Konzepte**

Installierbare Services und Komponenten für AIX, Linuxund Windows

**Zugehörige Verweise**

„Zeilengruppe 'Service' in der Datei 'qm.ini'" auf Seite 150
Die Zeilengruppe 'Service' wird verwendet, um Änderungen an installierbaren Services vorzunehmen. Diese Zeilengruppe enthält den Namen des Service und die Anzahl der für den Service definierten Eingangspunkte.

Installierbare Services und Komponenten unter IBM i

Referenzinformationen zu installierbaren Services

## Windows Zeilengruppe 'SPX' in der Datei 'qm.ini' (nur Windows)

Die SPX-Zeilengruppe gibt SPX-Protokollkonfigurationsparameter an. Diese Parameter überschreiben die Standardattribute für Kanäle.

Verwenden Sie die SPX-Zeilengruppe in der Datei `qm.ini` , um SPX-Protokollkonfigurationsparameter anzugeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer SPX -Warteschlangenmanagers verwenden.

**Socket = 5E86 (Standardwert) | *socketnummer***
Die SPX-Socket-Nummer in Hexadezimalschreibweise. Der Standardwert ist X'5E86 '.

**BoardNum = 0 (Standardwert) | *adapternummer***
Die LAN-Adapternummer. Der Standardwert ist Adapter 0.

**KeepAlive = NO | YES**
Schalten Sie die KeepAlive-Funktion ein oder aus.

KeepAlive=YES bewirkt, dass SPX in regelmäßigen Abständen überprüft, ob das andere Ende der Verbindung noch verfügbar ist. Ist dies nicht der Fall, wird der Kanal geschlossen.

**Library1= *DLLName1***
Der Name der SPX-DLL.

Der Standardwert ist WSOCK32.DLL.

**Library2= *DLLName2***
Dasselbe gilt für LibraryName1, das verwendet wird, wenn der Code in zwei separaten Bibliotheken gespeichert wird.

Der Standardwert ist WSOCK32.DLL.

**ListenerBacklog=number**
Überschreiben Sie die Standardanzahl ausstehender Anforderungen für den SPX-Listener.

Beim Empfang auf SPX wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Dies kann als ein Rückstand von Anforderungen betrachtet werden, die auf den SPX-Socket warten, damit der Listener die Anforderung akzeptiert. Die Standardwerte für das Listener-Rückstandsprotokoll werden in Tabelle 13 auf Seite 156 angezeigt.

*Tabelle 13. Ausstehende Standardverbindungsanforderungen (SPX)*

| **Plattform** | **Standardwert für ListenerBacklog** |
|---|---|
| Windows-Server | 100 |
| Windows-Workstation | 5 |

**Anmerkung:** Einige Betriebssysteme unterstützen einen größeren Wert als der angezeigte Standardwert. Verwenden Sie diese Option, um das Erreichen der Verbindungsgrenze zu vermeiden.

Umgekehrt können einige Betriebssysteme die Größe des SPX-Rückprotokolls begrenzen, so dass der effektive SPX-Rückstand kleiner als hier angefordert werden kann.

Wenn der Rückstand die in Tabelle 13 auf Seite 156 angezeigten Werte erreicht, wird die SPX-Verbindung abgelehnt und der Kanal kann nicht starten. Bei Nachrichtenkanälen führt dies dazu, dass der Kanal in einen RETRY-Status eingeht und die Verbindung zu einem späteren Zeitpunkt erneut versucht. Für Clientverbindungen empfängt der Client einen Ursachencode MQRC_Q_MGR_NOT_AVAILABLE von `MQCONN` und sollte die Verbindung zu einem späteren Zeitpunkt wiederholen.

## Multi Zeilengruppe 'SSL' in der Datei 'qm.ini'

Die SSL-Zeilengruppe wird verwendet, um die TLS-Kanäle in einem Warteschlangenmanager zu konfigurieren.

### OCSP (Online Certificate Status Protocol)

Ein Zertifikat kann die Erweiterung "AuthorityInfoAccess" enthalten. Diese Erweiterung gibt einen Server an, der über das Online Certificate Status Protocol (OCSP) kontaktiert werden soll. Um SSL-oder TLS-Kanäle in Ihrem Warteschlangenmanager zu ermöglichen, die Erweiterungen von AuthorityInfoAccess zu verwenden, stellen Sie sicher, dass der in ihnen benannte OCSP-Server verfügbar ist, ordnungsgemäß konfiguriert ist und über das Netz zugänglich ist. Weitere Informationen hierzu finden Sie im Abschnitt Mit widerrufenden Zertifikaten arbeiten .

## CrlDistributionPoint (CDP)

Ein Zertifikat kann eine CrlDistributionPoint-Erweiterung enthalten. Diese Erweiterung enthält eine URL, die sowohl das Protokoll, das zum Download einer Zertifikatswiderrufsliste (CRL) verwendet wird, als auch den zu kontaktierbaren Server enthält.

Wenn Sie SSL-oder TLS-Kanäle in Ihrem Warteschlangenmanager für die Verwendung von "CrlDistributionPoint" -Erweiterungen zulassen möchten, stellen Sie sicher, dass der in ihnen benannte CDP-Server verfügbar und korrekt konfiguriert ist und über das Netz zugänglich ist.

## Die SSL-Zeilengruppe

Verwenden Sie die SSL-Zeilengruppe in der Datei `qm.ini`, um zu konfigurieren, wie TLS-Kanäle auf Ihrem Warteschlangenmanager versuchen, die folgenden Funktionen zu verwenden, und wie sie reagieren, wenn Probleme auftreten, wenn sie verwendet werden.

Wenn der angegebene Wert nicht einer der gültigen Werte ist, wird in jedem der folgenden Fälle der Standardwert übernommen. Es werden keine Fehlernachrichten geschrieben, die darauf hingewiesen werden, dass ein ungültiger Wert angegeben wurde.

V 9.3.0 **OutboundSNI = CHANNEL | HOSTNAME**

Wenn **OutboundSNI** auf KANALgesetzt ist, setzen SNI-fähige Clients SNI auf den Namen des IBM MQ -Zielkanals des fernen Systems, wenn eine TLS-Verbindung eingeleitet wird.

Wenn dieses Attribut auf HOSTNAME gesetzt ist, legen SNI-fähige Clients als SNI-Header den Hostnamen fest, wodurch ausgehende Verbindungsanforderungen das Standardzertifikat des fernen Warteschlangenmanagers während des TLS-Handshakes empfangen und somit keine kanalweisen Zertifikate verwendet werden können.

**Anmerkung:** Wenn **OutboundSNI**=HOSTNAME verwendet wird, um eine Verbindung zu einem fernen Kanal mit einer konfigurierten Zertifikatsbezeichnung herzustellen, wird die Verbindung mit dem Fehler MQRC_SSL_INITIALIZATION_ERROR zurückgewiesen und eine AMQ9673 -Nachricht in den Fehlerprotokollen des fernen Warteschlangenmanagers ausgegeben.

**AllowOutboundSNI = YES (Standardwert) | NEIN**

Wenn diese Option aktiviert ist, legen SNI-fähige Clients als SNI den Namen des IBM MQ-Zielkanals für das ferne System fest, wenn eine TLS-Verbindung eingeleitet wird. Wenn dieses Attribut auf NO gesetzt ist, legen SNI-fähige Clients nicht den SNI-Header fest, wodurch ausgehende Verbindungsanforderungen das Standardzertifikat des fernen Warteschlangenmanagers während dem TLS-Handshake empfangen und somit keine kanalweisen Zertifikate verwendet werden können.

**Achtung:** V 9.3.0 Deprecated Ab IBM MQ 9.3.0 ist die Eigenschaft **AllowOutboundSNI** veraltet und nur zu Zwecken der Abwärtskompatibilität verfügbar.

**AllowOutboundSNI** set to JA bietet dieselbe Funktion wie **OutboundSNI** set to KANAL, während **AllowOutboundSNI** set to NEIN dieselbe Funktion wie **OutboundSNI** set to HOSTNAME bereitstellt.

Wenn die SSL-Zeilengruppe sowohl das Attribut **AllowOutboundSNI** als auch das Attribut **OutboundSNI** enthält, hat die Einstellung von **OutboundSNI** Vorrang.

**AllowedCipherSpecs=*Name*|*Namensliste*|ALL**

Gibt eine benutzerdefinierte Liste der CipherSpecs an, die für die Verwendung mit IBM MQ-Kanälen auf Multiplatforms angeordnet und aktiviert sind.

- Den Namen einer einzelnen CipherSpec.
- Eine durch Kommas getrennte Liste der Namen von IBM MQ-CipherSpecs, die wieder aktiviert werden können.
- Der Sonderwert ALL, der alle CipherSpecs darstellt (nicht empfohlen).

**Anmerkung:** Sie sollten **ALL** CipherSpecs nicht auswählen, weil dadurch SSL 3.0-und TLS 1.0-Protokolle und eine große Anzahl schwacher Verschlüsselungsalgorithmen aktiviert werden.

Weitere Informationen finden Sie unter Benutzerdefinierte Liste mit bestellten und aktivierten CipherSpecs unter IBM MQ for Multiplatforms in der CipherSpec -Reihenfolge beim TLS-Handshake bereitstellen.

IBM i ALW **AllowTLSV13 =Y | YES | T | TRUE (Standardwert) | N | NO | F | FALSE**

Gibt an, ob ein Warteschlangenmanager die TLS 1.3-CipherSpecs verwenden kann.

- Y (Standardwert), YES (Standardwert) T (Standardwert) oder TRUE (Standardwert): Aktiviert TLS 1.3 , damit der Warteschlangenmanager die TLS 1.3 CipherSpecsverwenden kann.
- N, NO, F oder FALSE: Inaktiviert TLS 1.3, was bedeutet, dass der Warteschlangenmanager die TLS 1.3-CipherSpecs nicht verwenden kann.

Weitere Informationen finden Sie unter CipherSpecs aktivieren.

**CDPCheckExtensions= YES |NO (Standardwert)**

Gibt an, ob TLS-Kanäle in diesem Warteschlangenmanager versuchen, CDP-Server zu überprüfen, die in den Zertifikatserweiterungen des CrlDistributionPoint-Zertifikats benannt sind.

- YES : TLS-Kanäle versuchen, die CDP-Server zu überprüfen, um festzustellen, ob ein digitales Zertifikat widerrufen wird.
- NO (Standardwert): TLS-Kanäle versuchen nicht, CDP-Server zu überprüfen. Dieser Wert stellt den Standardwert dar.

ALW **MinimumRSAKeySize=int**

Gibt die Mindestschlüsselgröße an, die RSA-Zertifikate haben müssen, damit sie während des TLS-Handshake akzeptiert werden. Jeder Wert größer-gleich 0 ist zulässig. Falls nichts angegeben ist, wird der Standardwert 1 verwendet.

**OCSPAuthentication=REQUIRED (Standardwert) | WARN | OPTIONAL**

Gibt die Aktion an, die ausgeführt werden soll, wenn ein Widerrufsstatus nicht von einem OCSP-Server bestimmt werden kann.

Wenn die OCSP-Prüfung aktiviert ist, versucht ein TLS-Kanalprogramm, einen OCSP-Server zu kontaktieren.

Wenn das Kanalprogramm keine OCSP-Server kontaktieren kann oder wenn kein Server den Widerrufsstatus des Zertifikats bereitstellen kann, wird der Wert des Parameters OCSPAuthentication verwendet.

- ERFORDERLICH (Standardwert): Wenn der Widerrufsstatus nicht ermittelt werden kann, wird die Verbindung mit einem Fehler geschlossen. Dieser Wert stellt den Standardwert dar.
- WARNUNG : Wenn der Widerrufsstatus nicht bestimmt wird, wird eine Warnung in das Fehlerprotokoll des Warteschlangenmanagers geschrieben, aber die Verbindung kann fortgesetzt werden.
- OPTIONAL : Wenn der Widerrufsstatus nicht festgestellt werden kann, kann die Verbindung unbeaufsichtigt fortgesetzt werden. Es werden keine Warnungen oder Fehler ausgegeben.

**OCSPCheckExtensions = YES (Standardwert) | NEIN**

Gibt an, ob TLS-Kanäle in diesem Warteschlangenmanager versuchen, OCSP-Server zu überprüfen, die in den Erweiterungen des Zertifikats 'AuthorityInfoAccess' angegeben sind.

- YES (Standardwert): TLS-Kanäle versuchen, OCSP-Server zu überprüfen, um festzustellen, ob ein digitales Zertifikat widerrufen wird. Dieser Wert stellt den Standardwert dar.
- NO : TLS-Kanäle versuchen nicht, die OCSP-Server zu überprüfen.

ALW **OCSPTimeout= *Anzahl***

Die Anzahl der Sekunden, die bei der Ausführung einer Widerrufsprüfung auf einen OCSP-Responder gewartet wird.

Ab IBM MQ 9.3.0wird das Standardzeitlimit von 30 Sekunden verwendet, wenn der Wert 0 festgelegt ist.

Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert von 30 Sekunden verwendet.

**SSLHTTPProxyName=** ***string***
Die Zeichenfolge ist entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers, der von IBM Global Security Kit (GSKit) für OCSP-Prüfungen verwendet wird. Auf die Adresse kann optional eine in Klammern gesetzte Portnummer folgen. Wenn Sie die Portnummer nicht angeben, wird der Standard-HTTP-Port 80 verwendet.

AIX Für 32-Bit-Clients unter AIX kann die Netzadresse nur eine IPv4- Adresse sein.

Auf anderen Plattformen kann die Netzadresse eine IPv4 oder IPv6-Adresse sein.

Dieses Attribut kann erforderlich sein, wenn z. B. eine Firewall den Zugriff auf die URL des OCSP-Responder verhindert.

ALW **PeerCertChainValidation=*Zeichenfolge***
Die Zeichenfolge kann einen der beiden folgenden Werte haben:

- `Usepeerchain` **[Standardwert]**: Mit der vom Peer bereitgestellten Zertifikatskette können bei der Validierung von Zertifikaten Lücken in der Vertrauenskette überbrückt werden. Eine Ausnahme ist das Stammzertifikat.
- `Truststoreonly` **[Nicht empfohlen]**: Nur Zertifikate im Truststore werden für die Validierung des Peerzertifikats verwendet.

ALW **SSLHTTPConnectTimeout=** ***Anzahl*****|0**
Die Anzahl der Sekunde, die beim Ausführen einer Widerrufsprüfung auf die erfolgreiche Herstellung einer Netzverbindung zu einem HTTP-Server gewartet wird.

Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert 0 (aus) verwendet.

## Beispielzeilengruppe

```
SSL:
       OutboundSNI=CHANNEL
       AllowedCipherSpecs=TLS13 CipherSpec list
       AllowTLSV13=Y
       CDPCheckExtensions=NO
       MinimumRSAKeySize=1
       OCSPAuthentication=REQUIRED
       OCSPCheckExtensions=YES
       OCSPTimeout=30
       PeerCertChainValidation=Usepeerchain
       SSLHTTPConnectTimeout=0
```

**Anmerkungen:**

- Der Standardwert für **OutboundSNI** ist **Channel**.
- Die Liste **TLS13 CipherSpec** ist eine Liste bestimmter CipherSpecs , nicht die Aliasverschlüsselungen. Wenn Sie nur TLS1.3 -Verschlüsselungen benötigen, müssen Sie sie auflisten. For example:

  TLS_CHACHA20_POLY1305_SHA256
  TLS_AES_256_GCM_SHA384
  TLS_AES_128_GCM_SHA256
  TLS_AES_128_CCM_SHA256
  TLS_AES_128_CCM_8_SHA256
- Der Standardwert für **AllowTLSV13** ist Y , es sei denn, Sie haben schwache Verschlüsselungen aktiviert. In diesem Fall wird die Verschlüsselung inaktiviert (es sei denn, Sie haben sie explizit aktiviert.
- Die Werte für **CDPCheckExtensions** können nur `Ja` oder `Nein`sein.

- Die Werte für **PeerCertChainValidation** können nur Usepeerchain oder Truststoreonlysein.

## Multi Zeilengruppe 'Subpool' in de Datei 'qm.ini'

Diese Zeilengruppe wird von IBM MQ erstellt. Ändern Sie sie nicht.

Die Zeilengruppe 'Subpool' und das Attribut **ShortSubpoolName** in dieser Zeilengruppe werden automatisch von IBM MQ geschrieben, wenn Sie einen Warteschlangenmanager erstellen. IBM MQ wählt einen Wert für **ShortSubpoolName** aus. Ändern Sie diesen Wert nicht.

Der Name entspricht einem Verzeichnis und einem symbolischen Link, der im /var/mqm/sockets -Verzeichnis erstellt wird, das IBM MQ für die interne Kommunikation zwischen den aktiven Prozessen verwendet.

## Multi Zeilengruppe 'TCP' in der Datei 'qm.ini'

Die TCP-Zeilengruppe gibt TCP/IP-Konfigurationsparameter (Transmission Control Protocol/Internet Protocol ) an. Diese Parameter überschreiben die Standardattribute für Kanäle.

Mithilfe der TCP-Zeilengruppe in der Datei qm.ini können Sie TCP/IP-Konfigurationsparameter angeben.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie Eigenschaftenseite des IBM MQ Explorer SPX -TCP-Warteschlangenmanagers verwenden.

**Port = 1414 (Standardwert) | *portnummer***
Die Standardportnummer (in Dezimalschreibweise) für TCP/IP-Sitzungen. Die *Standard*portnummer für IBM MQ ist 1414.

Windows **Library1= *DLLName1* (nur Windows)**
Der Name der TCP/IP-Sockets-DLL.

Der Standardwert ist WSOCK32.

Multi V 9.3.0 **SecureCommsOnly = NO | N | FALSE | F (Standardwert) |TRUE | T | YES | Y**
Geben Sie an, ob eine einfache Textkommunikation zulässig ist oder nicht.

**SecureCommsOnly=NO|N|FALSE|F**
Eine einfache Textkommunikation ist zulässig, und eine Warnung wird ausgegeben, wenn der Warteschlangenmanager gestartet wird.

**SecureCommsOnly=YES|Y|TRUE|T**
Eine einfache Textkommunikation ist nicht zulässig, und eine Informationsnachricht wird ausgegeben, wenn der Warteschlangenmanager gestartet wird.

**KeepAlive = NO (Standardwert) |JA**
Schalten Sie die KeepAlive-Funktion ein oder aus. KeepAlive=YES bewirkt, dass TCP/IP in regelmäßigen Abständen überprüft, ob das andere Ende der Verbindung noch verfügbar ist. Ist dies nicht der Fall, wird der Kanal geschlossen.

**ListenerBacklog=number**
Überschreiben Sie die Standardanzahl ausstehender Anforderungen für den TCP/IP-Listener.

Beim Empfang über TCP/IP wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Dies kann als ein Rückstand von Anforderungen betrachtet werden, die auf den TCP/IP-Port warten, bis der Listener die Anforderung akzeptiert. Die Standardwerte für das Listener-Rückstandsprotokoll werden in Tabelle 14 auf Seite 160 angezeigt.

*Tabelle 14. Ausstehende Standardverbindungsanforderungen (TCP)*

| Plattform | Standardwert für ListenerBacklog |
|---|---|
| Windows Windows -Server | 100 |

| *Tabelle 14. Ausstehende Standardverbindungsanforderungen (TCP) (Forts.)* | |
|---|---|
| **Plattform** | **Standardwert für ListenerBacklog** |
| Linux Linux | 100 |
| AIX AIX V5.3 oder höher | 100 |

**Anmerkung:** Einige Betriebssysteme unterstützen einen größeren Wert als der angezeigte Standardwert. Verwenden Sie diese Option, um das Erreichen der Verbindungsgrenze zu vermeiden.

Umgekehrt können einige Betriebssysteme die Größe des TCP-Rückstageprotokolls begrenzen, so dass der effektive TCP-Rückstand kleiner als hier angefordert werden kann.

Wenn der Rückstand die in Tabelle 14 auf Seite 160 angezeigten Werte erreicht, wird die TCP/IP-Verbindung abgelehnt und der Kanal kann nicht starten. Bei Nachrichtenkanälen führt dies dazu, dass der Kanal in einen RETRY-Status eingeht und die Verbindung zu einem späteren Zeitpunkt erneut versucht. Für Clientverbindungen empfängt der Client einen Ursachencode MQRC_Q_MGR_NOT_AVAILABLE von `MQCONN` und wiederholt die Verbindung zu einem späteren Zeitpunkt.

Die folgende Gruppe von Eigenschaften kann verwendet werden, um die Größe der Puffer zu steuern, die von TCP/IP verwendet werden. Die Werte werden direkt an die TCP/IP-Schicht des Betriebssystems übergeben. Bei der Verwendung dieser Eigenschaften ist eine große Sorgfalt zu berücksichtigen. Wenn die Werte falsch eingestellt sind, kann dies die TCP/IP-Leistung beeinträchtigen. Weitere Informationen dazu, wie sich diese Auswirkung auf die Leistung auswirkt, finden Sie in der TCP/IP-Dokumentation für Ihre Umgebung. Der Wert 0 gibt an, dass die Puffergrößen vom Betriebssystem verwaltet werden, im Gegensatz zu Festlegung der Puffergrößen durch IBM MQ.

**Connect_Timeout = `0` (Standardwert) |number**
Die Anzahl der Sekunden, bevor ein Versuch unternommen wird, das Socket-Zeitlimit zu verbinden. Der Standardwert 0 gibt an, dass kein Verbindungszeitlimit vorhanden ist.

IBM MQ-Kanalprozesse stellen Verbindungen über nicht blockierende Sockets her. Wenn das andere Ende des Sockets daher nicht bereit ist, wird die Verbindung () sofort mit *EINPROGRESS* oder *EWOULDBLOCK* zurückgegeben. Danach wird erneut versucht, eine Verbindung herzustellen, bis zu insgesamt 20 solcher Versuche, wenn ein Übertragungsfehler gemeldet wird.

Wenn `Connect_Timeout` auf einen Wert ungleich null gesetzt ist, wartet IBM MQ für den angegebenen Zeitraum auf select(), um den Socket bereit zu machen. Dies erhöht die Erfolgschancen eines nachfolgenden connect () -Aufrufs. Diese Option kann in Situationen nützlich sein, in denen eine Verbindung aufgrund einer hohen Auslastung im Netz einige Wartezeiten erfordern würde.

**SndBuffSize = Zahl |`0` (Standardwert)**
Die Größe des TCP/IP-Sendepuffers in Byte, der vom sendenden Ende der Kanäle verwendet wird. Dieser Zeilengruppenwert kann durch eine für den Kanaltyp spezifischere Zeilengruppe überschrieben werden, z. B. RcvSndBuffSize. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

Multi Ab IBM MQ 8.0 werden neue Warteschlangenmanager automatisch mit der Standardeinstellung 0 erstellt (siehe Abschnitt „Beispielzeilengruppe” auf Seite 162).

**RcvBuffSize = Zahl |`0` (Standardwert)**
Die Größe des TCP/IP-Empfangspuffers in Byte, der vom empfangenden Kanalende verwendet wird. Dieser Zeilengruppenwert kann durch eine für den Kanaltyp spezifischere Zeilengruppe überschrieben werden, z. B. RcvRcvBuffSize. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

Multi Ab IBM MQ 8.0 werden neue Warteschlangenmanager automatisch mit der Standardeinstellung 0 erstellt (siehe Abschnitt „Beispielzeilengruppe” auf Seite 162).

**RcvSndBuffSize = Zahl |0 (Standardwert)**
Die Größe des TCP/IP-Sendepuffers in Byte, der vom senderseitigen Ende eines Empfängerkanals verwendet wird. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

Multi Ab IBM MQ 8.0 werden neue Warteschlangenmanager automatisch mit der Standardeinstellung 0 erstellt (siehe Abschnitt „Beispielzeilengruppe” auf Seite 162).

**RcvRcvBuffSize = Zahl |0 (Standardwert)**
Die Größe des TCP/IP-Empfangspuffers in Byte, der vom empfangenden Ende eines Empfängerkanals verwendet wird. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

Multi Ab IBM MQ 8.0 werden neue Warteschlangenmanager automatisch mit der Standardeinstellung 0 erstellt (siehe Abschnitt „Beispielzeilengruppe” auf Seite 162).

**SvrSndBuffSize = Zahl |0 (Standardwert)**
Die Größe des TCP/IP-Sendepuffers (in Byte), der vom Serverende eines Clientverbindungs-Serververbindungskanals verwendet wird. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

Multi Ab IBM MQ 8.0 werden neue Warteschlangenmanager automatisch mit der Standardeinstellung 0 erstellt (siehe Abschnitt „Beispielzeilengruppe” auf Seite 162).

**SvrRcvBuffSize = Zahl |0 (Standardwert)**
Die Größe des TCP/IP-Empfangspuffers (in Byte), der vom Serverende eines Clientverbindungs-Serververbindungskanals verwendet wird. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

Multi Ab IBM MQ 8.0 werden neue Warteschlangenmanager automatisch mit der Standardeinstellung 0 erstellt (siehe Abschnitt „Beispielzeilengruppe” auf Seite 162).

## Beispielzeilengruppe

```
TCP:
   SndBuffSize=0
   RcvBuffSize=0
   RcvSndBuffSize=0
   RcvRcvBuffSize=0
   ClntSndBuffSize=0
   ClntRcvBuffSize=0
   SvrSndBuffSize=0
   SvrRcvBuffSize=0
```

**Anmerkung:** Multi Für neue Warteschlangenmanager auf Multiplatforms werden die Standardgrößen der TCP-Sende-und -Empfangspuffer in der TCP-Zeilengruppe der Datei `qm.ini file` für die Verwaltung durch das Betriebssystem festgelegt. Wie im vorherigen Beispiel gezeigt, werden neue WS-Manager automatisch mit der Standardeinstellung 0 für die Sende-und Empfangspuffer erstellt. Dies gilt nur für neue Warteschlangenmanager. Die Einstellungen der TCP-Sende- und -Empfangspuffer für Warteschlangenmanager, die aus früheren Versionen von IBM MQ migriert werden, werden beibehalten.

Wenn die Eigenschaften für die TCP-Puffergröße aus der Datei `qm.ini` entfernt werden, wird der Standardpuffer auf 32Kgesetzt. Sie sollten Vorsicht walten lassen, wenn Sie diesen Standardwert verwenden, da 32K möglicherweise kein geeigneter Puffer für alle Messaging-Szenarien ist.

Wenn die Eigenschaften des TCP-Sende-und Empfangspuffers auf null gesetzt sind, werden die Betriebssystemstandardwerte verwendet. Die Methode zur Auswahl dieser Standardwerte variiert je nach Betriebssystem, kann aber in der Regel auf den "tcp"-oder get/setsockopt () OS-Handbuchseiten gefunden werden.

## Multi Zeilengruppe TuningParameters in der Datei qm.ini

Die Zeilengruppe TuningParameters gibt Optionen für die Optimierung des Warteschlangenmanagers an.

**SuppressDspAuthFail= YES |NO (Standardwert)**
Wenn der Wert auf YESgesetzt ist, unterdrückt der Warteschlangenmanager die Generierung von Berechtigungsereignissen und das Schreiben von AMQ8077 -Fehlernachrichten in das Fehlerprotokoll, wenn eine Berechtigungsprüfung fehlschlägt, wenn die Verbindung über die Berechtigung + dsp für ein Objekt verfügt.

**ImplSyncOpenOutput= *Wert***
**ImplSyncOpenOutput** ist die minimale Anzahl von Anwendungen, für die die Warteschlange geöffnet ist, bevor ein impliziter Synchronisationspunkt für eine persistente, außerhalb von Synchronisationspunktstellen aktiviert werden kann. Der Standardwert von **ImplSyncOpenOutput** ist 2.

Dies hat zur Folge, dass, wenn nur eine Anwendung vorhanden ist, die diese Warteschlange für eine Eintragsoperation geöffnet hat, **ImplSyncOpenOutput** ausgeschaltet wird.

Die Angabe von `ImplSyncOpenOutput=1` bedeutet, dass immer ein impliziter Synchronisationspunkt berücksichtigt wird. Sie können einen beliebigen positiven ganzzahligen Wert festlegen. Wenn Sie nie möchten, dass ein impliziter Synchronisationspunkt hinzugefügt wird, setzen Sie `ImplSyncOpenOutput=OFF`.

**UniformClusterName=*Name des Clusters***
Der Name des IBM MQ-Clusters, den Sie als einheitlichen Cluster verwenden.

**OAMLdapConnectTimeout=Zeit|0 (Standardwert)**
Die maximale Zeit (in Sekunden), die der LDAP-Client wartet, um eine TCP-Verbindung zum Server herzustellen. Wenn Sie mehrere LDAP-Server über eine Verbindungsnamensliste angeben, gilt das Zeitlimit für jeden einzelnen Verbindungsversuch. Daher wird versucht, eine Verbindung mit dem nächsten Eintrag in der Namensliste herzustellen, wenn dieses Zeitlimit erreicht ist.

`Zeit` hat einen Maximalwert von 3600 Sekunden und der Wert `0`(dies ist der Mindestwert und der Standardwert) bedeutet, dass die Wartezeit unbegrenzt ist.

**OAMLdapQueryTimeLimit=Zeit|0 (Standardwert)**
Die maximale Zeit (in Sekunden), die der LDAP-Client wartet, um eine Antwort auf eine LDAP-Anforderung vom Server zu empfangen, sobald eine Verbindung hergestellt wurde und eine LDAP-Anforderung gesendet wurde.

`Zeit` hat einen Maximalwert von 3600 Sekunden und der Wert `0`(dies ist der Mindestwert und der Standardwert) bedeutet, dass die Wartezeit unbegrenzt ist.

V 9.3.0.5 V 9.3.2 **OAMLdapResponseWarningTime=*Schwellenwert***
Wenn eine Verbindung zu einem LDAP-Server länger als die im Parameter **OAMLdapResponseWarningTime** angegebene Anzahl von Sekunden dauerte, wird eine AMQ5544W -Nachricht in das Fehlerprotokoll geschrieben. Der Standardschwellenwert ist 10 Sekunden.

**ExpiryInterval**
Gibt die Häufigkeit an, mit der der Warteschlangenmanager die Warteschlangen nach abgelaufenen Nachrichten durchsucht, die noch nicht durch andere Warteschlangenaktivitäten bereinigt wurden. Dies ist ein Zeitintervall in Sekunden.

Standardmäßig wird der Ablaufscanner ungefähr alle fünf Minuten in IBM MQ -Produktionsbuilds ausgeführt.

**Vorsicht:** Das Ändern des Werts **ExpiryInterval** ist normalerweise nicht erforderlich. Sie sollten diesen Wert nur unter Anleitung des IBM Support ändern.

**LivenessHeartBeatLen**
Konfiguriert die Häufigkeit, mit der der Warteschlangenmanager prüft, ob in das Protokoll geschrieben wird, mit einer angemessenen Geschwindigkeit. Der Maximalwert für **LivenessHeartBeatLen** beträgt 600 Sekunden (10 Minuten) und der Mindestwert ist 0, was dazu führt, dass die Prüfung vollständig inaktiviert wird.

**Vorsicht:** In den meisten Fällen ist es nicht erforderlich, die Häufigkeit dieser Prüfungen zu ändern. Nehmen Sie keine Änderungen vor, es sei denn, Sie werden vom IBM Support dazu aufgefordert.

**ECHeartBeatLen**
Konfiguriert die Häufigkeit der allgemeinen Statusprüfungen des Warteschlangenmanagers. Der Mindestwert für **ECHeartBeatLen** beträgt 10000 Millisekunden (10 Sekunden) und der Maximalwert 60000 Millisekunden (60 Sekunden).

**Vorsicht:** In den meisten Fällen ist es nicht erforderlich, die Häufigkeit dieser Prüfungen zu ändern. Nehmen Sie keine Änderungen vor, es sei denn, Sie werden vom IBM Support dazu aufgefordert.

**FileLockHeartBeatLen**
Ändert den Standardwert für die Dateisperrenprüfungen für einen Warteschlangenmanager mit mehreren Instanzen, die der Ausführungscontroller regelmäßig ausführt, um sicherzustellen, dass er weiterhin die exklusive Sperre für die primäre Datei mit mehreren Instanzen hält. Standardmäßig werden diese Dateisperrenprüfungen alle 20 Sekunden durchgeführt. Der Mindestwert für **FileLockHeartBeatLen** beträgt 10 Sekunden und der Maximalwert 600 Sekunden (10 Minuten).

**Vorsicht:** In den meisten Fällen ist es nicht erforderlich, die Häufigkeit dieser Prüfungen zu ändern. Nehmen Sie keine Änderungen vor, es sei denn, Sie werden vom IBM Support dazu aufgefordert.

## Beispielzeilengruppe

V 9.3.0.5 V 9.3.2

```
TuningParameters:
       SuppressDspAuthFail=NO
       ImplSyncOpenOutput=2
       OAMLdapConnectTimeout=60
       OAMLdapQueryTimeLimit=60
       OAMLdapResponseWarningTime=10
       ExpiryInterval=300
```

**Zugehörige Konzepte**
Impliziter Synchronisationspunkt

# Multi Zeilengruppe 'Variables' in der Datei qm.ini

Die Zeilengruppe 'Variables' gibt Konfigurationsvariablen für die Verwendung mit automatischen Uniform-Clustern an.

Sie können Attribute, die in der Zeilengruppe 'Variables' aufgelistet sind, während der automatischen Clusterkonfiguration von CONNAME und der MQSC-Felder des Kanalnamens eines Clusterempfängerkanals verwenden. -Konfigurationsvariablen können nicht in einem anderen Element eines MQSC-Scripts verwendet werden.

***Attribut=Wert***
Gibt einen Namen und den zugehöriger Wert an, die bei MQSC-Definitionen eingefügt werden.

Die *Attribut=Wert*-Paare können beim Erstellen eines Warteschlangenmanagers mit der Befehlszeilenoption **-iv** im Befehl **crtmqm** übergeben werden.

## Beispielzeilengruppe

```
Variables:
       CONNAME=127.0.0.1(1414)
```

**Zugehörige Konzepte**

„Automatischer Ausgleich von Anwendungen” auf Seite 444
Die Verteilung und Verfügbarkeit von Anwendungen durch den automatischen Anwendungsausgleich wird erheblich verbessert, indem ein IBM MQ-Uniform-Cluster aktiviert wird, mit dem die Anwendungsverteilung im gesamten Cluster genau verwaltet wird, anstatt auf eine beliebige Festlegung oder manuelle Fixierung von Anwendungen auf bestimmte Warteschlangenmanager angewiesen zu sein.

**Zugehörige Tasks**

„Neuen Uniform-Cluster erstellen” auf Seite 459
Hier finden Sie Informationen, wie ein neuer Uniform-Cluster erstellt wird.

**Zugehörige Verweise**

„Automatische Clusterkonfiguration verwenden” auf Seite 463
Sie konfigurieren IBM MQ, um die automatische Konfiguration zu aktivieren, indem Sie die `qm.ini`-Konfigurationsinformationen ändern.

## Multi Zeilengruppe 'XAResourceManager' in der Datei 'qm.ini'

Die Zeilengruppe XAResourceManager gibt Informationen zu den Ressourcenmanagern an, die an globalen Arbeitseinheiten beteiligt sind, die vom Warteschlangenmanager koordiniert werden.

Mithilfe der Zeilengruppe XAResourceManager in der Datei `qm.ini` können Sie Informationen zu den Ressourcenmanagern angeben, die an globalen Arbeitseinheiten beteiligt sind, die vom Warteschlangenmanager koordiniert werden.

Windows Linux Alternativ können Sie unter Linux (x86 und x86-64) und Windowsdie IBM MQ Explorer -Eigenschaftenseite des XA-Ressourcenmanagerwarteschlangenmanagers verwenden.

Fügen Sie die Konfigurationsinformationen für den XA-Ressourcenmanager für jede Instanz eines Ressourcenmanagers, der an globalen Arbeitseinheiten beteiligt ist, manuell hinzu; es werden keine Standardwerte bereitgestellt.

Weitere Informationen zu Ressourcenmanagerattributen finden Sie unter Datenbankkoordinierung .

**Name = *name* (obligatorisch)**
Dieses Attribut gibt die Ressourcenmanagerinstanz an.

Der Wert für `Name` kann bis zu 31 Zeichen lang sein. Sie können den Namen des Ressourcenmanagers verwenden, wie er in der XA-Switch-Struktur definiert ist. Wenn Sie jedoch mehr als eine Instanz desselben Ressourcenmanagers verwenden, müssen Sie für jede Instanz einen eindeutigen Namen erstellen. Sie können die Eindeutigkeit sicherstellen, indem Sie z. B. den Namen der Datenbank in die Zeichenfolge `Name` einschließend.

IBM MQ verwendet den Wert `Name` in Nachrichten und in der Ausgabe aus dem Befehl `dspmqtrn`.

Ändern Sie den Namen einer Ressourcenmanagerinstanz nicht oder löschen Sie den zugehörigen Eintrag aus den Konfigurationsdaten, sobald der zugehörige Warteschlangenmanager gestartet wurde und der Name des Ressourcenmanagers wirksam ist.

**SwitchFile= *name* (obligatorisch)**
Der vollständig qualifizierte Name der Ladedatei, die die XA-Switchstruktur des Ressourcenmanagers enthält.

Wenn Sie einen 64-Bit-Warteschlangenmanager mit 32-Bit-Anwendungen verwenden, sollte der Wert von `name` nur den Basisnamen der Ladedatei enthalten, die die XA-Switchstruktur des Ressourcenmanagers enthält.

Die 32-Bit-Datei wird aus dem Pfad, der durch `ExitsDefaultPath` angegeben wurde, in die Anwendung geladen.

Die 64-Bit-Datei wird aus dem von `ExitsDefaultPath64` angegebenen Pfad in den WS-Manager geladen.

**XAOpenString= *string* (optional)**
Die Zeichenfolge der Daten, die an den Eingangspunkt xa_open des Ressourcenmanagers übergeben werden sollen. Der Inhalt der Zeichenfolge hängt vom Ressourcenmanager selbst ab. Die Zeichenfolge könnte z. B. die Datenbank angeben, auf die diese Instanz des Ressourcenmanagers zugreifen soll. Weitere Informationen zum Definieren dieses Attributs finden Sie unter:

- Ressourcenmanagerkonfigurationsinformationen für Db2 hinzufügen
- Ressourcenmanagerkonfigurationsinformationen für Oracle hinzufügen
- Ressourcenmanager-Konfigurationsinformationen für Sybase hinzufügen
- Ressourcenmanagerkonfigurationsinformationen für Informix hinzufügen

und ziehen Sie die Dokumentation zu Ihrem Ressourcenmanager nach der entsprechenden Zeichenfolge zu Rate.

**XACloseString= *string* (optional)**
Die Zeichenfolge der Daten, die an den Eingangspunkt xa_close des Ressourcenmanagers übergeben werden sollen. Der Inhalt der Zeichenfolge hängt vom Ressourcenmanager selbst ab. Weitere Informationen zum Definieren dieses Attributs finden Sie unter:

- Ressourcenmanagerkonfigurationsinformationen für Db2 hinzufügen
- Ressourcenmanagerkonfigurationsinformationen für Oracle hinzufügen
- Ressourcenmanager-Konfigurationsinformationen für Sybase hinzufügen
- Ressourcenmanagerkonfigurationsinformationen für Informix hinzufügen

und ziehen Sie die Datenbankdokumentation für die entsprechende Zeichenfolge zu Rate.

**ThreadOfControl = THREAD | PROCESS**
Windows Dieses Attribut ist für Windows obligatorisch. Der Warteschlangenmanager verwendet diesen Wert für die Serialisierung, wenn er den Ressourcenmanager von einem seiner eigenen Multithread-Prozesse aus aufrufen muss.

**THREAD**
Der Ressourcenmanager ist vollständig *Threadwissen* . In einem Multithread-Prozess von IBM MQ können XA-Funktionsaufrufe an den externen Ressourcenmanager aus mehreren Threads gleichzeitig ausgeführt werden.

**PROCESS**
Der Ressourcenmanager ist nicht *threadsicher* . In einem Multithread-Prozess von IBM MQ kann immer nur jeweils ein XA-Funktionsaufruf an den Ressourcenmanager erfolgen.

Der Eintrag **ThreadOfControl** gilt nicht für XA-Funktionsaufrufe, die vom WS-Manager in einem Multithread-Anwendungsprozess abgesetzt werden. Im Allgemeinen erfordert eine Anwendung, die parallele Arbeitseinheiten in verschiedenen Threads hat, diesen Modus der Operation, die von jedem der Ressourcenmanager unterstützt wird.

## Beispielzeilengruppe

```
XAResourceManager:
   Name=DB2 Resource Manager Bank
   SwitchFile=/usr/bin/db2swit
   XAOpenString=MQBankDB
   XACloseString=
   ThreadOfControl=THREAD
```

**Anmerkung:** Die maximale Anzahl an XAResourceManager-Zeilengruppen ist auf 255 begrenzt. Sie sollten jedoch nur eine kleine Anzahl von Zeilengruppen verwenden, um eine Verschlechterung der Transaktionsleistung zu vermeiden.

## IBM i Beispieldatei qm.ini für IBM i

Ein Beispiel, das zeigt, wie Gruppen von Attributen in einer Konfigurationsdatei des Warteschlangenmanagers für IBM iangeordnet werden können.

```
#*******************************************************************#
#* Module Name: qm.ini                                             *#
#* Type       : IBM MQ queue manager configuration file            *#
# Function    : Define the configuration of a single queue manager *#
#*                                                                 *#
#*******************************************************************#
#* Notes      :                                                    *#
#* 1) This file defines the configuration of the queue manager     *#
#*                                                                 *#
#*******************************************************************#
Log:
LogPath=QMSATURN.Q
LogReceiverSize=65536

CHANNELS:
MaxChannels = 20       ; Maximum number of channels allowed.
                       ; Default is 100.
MaxActiveChannels = 10 ; Maximum number of channels allowed to be
                       ; active at any time. The default is the
                       ; value of MaxChannels.

TCP:                   ; TCP/IP entries.
KeepAlive = Yes        ; Switch KeepAlive on.
SvrSndBuffSize=20000   ; Size in bytes of the TCP/IP send buffer for each
                       ; channel instance. Default is 32768.
SvrRcvBuffSize=20000   ; Size in bytes of the TCP/IP receive buffer for each
                       ; channel instance. Default is 32768.
Connect_Timeout=10000  ; Number of seconds before an attempt to connect the
                       ; channel instance times out. Default is zero (no timeout).

QMErrorLog:
ErrorLogSize = 262144
ExcludeMessage = 7234
SuppressMessage = 9001,9002,9202
SuppressInterval = 30

TuningParameters:
   ImplSyncOpenOutput=2
```

## ALW Installationskonfigurationsdatei, mqinst.ini

Auf AIX and Linux -Systemen enthält die Installationskonfigurationsdatei `mqinst.ini`Informationen zu allen IBM MQ -Installationen. Auf Windows -Systemen befinden sich die Informationen zur Installationskonfiguration in der Registry.

### Position der `mqinst.ini`-Datei

Linux AIX

Die Datei `mqinst.ini` befindet sich im Verzeichnis `/etc/opt/mqm` auf AIX and Linux-Systemen. Sie enthält Informationen darüber, welche Installation (falls vorhanden) die primäre Installation sowie die folgenden Informationen für jede Installation enthält:

- Der Installationsname
- Die Installationsbeschreibung
- Die Installations-ID
- Der Installationspfad

**Wichtig:** Die Datei `mqinst.ini` darf nicht direkt bearbeitet oder referenziert werden, da ihr Format nicht festgelegt ist und sich ändern kann.

Die Installationskennung (nur für die interne Verwendung) wird automatisch festgelegt und darf nicht geändert werden.

Anstatt die `mqinst.ini`-Datei direkt zu bearbeiten, müssen Sie die folgenden Befehle verwenden, um die Werte in der Datei zu erstellen, zu löschen, abzufragen und zu ändern:

crtmqinst zum Erstellen von Einträgen.
dltmqinst zum Löschen von Einträgen.
dspmqinst , um Einträge anzuzeigen.
setmqinst zum Festlegen von Einträgen.

### Informationen zur Installationskonfiguration unter Windows

Windows

Es ist keine `mqinst.ini`-Datei unter Windows vorhanden. Die Informationen zur Installationskonfiguration sind in der Registry enthalten und werden in der folgenden Datei gehalten:

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

**Wichtig:** Dieser Schlüssel darf nicht direkt bearbeitet oder referenziert werden, da sein Format nicht fixiert ist und sich ändern kann.

Stattdessen müssen Sie die folgenden Befehle verwenden, um die Werte in der Registry abzufragen und zu ändern:

dspmqinst , um Einträge anzuzeigen.
setmqinst zum Festlegen von Einträgen.

Unter Windows stehen die Befehle **`crtmqinst`** und **`dltmqinst`** nicht zur Verfügung. Die Installations- und Deinstallationsprozesse bearbeiten die Erstellung und das Löschen der erforderlichen Registry-Einträge.

## Multi IBM MQ MQI client -Konfigurationsdatei, mqclient.ini

Sie können Ihre Clients mithilfe von Attributen in einer Textdatei konfigurieren. Diese Attribute können durch Umgebungsvariablen oder auf andere plattformspezifische Methoden überschrieben werden.

Sie konfigurieren IBM MQ MQI clients mithilfe einer Textdatei, die der Konfigurationsdatei des Warteschlangenmanagers `qm.ini`ähnelt. Die Datei enthält eine Reihe von Zeilengruppen, die jeweils eine Reihe von Zeilen im Format **`attribute-name`** = *value* enthalten.

Die IBM MQ MQI client-Konfigurationsdatei wird im Allgemeinen `mqclient.ini` genannt, Sie können ihr aber auch einen anderen Namen geben. Die Konfigurationsinformationen in dieser Datei gelten für die folgenden Plattformen:

- ALW AIX, Linux, and Windows
- IBM i IBM i

  **Anmerkung:** Unter IBM igibt es keine Standarddatei `mqclient.ini` . Sie können die Datei jedoch im IBM i Integrated File System (IFS) erstellen.

Weitere Informationen finden Sie unter „Position der Clientkonfigurationsdatei” auf Seite 170.

**Anmerkung:** z/OS Die Plattform z/OS kann nicht zur Ausführung von IBM MQ -Clients verwendet werden. Daher ist die Datei `mqclient.ini` unter IBM MQ for z/OSnicht vorhanden.

Die Attribute in der Konfigurationsdatei IBM MQ MQI client gelten für Clients, die Folgendes verwenden:

- Die MQI
- IBM MQ classes for Java
- IBM MQ classes for JMS
- IBM MQ classes for .NET

- XMS

Obwohl die Attribute in der IBM MQ MQI client-Konfigurationsdatei für die meisten IBM MQ-Clients gelten, gibt es bestimmte Attribute, die nicht von verwalteten .NET- und XMS .NET-Clients oder von Clients verwendet werden, die entweder die IBM MQ classes for Java oder die IBM MQ classes for JMS verwenden. Weitere Informationen finden Sie unter „Welche IBM MQ-Clients können die einzelnen Attribute lesen” auf Seite 171.

Die Konfigurations-Features gelten für alle Verbindungen, die eine Clientanwendung an alle WS-Manager stellt, statt spezifisch für eine einzelne Verbindung zu einem Warteschlangenmanager zu sein. Attribute, die sich auf eine Verbindung zu einem einzelnen Warteschlangenmanager beziehen, können programmgestützt konfiguriert werden, z. B. mithilfe einer MQCD-Struktur oder mithilfe einer Clientkanaldefinitionstabelle (CCDT).

Im Folgenden finden Sie ein Beispiel für eine Clientkonfigurationsdatei:

```
#* Module Name: mqclient.ini                                          *#
#* Type       : IBM MQ MQI client configuration file                  *#
#  Function   : Define the configuration of a client                  *#
#*                                                                    *#
#**********************************************************************#
#* Notes      :                                                       *#
#* 1) This file defines the configuration of a client                 *#
#*                                                                    *#
#**********************************************************************#

ClientExitPath:
   ExitsDefaultPath=/var/mqm/exits
   ExitsDefaultPath64=/var/mqm/exits64

TCP:
   Library1=DLLName1
   KeepAlive = Yes
   ClntSndBuffSize=32768
   ClntRcvBuffSize=32768
   Connect_Timeout=0

MessageBuffer:
   MaximumSize=-1
   Updatepercentage=-1
   PurgeTime=0

LU62:
   TPName
   Library1=DLLName1
   Library2=DLLName2

PreConnect:
    Module=myMod
    Function=myFunc
    Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
    Sequence=1

CHANNELS:
DefRecon=YES
      ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)

Connection:
     ApplName=ExampleApplName
```

Sie können nicht mehrere Kanalverbindungen mithilfe der Clientkonfigurationsdatei einrichten.

Umgebungsvariablen, die in Releases vor IBM WebSphere MQ 7.0 unterstützt wurden, werden in späteren Releases weiterhin unterstützt, und wenn eine solche Umgebungsvariable mit einem entsprechenden Wert in der Clientkonfigurationsdatei übereinstimmt, überschreibt die Umgebungsvariable den Wert der Clientkonfigurationsdatei.

Für eine Clientanwendung, die IBM MQ classes for JMS verwendet, können Sie die Clientkonfigurationsdatei auch auf die folgenden Arten überschreiben:

- Durch Festlegen von Eigenschaften in der JMS-Konfigurationsdatei.

- Durch Festlegen von Java-Systemeigenschaften, wodurch auch die JMS-Konfigurationsdatei überschrieben wird.

Für den .NET-Client können Sie die Clientkonfigurationsdatei und die entsprechenden Umgebungsvariablen auch mit der Anwendungskonfigurationsdatei von .NET überschreiben.

## Kommentare in der Konfigurationsdatei

Linux AIX

Sie können das Semikolon ';' und das Hashzeichen '#' verwenden, um den Anfang eines Kommentars in der Konfigurationsdatei zu markieren. Dies kann eine ganze Zeile als Kommentar markieren oder einen Kommentar am Ende einer Zeile angeben, die nicht im Wert einer Einstellung enthalten sein wird.

Wenn ein Wert eines dieser Zeichen erfordert, müssen Sie dieses Zeichen mit dem Backslash-Zeichen '\' als Escapezeichen versehen.

Das folgende Beispiel zeigt die Verwendung von Kommentaren in der Konfigurationsdatei:

```
# Example of an SSL stanza with comments
SSL:
    ClientRevocationChecks=REQUIRED ; Example of an end of line comment
    SSLCryptoHardware=GSK_PKCS11=/driver\;label\;password\;SYMMETRIC_CIPHER_ON # Example of es
caped comment characters.
```

**Zugehörige Konzepte**

Konfigurationsdatei für die IBM MQ-Klassen für Java

## Multi Position der Clientkonfigurationsdatei

Eine IBM MQ MQI client-Konfigurationsdatei kann an einer Reihe von Positionen gehalten werden.

Eine Clientanwendung verwendet den folgenden Suchpfad, um die IBM MQ MQI client-Konfigurationsdatei zu lokalisieren:

1. Die durch die Umgebungsvariable **MQCLNTCF**angegebene Position.

   Das Format dieser Umgebungsvariablen ist eine vollständige URL. Dies bedeutet, dass der Dateiname nicht unbedingt `mqclient.ini` sein muss, und erleichtert das Platzieren der Datei in einem mit einem Netz verbundenen Dateisystem.

   **Anmerkungen:**

   - C-, .NET -und XMS -Clients unterstützen nur das Protokoll `file:` . Das Protokoll `file:` wird vorausgesetzt, wenn die Zeichenfolge URL nicht mit `protocol:` beginnt.
   - Um Java 1.4.2 JREs zu ermöglichen, die das Lesen von Umgebungsvariablen nicht unterstützen, kann die Umgebungsvariable **MQCLNTCF** mit einer **MQCLNTCF** Java -Systemeigenschaft überschrieben werden.
2. Eine Datei mit dem Namen `mqclient.ini` in dem aktuellen Arbeitsverzeichnis der Anwendung.
3. Eine Datei mit dem Namen `mqclient.ini` im IBM MQ-Datenverzeichnis für AIX, Linux, and Windows-Systeme.

   **Anmerkungen:**

   - Das IBM MQ-Datenverzeichnis ist auf bestimmten Plattformen nicht vorhanden, z. B. IBM i und z/OS, oder in Fällen, in denen der Client mit einem anderen Produkt versorgt wurde.

     IBM i Unter IBM igibt es keine Standarddatei `mqclient.ini` . Die Datei kann jedoch im IBM i Integrated File System (IFS) im Verzeichnis `/QIBM/UserData/mqm/`erstellt werden und die Umgebungsvariable **MQCLNTCF** kann so definiert werden, dass sie auf sie verweist. For example:

     ```
     ADDENVVAR ENVVAR(MQCLNTCF) VALUE('QIBM/UserData/mqm/mqclient.ini') REPLACE(*YES)
     ```

Weitere Beispiele für Umgebungsvariablen finden Sie unter „Beschreibung der Umgebungsvariablen” auf Seite 68.

z/OS Die Plattform z/OS kann nicht zur Ausführung von IBM MQ -Clients verwendet werden. Daher ist die Datei `mqclient.ini` unter IBM MQ for z/OSnicht vorhanden.

- Linux AIX Auf AIX and Linux -Systemen ist das Verzeichnis `/var/mqm`.
- Windows Auf Windows -Plattformen konfigurieren Sie die Umgebungsvariable **MQ_DATA_PATH** während der Installation so, dass sie auf das Datenverzeichnis verweist. Dies ist normalerweise `C:\ProgramData\IBM\MQ.` .

  **Anmerkung:** Wenn Sie nur einen Client installieren, lautet die Umgebungsvariable möglicherweise **MQ_FILE_PATH**.
- Um Java 1.4.2 JREs zu ermöglichen, die das Lesen von Umgebungsvariablen nicht unterstützen, können Sie die Umgebungsvariable **MQ_DATA_PATH** mit einer **MQ_DATA_PATH** Java -Systemeigenschaft manuell überschreiben.

4. Eine Datei mit dem Namen `mqclient.ini` in einem Standardverzeichnis, das für die Plattform geeignet ist und für die Benutzer zugänglich ist:
   - Für alle Java-Clients ist dies der Wert der Systemeigenschaft von `user.home` Java.
   - Linux AIX Für C-Clients auf AIX and Linux -Plattformen ist dies der Wert der Umgebungsvariable **HOME** .
   - Windows Für C-Clients unter Windows sind dies die verknüpften Werte der Umgebungsvariablen **HOMEDRIVE** und **HOMEPATH** .

## Multi Welche IBM MQ-Clients können die einzelnen Attribute lesen

Die meisten Attribute in der IBM MQ MQI client-Konfigurationsdatei können vom C-Client und von den nicht verwalteten .NET-Clients verwendet werden. Allerdings gibt es einige Attribute, die nicht von verwalteten .NET- und XMS .NET-Clients gelesen werden, oder von Clients, die entweder IBM MQ classes for Java oder IBM MQ classes for JMS verwenden.

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| **CHANNELS-Zeilengruppe** | | | | | | |
| CCSID | Die Nummer des codierten Zeichensatzes, der verwendet werden soll. | Ja | Nein | Nein | Ja | Ja |
| ChannelDefinitionDirectory | Der Verzeichnispfad zu der Datei, die die Definitionstabelle für den Clientkanal enthält. | Ja | Nein | Nein | Ja | Ja |

| *Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)* | | | | | | |
|---|---|---|---|---|---|---|
| **`mqclient.ini`, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
| ChannelDefinitionFile | Der Name der Datei, die die Definitionstabelle für den Clientkanal enthält. | Ja | Nein | Nein | Ja | Ja |
| ReconDelay | Eine Verwaltungsoption, mit der die Verzögerung für die Verbindungswiederverbindung für Clientprogramme konfiguriert werden kann, die die Verbindung automatisch herstellen können. | Ja | Nein | Ja | Ja | Ja |
| DefRecon | Eine Verwaltungsoption, mit der Clientprogramme automatisch erneut verbunden werden können, oder um die automatische Neuverbindung eines Clientprogramms zu inaktivieren, das so geschrieben wurde, dass es automatisch wieder verbunden wird. | Ja | Nein | Ja | Ja | Ja |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| MQReconnectTimeout | Das Zeitlimit in Sekunden, in dem die Verbindung zu einem Client wieder hergestellt werden kann. | Ja | Nein | Nein | Ja | Nein |
| ServerConnectionParms | Die Position des IBM MQ-Servers und das zu verwendende Kommunikationsverfahren. | Ja | Nein | Nein | Ja | Ja |
| Put1DefaultAlwaysSync | Steuert das Verhalten des Funktionsaufrufs MQPUT1 mit der Option MQPMO_RESPONSE_AS_Q_DEF. | Ja | Ja | Ja | Ja | Ja |
| PasswordProtection | Hier können Sie geschützte Kennwörter in der MQCSP-Struktur festlegen, statt SSL oder TLS zu verwenden. | Ja | Ja | Ja | Ja | Ja |
| **Zeilengruppe 'ClientExitPath'** | | | | | | |
| ExitsDefaultPath | Gibt die Position der 32-Bit-Kanalexits für Clients an. | Ja | Ja | Ja | Ja | Ja |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| ExitsDefaultPath64 | Gibt die Position der 64-Bit-Kanalexits für Clients an. | Ja | Ja | Ja | Ja | Ja |
| JavaExitsClassPath | Die Werte, die dem Klassenpfad hinzugefügt werden sollen, wenn ein Java-Exit ausgeführt wird. | Nein | Ja | Ja | Nein | Nein |
| **Zeilengruppe 'Connection'** | | | | | | |
| ApplName | Der Anwendungsname, der in der Clientkonfigurationsdatei angegeben ist. | Ja | Nein | Nein | Nein | Nein |
| **JMQI-Zeilengruppe** | | | | | | |
| useMQCSPauthentication | Steuert, ob IBM MQ classes for Java- und IBM MQ classes for JMS-Anwendungen bei der Authentifizierung mit einem WS-Manager den Kompatibilitätsmodus oder den MQCSP-Authentifizierungsmodus verwenden sollen. | Nein | Ja | Ja | Nein | Nein |
| **Zeilengruppe 'MessageBuffer'** | | | | | | |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| MaximumSize | Größe (in Kilobyte) des Read-Ahead-Puffers im Bereich von 1 bis 999 999. | Ja | Ja | Ja | Ja | Ja |
| PurgeTime | Intervall (in Sekunden), nach dem Nachrichten, die im Puffer für die Lesepuffer (Read-ahead-Puffer) bleiben, gelöscht | Ja | Ja | Ja | Ja | Ja |
| UpdatePercentage | Der Prozentwert für die Aktualisierung im Bereich von 1-100, der bei der Berechnung des Schwellenwerts verwendet wird, um zu ermitteln, wann eine Clientanwendung eine neue Anforderung an den Server stellt. | Ja | Ja | Ja | Ja | Ja |
| **Zeilengruppe 'PreConnect'** | | | | | | |
| Daten | URL des Repositorys, in dem Verbindungsdefinitionen gespeichert werden. | Ja | Nein | Nein | Nein | Nein |

| *Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)* | | | | | | |
|---|---|---|---|---|---|---|
| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
| FUNCTION | Der Name des funktionalen Eingangspunkts in der Bibliothek, die den PreConnect-Exit-Code enthält. | Ja | Nein | Nein | Nein | Nein |
| Modul | Der Name des Moduls, das den API-Exit-Code enthält. | Ja | Nein | Nein | Nein | Nein |
| Sequenz | Die Sequenz, in der dieser Exit relativ zu anderen Exits aufgerufen wird. | Ja | Nein | Nein | Nein | Nein |
| **Sicherheitszeilengruppe** | | | | | | |
| DisableClientAMS | Aktiviert oder inaktiviert AMS für Clientverbindungen zu einem Warteschlangenmanager. | Ja | Ja | Ja | Nein | Nein |
| **SSL-Zeilengruppe** | | | | | | |
| V 9.3.0 OutboundSNI | Es wird angegeben, ob SNI-fähige Clients beim Einleiten einer TLS-Verbindung als SNI für das ferne System den Namen des IBM MQ-Zielkanals oder den Hostnamen festlegen. | Ja | Ja | Ja | Ja | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| AllowOutboundSNI | Es wird angegeben, ob SNI-fähige Clients als SNI den Namen des IBM MQ-Zielkanals für das ferne System festlegen, wenn eine TLS-Verbindung eingeleitet wird.<br>**Achtung:** V 9.3.0 Deprecated<br>Ab IBM MQ 9.3.0 ist diese Eigenschaft veraltet. Verwenden Sie stattdessen **OutboundSNI** . | Ja | Ja | Ja | Nein | Nein |
| AllowTLSV13 | Gibt an, ob ein Warteschlangenmanager die TLS 1.3 CipherSpecs verwenden kann. | Ja (C/C++-Clients) | Nein | Nein | Nein | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| CDPCheckExtensions | Gibt an, ob SSL-oder TLS-Kanäle in diesem Warteschlangenmanager versuchen, CDP-Server zu überprüfen, die in den Zertifikatserweiterungen des CrlDistributionPoint-Zertifikats benannt sind. | Ja | Nein | Nein | Nein | Nein |
| CertificateLabel | Die Zertifikatsbezeichnung der Kanaldefinition. | Ja | Nein | Nein | Nein | Nein |
| CertificateValPolicy | Bestimmt den Typ der verwendeten Zertifikatsprüfung. | Ja | Nein | Nein | Nein | Nein |
| ClientRevocationChecks | Legt fest, wie die Überprüfung der Zertifikatswiderrufung konfiguriert wird, wenn der Clientverbindungsaufruf einen SSL/TLS-Kanal verwendet. | Ja | Nein | Nein | Nein | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| EncryptionPolicySuiteB | Legt fest, ob ein Kanal eine Suite-B-kompatible Verschlüsselung verwendet und welcher Grad der Stärke verwendet werden soll. | Ja | Nein | Nein | Nein | Nein |
| V 9.3.0 EnvironmentScope | Steuert, ob IBM MQ eine einzelne IBM Global Security Kit (GSKit) -Umgebung für den gesamten Prozess oder eine GSKit -Umgebung pro Verbindung verwendet. | Ja (C-Clients) | Nein | Nein | Nein | Nein |
| MinimumRSAKeyGröße | Gibt die Mindestschlüsselgröße an, die RSA-Zertifikate haben müssen, damit sie akzeptiert werden. | Ja (C/C++-Clients) | Nein | Nein | Nein | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| OCSPAuthentication | Definiert das Verhalten von IBM MQ, wenn OCSP aktiviert ist und die OCSP-Widerrufsprüfung nicht in der Lage ist, den Status des Zertifikatswiderrufs zu ermitteln. | Ja | Nein | Nein | Nein | Nein |
| OCSPCheckExtensions | Steuert, ob IBM MQ auf die AuthorityInfoAccess-Zertifikatserweiterungen einwirkt. | Ja | Nein | Nein | Nein | Nein |
| OCSPZeitlimit | Die Anzahl der Sekunden, die bei der Ausführung einer Widerrufsprüfung auf einen OCSP-Responder gewartet wird. | Ja | Nein | Nein | Nein | Nein |
| ALW PeerCertChainValidation | Die Einstellung für die GSKit-Zertifikatsprüfung. | Ja | Nein | Nein | Nein | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| SSLCryptoHardware | Legt die Parameterzeichenfolge fest, die erforderlich ist, um die auf dem System vorhandene Verschlüsselungshardware PKCS #11 zu konfigurieren. | Ja | Nein | Nein | Nein | Nein |
| SSLCryptoHardwareKeyFile | Gibt den vollständigen Pfad und Namen der Datei an, die den Anfangsschlüssel enthält, der zum Verschlüsseln des Kennworts in der Konfigurationszeichenfolge der PKCS #11 -Verschlüsselungshardware verwendet wurde, die mit dem Attribut **SSLCryptoHardware** angegeben wird. | Ja | Nein | Nein | Nein | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| SSLFipsRequired | Gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen, wenn die Verschlüsselung in IBM MQ ausgeführt wird. | Ja | Nein | Nein | Nein | Nein |
| SSLHTTPProxyName | Die Zeichenfolge ist entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers, der von GSKit für OCSP-Prüfungen verwendet wird. | Ja | Nein | Nein | Nein | Nein |
| SSLHTTPConnectTimeout | Die Anzahl der Sekunde, die beim Ausführen einer Widerrufsprüfung auf die erfolgreiche Herstellung einer Netzverbindung zu einem HTTP-Server gewartet wird. | Ja | Nein | Nein | Nein | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| SSLKeyRepository | Die Position des Schlüsselrepositorys, in dem das digitale Zertifikat des Benutzers gespeichert ist, im Stammformat. | Ja | Nein | Nein | Nein | Nein |
| V 9.3.0 V 9.3.0 SSLKeyRepositoryKennwort | Die Kennphrase für den Zugriff auf das Schlüsselrepository. | Ja | Nein | Nein | Nein | Nein |
| SSLKeyResetCount | Die Anzahl der nicht verschlüsselten Byte, die in einem SSL- oder TLS-Kanal gesendet und empfangen wurden, bevor der geheime Schlüssel neu verhandelt wird. | Ja | Nein | Nein | Nein | Nein |
| **TCP-Zeilengruppe** | | | | | | |
| ClntRcvBuffSize | Die Größe des TCP/IP-Empfangspuffers (in Byte), der vom Clientende eines Clientverbindungs-Serververbindungskanals verwendet wird. | Ja | Ja | Ja | Ja | Ja |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| ClntSndBuffSize | Die Größe des TCP/IP-Sendepuffers (in Byte), der vom Clientende eines Clientverbindungs-Serververbindungskanals verwendet wird. | Ja | Ja | Ja | Ja | Ja |
| Connect_Timeout | Die Anzahl der Sekunden, bevor ein Versuch unternommen wird, das Socket-Zeitlimit zu verbinden. | Ja | Ja | Ja | Nein | Nein |
| IPAddressVersion | Gibt das IP-Protokoll an, das für eine Kanalverbindung verwendet werden soll. | Ja | Nein | Nein | Ja | Ja |
| KeepAlive | Schaltet die KeepAlive-Funktion ein oder aus. | Ja | Ja | Ja | Ja | Ja |
| Windows Library1 | Nur unter Windows: der Name der TCP/IP-Sockets-DLL. | Ja | Nein | Nein | Nein | Nein |
| **Zeilengruppe 'Trace'**<br>**Anmerkung:** Die Zeilengruppe 'Trace' gilt nur für die IBM MQ .NET -und XMS .NET -Clients. | | | | | | |
| V 9.3.3 MQDotnetTrace-Stufe | Wird zum Aktivieren des IBM MQ .NET -Trace verwendet. | Nein | Nein | Nein | Ja | Nein |

*Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.)*

| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
|---|---|---|---|---|---|---|
| V 9.3.3 PfadMQDotnetTrace | Verweist auf einen Ordner, in dem die IBM MQ .NET -Tracedateien erstellt werden | Nein | Nein | Nein | Ja | Nein |
| V 9.3.3 PfadMQDotnetError | Verweist auf einen Ordner, in dem Fehlerprotokolldateien für den IBM MQ .NET -Trace erstellt werden. | Nein | Nein | Nein | Ja | Nein |
| V 9.3.3 StufeXMSDotnetTrace | Wird zum Aktivieren des XMS .NET -Trace verwendet. | Nein | Nein | Nein | Nein | Ja |
| V 9.3.3 XMSDotnetTraceFilePath | Verweist auf einen Ordner, in dem die XMS .NET -Tracedateien erstellt werden. | Nein | Nein | Nein | Nein | Ja |
| V 9.3.3 SpezifikationXMSDotnetTrace | Gibt den Namen der Klasse an, für die Sie einen Trace für XMS .NETerstellen möchten | Nein | Nein | Nein | Nein | Ja |
| V 9.3.3 SpezifikationXMSDotnetTrace | Gibt die maximale Größe der Tracedatei an, die für XMS .NETgeneriert werden soll | Nein | Nein | Nein | Nein | Ja |

| Tabelle 15. Welche Attribute gelten für jeden Typ von Client (Forts.) | | | | | | |
|---|---|---|---|---|---|---|
| **mqclient.ini, Zeilengruppenname und -attribute** | **Beschreibung** | **C und nicht verwaltete .NET** | **Java** | **JMS** | **Verwaltete .NET-Instanz** | **Verwaltete XMS .NET-Instanz** |
| V 9.3.3 XMSDotnetTraceFileSize | Die Anzahl der Tracedateien, die für XMS .NETauf bewahrt werden sollen | Nein | Nein | Nein | Nein | Ja |

## V 9.3.0 Zeilengruppe Application in der Clientkonfigurationsdatei

Mithilfe der Zeilengruppe Application können Sie Attribute angeben, die das Verhalten des einheitlichen Clusterausgleichs für eine bestimmte Anwendung beeinflussen, die mit dieser Konfiguration eine Verbindung herstellt. Werte in dieser Zeilengruppe haben Vorrang vor der Zeilengruppe ApplicationDefaults , können jedoch durch eine MQBNO-Struktur überschrieben werden, die über ein Programm bereitgestellt wird.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die Anwendungszeilengruppe eingeschlossen werden:

**Name = ApplicationName**
Gibt an, auf welchen Anwendungsnamen die Optionen angewendet werden sollen.

**Typ = *Einfach,ReqRep***
Gibt an IBM MQ das allgemeine Muster der IBM MQ-Aktivität an, an der diese Anwendung teilnimmt.

**BalanceTimeout = *Never,Immediate,0-999999999, Standardwert***
Gibt IBM MQ das Zeitlimit an, bevor die Anwendungsaktivität unterbrochen werden kann, um einen Neuausgleich zu ermöglichen; entweder nie oder bis zu einem Wert von maximal 999.999.999 Sekunden mit einem Standardwert von 10 Sekunden.

**Ausgleichsoptionen = *Keine, IgnTrans***
Entweder sind keine Ausgleichsoptionen definiert oder es können Anwendungen, die derzeit an einer Transaktion beteiligt sind, sofort unterbrochen werden.

## Multi V 9.3.0 Zeilengruppe ApplicationDefaults in der Clientkonfigurationsdatei

Mithilfe der Zeilengruppe ApplicationDefaults können Sie Attribute angeben, die das Standardverhalten der gleichmäßigen Clusterverteilung für Clientanwendungen beeinflussen, die über diese Konfiguration eine Verbindung herstellen. Diese Standardwerte können entweder durch eine anwendungsspezifische Zeilengruppe Application oder durch eine MQBNO-Struktur überschrieben werden, die über ein Programm bereitgestellt wird.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die Zeilengruppe "ApplicationDefaults" aufgenommen werden:

**Typ = *Einfach,ReqRep***
Gibt an IBM MQ das allgemeine Muster der IBM MQ-Aktivität an, an der diese Anwendung teilnimmt.

**BalanceTimeout = *Never,Immediate,0-999999999, Standardwert***
Gibt IBM MQ das Zeitlimit an, bevor die Anwendungsaktivität unterbrochen werden kann, um einen Neuausgleich zu ermöglichen; entweder nie oder bis zu einem Wert von maximal 999.999.999 Sekunden mit einem Standardwert von 10 Sekunden.

**Ausgleichsoptionen = *Keine, IgnTrans***
Entweder sind keine Ausgleichsoptionen definiert oder es können Anwendungen, die derzeit an einer Transaktion beteiligt sind, sofort unterbrochen werden.

## Multi Zeilengruppe 'CHANNELS' in der Clientkonfigurationsdatei

Verwenden Sie die Zeilengruppe CHANNELS, um Informationen zu Clientkanälen anzugeben.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die Zeilengruppe CHANNELS aufgenommen werden:

**CCSID = *number***
Die Nummer des codierten Zeichensatzes, der verwendet werden soll.

Dieses Attribut kann von C-, nicht verwalteten .NET-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Die CCSID-Nummer entspricht der Umgebungsvariablen MQCCSID.

**ChannelDefinitionDirectory = *path***
Der Verzeichnispfad zu der Datei, die die Definitionstabelle für den Clientkanal enthält.

Dieses Attribut kann von C-, nicht verwalteten .NET-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Windows Auf Windows-Systemen ist der Standardwert das IBM MQ-Daten-und Protokolldateien-Verzeichnis, in der Regel `C:\ProgramData\IBM\MQ`.

Linux AIX Auf AIX and Linux-Systemen ist der Standardwert `/var/mqm`.

ChannelDefinitionDirectory kann eine URL enthalten, die in Kombination mit dem Attribut "ChannelDefinitionFile" funktioniert (siehe „URL-Zugriff auf die CCDT” auf Seite 56).

Der Pfad 'ChannelDefinitionDirectory' entspricht der Umgebungsvariablen **MQCHLLIB**.

**ChannelDefinitionFile = *filename* | `AMQCLCHL.TAB`**
Der Name der Datei, die die Definitionstabelle für den Clientkanal enthält.

Dieses Attribut kann von C-, nicht verwalteten .NET-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Die Definitionstabelle für den Clientkanal entspricht der Umgebungsvariablen **MQCHLTAB**.

**ReconDelay = `(delay[,rand]) (delay[,rand])...`**
Das Attribut ReconDelay stellt eine Verwaltungsoption zur Verfügung, mit der die Verzögerung der Verbindungswiederherleitung für Clientprogramme konfiguriert werden kann, die die Verbindung automatisch wiederherstellen können.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Nachfolgend ist eine Beispielkonfiguration zu finden:

```
ReconDelay=(1000,200)(2000,200)(4000,1000)
```

Das gezeigte Beispiel definiert eine Anfangsverzögerung von einer Sekunde plus einem zufälligen Intervall von bis zu 200 Millisekunden. Die nächste Verzögerung beträgt zwei Sekunden plus ein zufälliger Intervall von bis zu 200 Millisekunden. Alle nachfolgenden Verzögerungen sind vier Sekunden, plus ein zufälliger Zeitraum von bis zu 1000 Millisekunden.

**DefRecon = NO | YES | QMGR | DISABLED**

Das Attribut DefRecon bietet eine Verwaltungsoption, mit der Clientprogramme automatisch erneut verbunden werden können, oder um die automatische Neuverbindung eines Clientprogramms zu inaktivieren, das so geschrieben wurde, dass es automatisch wieder verbunden wird. Wenn ein Programm eine Option verwendet, wie z. B. `MQPMO_LOGICAL_ORDER`, können Sie diese Option auswählen, wenn ein Programm mit der Neuverbindung nicht kompatibel ist.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Die automatische Clientverbindungswiederholung wird von IBM MQ classes for Java nicht unterstützt.

Die automatische Clientverbindungswiederholung hängt in der Regel von zwei Werten ab:

- Option für Verbindungswiederherstellung in der Anwendung MQCONNX (oder JMS Connection Factory)
- Die Standardoption für die Verbindungswiederherstellung, die in allen verwendeten Clientverbindungsdefinitionen bereitgestellt wird (MQCD-Struktur, z. B. bereitgestellt durch eine CCDT-Datei)

Das Dateiattribut `mqclient.ini` gilt **nur** , wenn keine Kanaldefinition verwendet wird, die das Attribut **`DefReconnect`** festlegt, und sich in dieser Situation so verhält, als wäre eine angegeben worden. Das Kanalattribut **`DefReconnect`** (und daher dieses Attribut, falls zutreffend):

- Überschreiben Sie den Anwendungscode, wenn einer der beiden auf DISABLED gesetzt ist.
- Werden in allen anderen Fällen vom Anwendungscode überschrieben, wenn die Optionen in MQCONNX angegeben werden

In der Beschreibung von DEFRECON finden Sie eine Tabelle mit allen möglichen Kombinationen der bereitgestellten Werte für die Anwendungs-und Kanaldefinition.

**Anmerkungen:**

- Wenn eine MQCD verwendet wird, aber vor MQCD_VERSION_10liegt, ist der Parameter **`DefReconnnect`** nicht Teil der Struktur. In dieser Situation wird der Wert dieses fehlenden Parameters mit dem Wert `mqclient.ini` file **`DefReconnnect`** gefüllt, wenn ein Wert angegeben wird. Dies kann beispielsweise auftreten, wenn eine CCDT im Binärformat, die in einer älteren Version von IBM MQ generiert wurde, noch von einer Clientanwendung verwendet wird.
- Bei der Interpretation durch den IBM MQ -Client-Code generiert eine JSON-CCDT (siehe „CCDT im JSON-Format konfigurieren” auf Seite 47) immer MQCD-Strukturen in der neuesten Version und stellt daher immer den Standardwert (NO) für dieses Attribut bereit, sofern es nicht explizit mit einem anderen Wert vorhanden ist.

**MQReconnectTimeout**

Die maximale Dauer (in Sekunden), während der die Funktion für die automatische Wiederherstellung der Clientverbindung in einem Client versucht, die Verbindung wiederherzustellen. Der Standardwert ist 1800 Sekunden (30 Minuten).

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients sowie von verwalteten .NET-Clients gelesen werden.

IBM MQ classes for JMS-Clients können mithilfe der Verbindungsfactory-Eigenschaft CLIENTRECONNECTTIMEOUT ein Zeitlimit für die Verbindungswiederherstellung angeben. Der Standardwert für diese Eigenschaft ist 1800 Sekunden (30 Minuten).

IBM MQ classes for XMS .NET-Clients können einen Zeitlimitwert für die Verbindungswiederholung mithilfe der folgenden Eigenschaften angeben:

- Die Eigenschaft CLIENTRECONNECTTIMEOUT einer Verbindungsfactory. Der Standardwert für diese Eigenschaft ist 1800 Sekunden (30 Minuten). Diese Eigenschaft ist nur für den verwalteten Modus gültig.
- Die Eigenschaft XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT. Der Standardwert für diese Eigenschaft ist 1800 Sekunden (30 Minuten). Diese Eigenschaft ist nur für den verwalteten Modus gültig.

**ServerConnectionParms**

ServerConnectionParameter entsprechen der Umgebungsvariablen *MQSERVER* und geben die Position des IBM MQ -Servers und die zu verwendende Übertragungsmethode an.

Dieses Attribut kann von C-, nicht verwalteten .NET-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Das Attribut "ServerConnectionParms" definiert nur einen einfachen Kanal. Sie können ihn nicht verwenden, um einen TLS-Kanal oder einen Kanal mit Kanalexits zu definieren. Es handelt sich um eine Zeichenfolge im Format *ChannelName*/*TransportType*/*ConnectionName*, *ConnectionName* muss ein vollständig qualifizierter Netzname sein. *ChannelName* darf nicht den Schrägstrich (/) enthalten, da dieses Zeichen zum Trennen des Kanalnamens, des Transporttyps und des Verbindungsnamens verwendet wird.

Wenn "ServerConnectionParms" zum Definieren eines Clientkanals verwendet wird, wird eine maximale Nachrichtenlänge von 100 MB verwendet. Daher ist die maximale Nachrichtengröße, die für den Kanal wirksam ist, der im SVRCONN-Kanal auf dem Server angegebene Wert.

Beachten Sie, dass nur eine einzige Clientkanalverbindung hergestellt werden kann. Wenn Sie z. B. zwei Einträge haben:

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

wird nur die zweite verwendet.

Geben Sie *ConnectionName* als durch Kommas getrennte Liste mit Namen für den angegebenen Transporttyp an. Im Allgemeinen ist nur ein Name erforderlich. Sie können mehrere *hostnames* bereitstellen, um mehrere Verbindungen mit den gleichen Eigenschaften zu konfigurieren. Die Verbindungen werden in der Reihenfolge versucht, in der sie in der Verbindungsliste angegeben sind, bis eine Verbindung erfolgreich hergestellt wurde. Wenn keine Verbindung erfolgreich ist, beginnt der Client erneut zu verarbeiten. Verbindungslisten sind eine Alternative zu WS-Manager-Gruppen, um Verbindungen für wiederverbindbare Clients zu konfigurieren.

**Put1DefaultAlwaysSync = NO (Standardwert) | YES**

Steuert das Verhalten des Funktionsaufrufs MQPUT1 mit der Option MQPMO_RESPONSE_AS_Q_DEF.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

**NEIN**

Wenn MQPUT1 mit MQPMO_SYNCPOINT festgelegt ist, verhält es sich wie MQPMO_ASYNC_RESPONSE. Wenn MQPUT1 mit MQPMO_NO_SYNCPOINT festgelegt ist, verhält es sich ebenfalls wie MQPMO_SYNC_RESPONSE. Dies ist der Standardwert.

**JA**

MQPUT1 verhält sich so, als ob MQPMO_SYNC_RESPONSE gesetzt ist, unabhängig davon, ob MQPMO_SYNCPOINT oder MQPMO_NO_SYNCPOINT festgelegt ist.

**PasswordProtection = Kompatibel (Standardwert) |always|optional**

Ab IBM MQ 8.0können Authentifizierungsnachweise, die von IBM MQ client -Anwendungen angegeben werden, wenn sie eine Verbindung zu einem Warteschlangenmanager herstellen, mithilfe der IBM MQ MQCSP-Kennwortschutzfunktion geschützt werden, wenn die Verbindung keine TLS-Verschlüsselung verwendet.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Der MQCSP-Kennwortschutz ist für Test-und Entwicklungszwecke nützlich, da die Verwendung des MQCSP-Kennwortschutzes einfacher ist, als die TLS-Verschlüsselung zu konfigurieren, aber nicht als sicher.

Weitere Informationen zum Schutz von Berechtigungsnachweisen in der MQCSP-Struktur und zu den Werten, die für dieses Attribut festgelegt werden können, enthält der Abschnitt MQCSP-Kennwortschutz.

**Zugehörige Tasks**

IBM MQ-MQI-Anwendungen mit Warteschlangenmanagern verbinden

## Multi ClientExitPath, Zeilengruppe der Clientkonfigurationsdatei

Verwenden Sie die Zeilengruppe "ClientExitPath", um die Standardpositionen von Kanalexits auf dem Client anzugeben.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die ClientExitPath-Zeilengruppe aufgenommen werden:

**ExitsDefaultPath = *string***
Gibt die Position von 32-Bit-Kanalexits für Clients an.

Dieses Attribut kann von C, nicht verwalteten .NET, verwalteten .NET-, verwalteten XMS .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden. IBM MQ classes for Java- und IBM MQ classes for JMS-Clients verwenden dieses Attribut, um 32-Bit-Kanalexits zu lokalisieren, die nicht in Java geschrieben sind.

**ExitsDefaultPath64 = *string***
Gibt die Position der 64-Bit-Kanalexits für Clients an.

Dieses Attribut kann von C, nicht verwalteten .NET, verwalteten .NET-, verwalteten XMS .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden. IBM MQ classes for Java- und IBM MQ classes for JMS-Clients verwenden dieses Attribut, um 64-Bit-Kanalexits zu lokalisieren, die nicht in Java geschrieben sind.

**JavaExitsClassPath = *string***
Die Werte, die dem Klassenpfad hinzugefügt werden sollen, wenn ein Java-Exit ausgeführt wird. Dies wird von Exits in einer anderen Sprache ignoriert.

Dieses Attribut kann von IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden.

In der JMS-Konfigurationsdatei erhält der Java-ExitsClassPath-Name das standardmäßige Präfix "com.ibm.mq.cfg." Präfix und dieser vollständige Name werden auch in der Systemeigenschaft IBM MQ verwendet.

## Multi Zeilengruppe 'Connection' der Clientkonfigurationsdatei

Mit der Zeilengruppe 'Connection' können Sie einen Anwendungsnamen angeben.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Das folgende Attribut kann in die Zeilengruppe 'Connection' eingeschossen werden:

**ApplName = `ExampleApplname`**
Sie können in der Clientkonfigurationsdatei einen Anwendungsname angeben.

Dieses Attribut kann von C-Clients und von nicht verwalteten .NET-Clients verwendet werden.

## Multi Zeilengruppe 'JMQI' der Clientkonfigurationsdatei

Verwenden Sie die Zeilengruppe 'JMQI', um Konfigurationsparameter für die Java Message Queuing Interface (JMQI) anzugeben, die von IBM MQ classes for Java und IBM MQ classes for JMS verwendet wird.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Das folgende Attribut kann in die Zeilengruppe 'JMQI' eingeschlossen werden:

**useMQCSPauthentication = NO | YES**
Steuert, ob IBM MQ classes for Java- und IBM MQ classes for JMS-Anwendungen bei der Authentifizierung mit einem WS-Manager den Kompatibilitätsmodus oder den MQCSP-Authentifizierungsmodus verwenden sollen.

Dieses Attribut kann von IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden.

Dieses Attribut kann die folgenden Werte aufweisen:

**NEIN**
Verwenden Sie den Kompatibilitätsmodus, wenn Sie sich mit einem Warteschlangenmanager authentifizieren. Dies ist in Versionen vor IBM MQ 9.3.0 der Standardwert.

**JA**
Verwenden Sie den MQCSP-Authentifizierungsmodus beim Authentifizieren mit einem Warteschlangenmanager. V 9.3.0 Dies ist ab IBM MQ 9.3.0 der Standardwert.

Es gibt mehrere andere Möglichkeiten zum Festlegen des Authentifizierungsmodus, der Vorrang vor dem Wert des Attributs **useMQCSPauthentication** hat. Weitere Informationen zum Kompatibilitätsmodus und dem MQCSP-Authentifizierungsmodus finden Sie im Abschnitt Verbindungsauthentifizierung mit dem Java-Client.

## Windows LU62-, NETBIOS-und SPX-Zeilengruppen in der Clientkonfigurationsdatei

Diese Zeilengruppen kann nur auf Windows-Systemen zur Angabe von Konfigurationsparametern für die angegebenen Netzprotokolle verwendet werden.

### LU62, Zeilengruppe

Verwenden Sie die Zeilengruppe LU62, um die SNA LU 6.2-Protokollkonfigurationsparameter anzugeben. Die folgenden Attribute können in diese Zeilengruppe aufgenommen werden:

**Library1 = *DLLName* | WCPIC32**
Der Name der APPC-DLL.

**Library2 = *DLLName* | WCPIC32**
Wie Library1, wird verwendet, wenn der Code in zwei separaten Bibliotheken gespeichert ist.

**TP-Name**
Der Name des TP-Namens, der auf dem fernen Standort gestartet werden soll.

### NETBIOS-Zeilengruppe

Verwenden Sie die NETBIOS-Zeilengruppe, um die Konfigurationsparameter für das NetBIOS-Protokoll anzugeben. Die folgenden Attribute können in diese Zeilengruppe aufgenommen werden:

**AdapterNum = *number* | 0**
Die Nummer des LAN-Adapters.

**Library1 = *DLLName* | NETAPI32**
Der Name der NetBIOS-DLL.

**LocalName = *name***
Der Name, unter dem dieser Computer im LAN bekannt ist.

Dieser entspricht der Umgebungsvariablen MQNAME.

**NumCmds = *number* | 1**
Gibt an, wie viele Befehle zugeordnet werden sollen.

**NumSess = *number* | 1**
Gibt an, wie viele Sitzungen zugeordnet werden sollen.

## SPX-Zeilengruppe

Verwenden Sie die SPX-Zeilengruppe, um SPX-Protokollkonfigurationsparameter anzugeben. Die folgenden Attribute können in diese Zeilengruppe aufgenommen werden:

**BoardNum = *number* | 0**
Die LAN-Adapternummer.

**KeepAlive = JA | NEIN**
Schalten Sie die KeepAlive-Funktion ein oder aus.

KeepAlive = YES bewirkt, dass SPX in regelmäßigen Abständen überprüft, ob das andere Ende der Verbindung noch verfügbar ist. Ist dies nicht der Fall, wird der Kanal geschlossen.

**Library1 = *DLLName* | WSOCK32.DLL**
Der Name der SPX-DLL.

**Library2 = *DLLName* | WSOCK32.DLL**
Wie Library1, wird verwendet, wenn der Code in zwei separaten Bibliotheken gespeichert ist.

**Socket = *number* | 5E86**
Die SPX-Socket-Nummer in Hexadezimalschreibweise.

## Multi MessageBuffer-Zeilengruppe der Clientkonfigurationsdatei

Verwenden Sie die Zeilengruppe 'MessageBuffer', um Informationen zu Nachrichtenpuffern anzugeben.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in der Zeilengruppe 'MessageBuffer' enthalten sein:

**MaximumSize = *integer* | 1**
Größe (in Kilobyte) des Read-Ahead-Puffers im Bereich von 1 bis 999 999.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Die folgenden Sonderwerte sind vorhanden:

**-1**
Der Client bestimmt den entsprechenden Wert.

**0**
Vorauslesen ist für den Client inaktiviert.

**PurgeTime = *integer* | 600**
Intervall (in Sekunden), nach dem Nachrichten, die im Puffer für die Lesepuffer (Read-ahead-Puffer) bleiben, gelöscht

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Wenn die Clientanwendung Nachrichten auf der Basis von MsgId oder CorrelId auswählt, ist es möglich, dass der Read-Ahead-Puffer Nachrichten enthält, die mit einer zuvor angeforderten MsgId oder CorrelId an den Client gesendet wurden. Diese Nachrichten würden dann in den Read-Ahead-Puffer gestellt, bis ein MQGET-Aufruf mit einer entsprechenden MsgId oder CorrelId ausgegeben wird. Sie können Nachrichten aus dem Read-Ahead-Puffer löschen, indem Sie PurgeTime festlegen. Alle Nachrichten, die länger als das Bereinigungsintervall im Read-Ahead-Puffer geblieben sind, werden automatisch gelöscht. Diese Nachrichten wurden bereits aus der Warteschlange auf dem Warteschlangenmanager entfernt, so dass sie verloren gehen, wenn sie nicht durchsucht werden.

Der gültige Bereich liegt im Bereich von 1 bis 999 999 Sekunden oder der Sonderwert 0, d. e. es findet keine Bereinigung statt.

**UpdatePercentage = *integer* | -1**

Der Prozentwert für die Aktualisierung im Bereich von 1-100, der bei der Berechnung des Schwellenwerts verwendet wird, um zu ermitteln, wann eine Clientanwendung eine neue Anforderung an den Server stellt. Der Sonderwert -1 gibt an, dass der Client den entsprechenden Wert bestimmt.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Der Client sendet in regelmäßigen Abständen eine Anforderung an den Server, die angibt, wie viele Daten die Clientanwendung verbraucht hat. Eine Anforderung wird gesendet, wenn die Anzahl der Byte ( *n*), die vom Client über MQGET-Aufrufe abgerufen werden, den Schwellenwert *T*überschreitet. *n* wird jedes Mal, wenn eine neue Anforderung an den Server gesendet wird, auf null zurückgesetzt.

Der Schwellenwert T wird wie folgt berechnet:

```
T = Upper - Lower
```

Der obere Wert entspricht der im Attribut *MaximumSize* angegebenen Größe des Lesepuffers (in Kilobyte). Der Standardwert ist 100 Kb.

Der untere Bereich ist kleiner als der obere und wird durch das Attribut *UpdatePercentage* angegeben. Dieses Attribut ist eine Zahl im Bereich von 1 bis 100 und hat einen Standardwert von 20. Der untere wird wie folgt berechnet:

```
Lower = Upper x UpdatePercentage / 100
```

**Beispiel 1:**

Die Attribute MaximumSize und UpdatePercentage nehmen die Standardwerte von 100 Kb und 20 Kb an.

Der Client ruft MQGET auf, um eine Nachricht abzurufen, und wiederholt dies wiederholt. Dies wird so lange fortgesetzt, bis MQGET n Byte belegt hat.

Berechnung verwenden

```
T = Upper - Lower
```

T ist (100-20) = 80 Kb.

Wenn MQGET-Aufrufe also 80 Kb aus einer Warteschlange entfernt haben, stellt der Client automatisch eine neue Anforderung aus.

**Beispiel 2:**

Die MaximumSize-Attribute nimmt ihren Standardwert 100 Kb und der Wert 40 für 'UpdatePercentage' (UpdatePercentage) ausgewählt.

Der Client ruft MQGET auf, um eine Nachricht abzurufen, und wiederholt dies wiederholt. Dies wird so lange fortgesetzt, bis MQGET n Byte belegt hat.

Berechnung verwenden

```
T = Upper - Lower
```

T ist (100-40) = 60 Kb

Wenn MQGET-Aufrufe also 60 Kb aus einer Warteschlange entfernt haben, stellt der Client automatisch eine neue Anforderung ab. Dies ist früher als in BEISPIEL 1, wo die Standardwerte verwendet wurden.

Die Auswahl eines größeren Schwellenwerts *T* neigt daher dazu, die Häufigkeit zu verringern, mit der Anforderungen vom Client an den Server gesendet werden. Umgekehrt erhöht die Auswahl eines kleineren Schwellenwerts *T* die Häufigkeit von Anforderungen, die vom Client an den Server gesendet werden.

Die Auswahl eines großen Schwellenwerts *T* kann jedoch bedeuten, dass die Leistungssteigerung von Read-Ahead-Wert reduziert wird, da die Wahrscheinlichkeit, dass der Read-Ahead-Puffer leer wird, erhöht werden kann. Wenn dies geschieht, muss ein MQGET-Aufruf möglicherweise anhalten und darauf warten, dass Daten vom Server eintreffen.

## Multi Zeilengruppe für PreConnect der Clientkonfigurationsdatei

Verwenden Sie die PreConnect-Zeilengruppe, um den PreConnect-Exit in der `mqclient.ini`-Datei zu konfigurieren.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die Zeilengruppe PreConnect aufgenommen werden:

**Daten = *user_data***
Dieses Attribut gibt die Benutzerdaten an, die an den Preconnect-Exit übergeben werden. Die Daten, die an den Preconnect-Exit übergeben werden, sind spezifisch für die Implementierung des von Ihnen verwendeten Preconnect-Exits und die Daten, die an den Preconnect-Exit übergeben werden sollen.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Dieses Attribut kann beispielsweise verwendet werden, um die URL des Repositorys anzugeben, in dem Verbindungsdefinitionen gespeichert werden, wie z. B. bei Verwendung eines LDAP-Servers:

```
Data = ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
```

**Funktion = *myFunc***
Der Name des funktionalen Eingangspunkts in der Bibliothek, die den PreConnect-Exit-Code enthält.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Die Funktionsdefinition entspricht dem PreConnect-Exit-Prototyp MQ_PRECONNECT_EXIT .

Die maximale Länge dieses Feldes ist MQ_EXIT_NAME_LENGTH.

**Modul = *myMod***
Der Name des Moduls, das den API-Exit-Code enthält.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Wenn dieses Feld den vollständigen Pfadnamen des Moduls enthält, wird es wie angegeben verwendet.

**Folge = *sequence_number***
Die Sequenz, in der dieser Exit relativ zu anderen Exits aufgerufen wird. Ein Exit mit einer niedrigen Folgenummer wird vor einem Exit mit einer höheren Folgenummer aufgerufen. Es ist nicht erforderlich, dass die Folgenummerierung von Ausgängen stetig ist. Eine Folge von 1, 2, 3 hat das gleiche Ergebnis wie eine Folge von 7, 42, 1096. Dieses Attribut ist ein numerischer Wert ohne Vorzeichen.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Innerhalb der `mqclient.ini`-Datei können mehrere PreConnect-Zeilengruppen definiert werden. Die Verarbeitungsreihenfolge der einzelnen Exit wird durch das Attribut "Sequence" der Zeilengruppe festgelegt.

**Zugehörige Tasks**

Verbindungsdefinitionen unter Verwendung eines Vorverbindungsexits aus einem Repository referenzieren

## Multi Zeilengruppe 'Security' der Clientkonfigurationsdatei

Mit der Zeilengruppe 'Security' wird AMS für Clientverbindungen zu einem Warteschlangenmanager aktiviert oder inaktiviert.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Das folgende Attribut kann in der Zeilengruppe 'Security' enthalten sein:

**DisableClientAMS = NO|YES**

Das Attribut DisableClientAMS ermöglicht Ihnen die Inaktivierung von IBM MQ Advanced Message Security (AMS), wenn Sie einen IBM MQ -Client verwenden, um eine Verbindung zu einem Warteschlangenmanager aus einer früheren Version des Produkts herzustellen, und ein Fehler `2085 (MQRC_UNKNOWN_OBJECT_NAME)` gemeldet wird.

IBM MQ Advanced Message Security (AMS) wird automatisch in einem IBM MQ -Client aktiviert, sodass der Client standardmäßig versucht, die Sicherheitsrichtlinien für Objekte auf dem Warteschlangenmanager zu überprüfen.

In den folgenden Beispielen wird die Verwendung des Attributs DisableClientAMS gezeigt:

- So inaktivieren Sie AMS:

```
Security:
DisableClientAMS=Yes
```

- So aktivieren Sie AMS:

```
Security:
DisableClientAMS=No
```

Dieses Attribut dann von C-, IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden.

V 9.3.0 V 9.3.0 **MQIInitialKeyFile = *Pfadname***

Der vollständige Pfad und Name der Datei mit dem ursprünglichen Schlüssel, der zum Verschlüsseln der vom Client bereitgestellten Berechtigungsnachweise verwendet wurde. Der Anfangsschlüssel muss angegeben werden, wenn eine Anfangsschlüsseldatei angegeben wurde, als die Kennphrase des Schlüsselrepositorys mit dem Dienstprogramm **runmqicred** verschlüsselt wurde.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

**Zugehörige Tasks**

Advanced Message Security auf dem Client inaktivieren

## Multi SSL-Zeilengruppe der Clientkonfigurationsdatei

Verwenden Sie die SSL-Zeilengruppe, um Informationen über die Verwendung von TLS anzugeben.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizei-

lengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die SSL-Zeilengruppe eingeschlossen werden:

V 9.3.0 **OutboundSNI = CHANNEL | HOSTNAME**

Wenn **OutboundSNI** auf KANALgesetzt ist, setzen SNI-fähige Clients SNI auf den Namen des IBM MQ -Zielkanals des fernen Systems, wenn eine TLS-Verbindung eingeleitet wird.

Wenn dieses Attribut auf HOSTNAME gesetzt ist, legen SNI-fähige Clients als SNI-Header den Hostnamen fest, wodurch ausgehende Verbindungsanforderungen das Standardzertifikat des fernen Warteschlangenmanagers während des TLS-Handshakes empfangen und somit keine kanalweisen Zertifikate verwendet werden können.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden.

Der Java/JMS-Client berücksichtigt bei Eigenschaftswerten die Groß-/Kleinschreibung, deshalb sollten die Werte YES/NO in Großschreibung angegeben werden.

Ab IBM MQ 9.3.0wurde der IBM MQ verwaltete .NET -Client aktualisiert, um SERVERNAME auf den entsprechenden Hostnamen zu setzen, wenn die Eigenschaft **OutboundSNI** auf HOSTNAMEgesetzt ist, was einem IBM MQ verwalteten .NET -Client ermöglicht, über Red Hat OpenShift -Routeneine Verbindung zu einem Warteschlangenmanager herzustellen.

**Anmerkung:** Wenn eine Anwendung mit der **OutboundSNI** -Einstellung HOSTNAME eine Verbindung zu einem Kanal mit einer konfigurierten Zertifikatsbezeichnung herstellt, wird die Anwendung mit MQRC_SSL_INITIALIZATION_ERROR abgelehnt und eine Nachricht AMQ9673 in den Fehlerprotokollen des Warteschlangenmanagers ausgegeben.

**AllowOutboundSNI = YES (Standardwert) | NEIN**

Wenn diese Option aktiviert ist, legen SNI-fähige Clients als SNI den Namen des IBM MQ-Zielkanals für das ferne System fest, wenn eine TLS-Verbindung eingeleitet wird. Wenn dieses Attribut auf NO gesetzt ist, legen SNI-fähige Clients nicht den SNI-Header fest, wodurch ausgehende Verbindungsanforderungen das Standardzertifikat des fernen Warteschlangenmanagers während dem TLS-Handshake empfangen und somit keine kanalweisen Zertifikate verwendet werden können.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden.

Der Java/JMS-Client berücksichtigt bei Eigenschaftswerten die Groß-/Kleinschreibung, deshalb sollten die Werte YES/NO in Großschreibung angegeben werden.

**Achtung:** V 9.3.0 Deprecated Ab IBM MQ 9.3.0 ist die Eigenschaft **AllowOutboundSNI** veraltet und nur zu Zwecken der Abwärtskompatibilität verfügbar.

**AllowOutboundSNI** set to JA bietet dieselbe Funktion wie **OutboundSNI** set to KANAL, während **AllowOutboundSNI** set to NEIN dieselbe Funktion wie **OutboundSNI** set to HOSTNAME bereitstellt.

Wenn die SSL-Zeilengruppe sowohl das Attribut **AllowOutboundSNI** als auch das Attribut **OutboundSNI** enthält, hat die Einstellung von **OutboundSNI** Vorrang.

IBM i ALW **AllowTLSV13 =Y | YES | T | TRUE (Standardwert) | N | NO | F | FALSE**

Gibt an, ob ein Warteschlangenmanager die TLS 1.3-CipherSpecs verwenden kann (siehe CipherSpecs aktivieren).

Dieses Attribut kann von C/C++-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

- Y (Standardwert), YES (Standardwert) T (Standardwert) oder TRUE (Standardwert): Aktiviert TLS 1.3 , damit der Warteschlangenmanager die TLS 1.3 CipherSpecsverwenden kann.
- N, NO, F oder FALSE: Inaktiviert TLS 1.3, was bedeutet, dass der Warteschlangenmanager die TLS 1.3-CipherSpecs nicht verwenden kann.

**Anmerkung:** Bei Verwendung des MQI-Clients wird der Wert von **AllowTLSV13** abgeleitet, sofern er nicht explizit in der SSL-Zeilengruppe der Datei „SSL-Zeilengruppe der Clientkonfigurationsdatei" auf Seite 195 angegeben ist, die von der Anwendung verwendet wird. Weitere Informationen finden Sie unter IBM MQ MQI-Client und TLS 1.3.

**CDPCheckExtensions = YES|NO (Standardwert)**

CDPCheckExtensions gibt an, ob TLS-Kanäle in diesem Warteschlangenmanager versuchen, CDP-Server zu überprüfen, die in den Zertifikatserweiterungen des CrlDistributionPoint-Zertifikats benannt sind.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

- YES (Standardwert): TLS-Kanäle versuchen, CDP-Server zu überprüfen, um festzustellen, ob ein digitales Zertifikat widerrufen wurde.
- NO : TLS-Kanäle versuchen nicht, die CDP-Server zu überprüfen. Dieser Wert stellt den Standardwert dar.

**CertificateLabel = *string***
Die Zertifikatsbezeichnung der Kanaldefinition.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Weitere Informationen finden Sie unter Zertifikatsbezeichnung (CERTLABL) .

**CertificateValPolicy = *string***
Bestimmt den Typ der verwendeten Zertifikatsprüfung.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

**ANY**
Verwenden Sie eine beliebige Zertifikatvalidierungsrichtlinie, die von der zugrunde liegenden Secure Sockets Library unterstützt wird Dies ist die Standardeinstellung.

**RFC5280**
Verwenden Sie nur die Zertifikatsprüfung, die mit dem Standard RFC 5280 kompatibel ist.

**ClientRevocationChecks = REQUIRED | OPTIONAL | DISABLED**
Legt fest, wie die Überprüfung der Zertifikatswiderrufprüfung konfiguriert wird, wenn der Clientverbindungsaufruf einen TLS-Kanal verwendet. Siehe auch **OCSPAuthentication**.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

**ERFORDERLICH (Standardwert)**
Es wird versucht, die Zertifikatswiderrufkonfiguration aus der CCDT zu laden und die Widerrufsprüfung wie konfiguriert auszuführen. Wenn die CCDT-Datei nicht geöffnet werden kann oder es nicht möglich ist, das Zertifikat zu prüfen (z. B. weil ein OCSP-oder CRL-Server nicht verfügbar ist), schlägt der MQCONN-Aufruf fehl. Es wird keine Widerrufsprüfung durchgeführt, wenn die CCDT keine Widerrufkonfiguration enthält, dies jedoch nicht dazu führt, dass der Kanal fehlschlägt.

Windows Auf Windows-Systemen können Sie auch Active Directory für die CRL-Widerrufsprüfung verwenden. Sie können Active Directory nicht für die OCSP-Widerrufsprüfung verwenden.

Wenn Sie MQSCO oder CCDT verwenden, ist die Verbindung erfolgreich. Wenn keine CCDT-Datei vorhanden ist und MQSCO ebenfalls nicht bereitgestellt wird, schlägt die Verbindung mit dem Ursachencode 2059 fehl und das Fehlerprotokoll meldet `AMQ9518E: Datei '/var/mqm/AMQCLCHL.TAB' nicht gefunden`.

**OPTIONAL**
Wie für REQUIRED , aber wenn es nicht möglich ist, die Zertifikatswiderrufkonfiguration zu laden, schlägt der Kanal nicht fehl.

**INAKTIVIERT**
Es wird kein Versuch unternommen, die Zertifikatswiderrufkonfiguration aus der CCDT zu laden, und es wird keine Überprüfung der Zertifikatswiderrufsüberprüfung durchgeführt.

**Anmerkung:** Wenn Sie MQCONNX anstelle von MQCONN-Aufrufen verwenden, können Sie die Authentifizierungsinformationen (MQAIR) über das MQSCO angeben. Das Standardverhalten mit MQCONNX ist daher nicht zu scheitern, wenn die CCDT-Datei nicht geöffnet werden kann, sondern angenommen wird, dass Sie eine MQAIR-Datei bereitstellen (auch wenn Sie dies nicht tun).

**EncryptionPolicySuiteB = *string***
Legt fest, ob ein Kanal eine Suite-B-kompatible Verschlüsselung verwendet und welcher Grad der Stärke verwendet werden soll.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

**KEINE**
Die Suite-B-kompatible Verschlüsselung wird nicht verwendet. Dies ist die Standardeinstellung.

**128_BIT,192_BIT**
Setzt die Sicherheitsstärke sowohl auf 128-Bit-als auch auf 192-Bit-Stufen.

**128_BIT**
Setzt die Sicherheitsstärke auf 128-Bit-Ebene.

**192_BIT**
Setzt die Sicherheitsstärke auf 192-Bit-Ebene.

V 9.3.0 ALW **EnvironmentScope=PROCESS|CONNECTION**
Steuert, ob IBM MQ eine einzelne IBM Global Security Kit (GSKit) -Umgebung für den gesamten Prozess oder eine GSKit -Umgebung pro Verbindung verwendet.

Dieses Attribut kann von C-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

**PROCESS**
Eine einzelne GSKit -Umgebung wird für mehrere Verbindungen verwendet, die vom Prozess erstellt werden. Die Verwendung dieser Einstellung bedeutet, dass die Änderungen des TLS-Keystores erst dann verfügbar sind, wenn alle aktiven TLS-Verbindungen innerhalb des Prozesses gestoppt wurden.

Dies ist der Standardwert.

**CONNECTION**
Für jede Verbindung innerhalb desselben Prozesses wird eine GSKit -Umgebung erstellt. Dies bedeutet, dass die TLS-Keystore-Änderungen sofort von allen neuen TLS-Verbindungen, die vom Prozess gestartet wurden, übernommen werden.

**Warnung:** Wenn Sie diesen Betriebsmodus aktivieren, verwenden Anwendungen zusätzliche CPU-und Speicherressourcen, um jede GSKit -Umgebung zu erstellen. Dieser Ressourcenverbrauch erhöht sich bei jeder zusätzlichen gleichzeitigen TLS-Verbindung.

ALW **MinimumRSAKeySize=int**

Gibt die Mindestschlüsselgröße an, die RSA-Zertifikate haben müssen, damit sie akzeptiert werden. Jeder Wert größer-gleich 0 ist zulässig. Falls nichts angegeben ist, wird der Standardwert 1 verwendet.

Dieses Attribut kann von C/C++-Clients gelesen werden.

**OCSPAuthentication = OPTIONAL | REQUIRED | WARN**
Definiert das Verhalten von IBM MQ, wenn OCSP aktiviert ist und die OCSP-Widerrufsprüfung nicht in der Lage ist, den Status des Zertifikatswiderrufs zu ermitteln. Siehe auch **ClientRevocationChecks**.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Für dieses Attribut sind die folgenden Werte möglich:

**OPTIONAL**
Jedes Zertifikat mit einem Widerrufsstatus, das durch die OCSP-Prüfung nicht festgestellt werden kann, wird akzeptiert, und es wird keine Warnung oder Fehlernachricht generiert. Die SSL-oder TLS-Verbindung wird so fortgesetzt, als wäre keine Widerrufsüberprüfung durchgeführt worden.

**ERFORDERLICH**
Die OCSP-Prüfung muss für jedes überprüfte SSL-oder TLS-Zertifikat zu einem endgültigen Widerrufsergebnis führen. Jedes SSL-oder TLS-Zertifikat mit einem Widerrufsstatus, der nicht geprüft werden kann, wird mit einer Fehlernachricht zurückgewiesen. Wenn WS-Manager-SSL-Ereignisnachrichten aktiviert sind, wird eine Nachricht MQRC_CHANNEL_SSL_ERROR mit einem 'Reason-Qualifier' von MQRQ_SSL_HANDSHAKE_ERROR generiert. Die Verbindung ist geschlossen.

Dies ist der Standardwert.

**WARN**
Eine Warnung wird in den Fehlerprotokollen des Warteschlangenmanagers angezeigt, wenn eine OCSP-Widerrufsprüfung nicht in der Lage ist, den Widerrufstatus eines SSL-oder TLS-Zertifikats zu ermitteln. Wenn WS-Manager-SSL-Ereignisnachrichten aktiviert sind, wird eine Nachricht MQRC_CHANNEL_SSL_WARNING mit einem 'ReasonQualifier' von MQRQ_SSL_UNKNOWN_REVOCATION generiert. Die Verbindung darf bestehen bleiben.

**OCSPCheckExtensions = YES | NO**
Steuert, ob IBM MQ auf die AuthorityInfoAccess-Zertifikatserweiterungen einwirkt.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Wenn der Wert auf NOgesetzt ist, ignoriert IBM MQ die AuthorityInfoAccess-Zertifikatserweiterungen und versucht nicht, eine OCSP-Sicherheitsprüfung auszuführen. Der Standardwert ist YES .

ALW **OCSPTimeout = *Anzahl***
Die Anzahl der Sekunden, die bei der Ausführung einer Widerrufsprüfung auf einen OCSP-Responder gewartet wird.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Ab IBM MQ 9.3.0wird das Standardzeitlimit von 30 Sekunden verwendet, wenn der Wert 0 festgelegt ist.

Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert von 30 Sekunden verwendet.

ALW **PeerCertChainValidation=*Zeichenfolge***
Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

Die Zeichenfolge kann einen der beiden folgenden Werte haben:

- `Usepeerchain` **[Standardwert]**: Mit der vom Peer bereitgestellten Zertifikatskette können bei der Validierung von Zertifikaten Lücken in der Vertrauenskette überbrückt werden. Eine Ausnahme ist das Stammzertifikat.
- `Truststoreonly` **[Nicht empfohlen]**: Nur Zertifikate im Truststore werden für die Validierung des Peerzertifikats verwendet.

**SSLCryptoHardware = *string***
Legt die Parameterzeichenfolge fest, die erforderlich ist, um die auf dem System vorhandene Verschlüsselungshardware PKCS #11 zu konfigurieren.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Geben Sie eine Zeichenfolge im folgenden Format an: `GSK_PKCS11 = driver path and filename;token label;token password;symmetric cipher setting;`

Beispiel: `GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;passw0rd;SYMMETRIC_CIPHER_ON`

Der Treiberpfad ist ein absoluter Pfad zur gemeinsam genutzten Bibliothek, die Unterstützung für die PKCS #11-Karte bietet. Der Name der Treiberdatei ist der Name der gemeinsam genutzten Bibliothek. Ein Beispiel für den Wert, der für den PKCS-#11-Treiberpfad und den Dateinamen erforderlich ist, ist `/usr/lib/pkcs11/PKCS11_API.so`. Für den Zugriff auf symmetrische Verschlüsselungsoperationen über GSKitgeben Sie den Parameter für symmetrische Verschlüsselungseinstellungen an. Der Wert dieses Parameters lautet entweder:

**SYMMETRIC_CIPHER_OFF**
Es werden keine symmetrischen Verschlüsselungsoperationen aufgerufen. Dies ist die Standardeinstellung.

**SYMMETRIC_CIPHER_ON**
Zugriff auf Operationen zur symmetrischen Verschlüsselung.

Linux AIX Wenn Sie die verschiedenen Komponenten der Zeichenfolge angeben, müssen Sie die Semikolon-Zeichen mithilfe des Backslash-Zeichens verlassen, da das Semikolon als Kommentar behandelt wird. Beispiel: `'\;'`

V 9.3.0 Sie sollten das Tokenkennwort, das in der Attributzeichenfolge von **SSLCryptoHardware** enthalten ist, schützen. Weitere Informationen finden Sie unter IBM MQ -Clients verwenden Verschlüsselungshardware .

V 9.3.0 Um verschlüsselte Kennwörter zu verarbeiten, gibt es jetzt keine Begrenzung auf die Länge der Zeichenfolge.

Standardmäßig erfolgt keine Angabe. Wenn Sie eine Zeichenfolge angeben, die sich nicht im richtigen Format befindet, wird ein Fehler generiert.

**SSLCryptoHardwareKeyFile =** ***Pfadname***
Der vollständige Pfad und Name der Datei, die den Anfangsschlüssel enthält, der zum Verschlüsseln des Kennworts in der Konfigurationszeichenfolge der PKCS #11 -Verschlüsselungshardware verwendet wurde, die mit dem Attribut **SSLCryptoHardware** angegeben wird Der Anfangsschlüssel muss angegeben werden, wenn eine Anfangsschlüsseldatei angegeben wurde, als das Kennwort in der Konfigurationszeichenfolge der Verschlüsselungshardware mit dem Befehl **runp11cred** verschlüsselt wurde. Weitere Informationen finden Sie unter IBM MQ-Clients mit Verschlüsselungshardware.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

**SSLFipsRequired = JA | NEIN**
Gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen, wenn die Verschlüsselung in IBM MQ ausgeführt wird.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

Wenn eine Verschlüsselungshardware konfiguriert ist, werden die vom Hardwareprodukt bereitgestellten Verschlüsselungsmodule verwendet. Diese können-je nach dem verwendeten Hardwareprodukt-möglicherweise FIPS-zertifiziert sein, die auf eine bestimmte Ebene zertifiziert sind.

**SSLHTTPProxyName =** ***string***
Die Zeichenfolge ist entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers, der von GSKit für OCSP-Prüfungen verwendet wird. Auf die Adresse kann optional eine in Klammern gesetzte Portnummer folgen. Wenn Sie die Portnummer nicht angeben, wird der Standard-HTTP-Port 80 verwendet.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

AIX Für 32-Bit-Clients unter AIX kann die Netzadresse nur eine IPv4-Adresse sein.

Auf anderen Plattformen kann die Netzadresse eine IPv4- oder IPv6-Adresse sein.

Dieses Attribut kann erforderlich sein, wenn z. B. eine Firewall den Zugriff auf die URL des OCSP-Responder verhindert.

ALW **SSLHTTPConnectTimeout = *Anzahl*|0**

Die Anzahl der Sekunde, die beim Ausführen einer Widerrufsprüfung auf die erfolgreiche Herstellung einer Netzverbindung zu einem HTTP-Server gewartet wird.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert 0 (aus) verwendet.

**SSLKeyRepository = *Pfadname***

V 9.3.0 V 9.3.0 Der vollständige Pfad und der Dateiname des Schlüsselrepositorys, das das digitale Zertifikat des Benutzers enthält. Wird die Dateierweiterung nicht angegeben, wird angenommen, dass es sich um `.kdb`handelt.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

V 9.3.0 V 9.3.0 **SSLKeyRepositoryPassword = *Kennphrase***

Die Kennphrase für den Zugriff auf das Schlüsselrepository. Der Wert kann eine einfache Textzeichenfolge oder eine Kennphrase sein, die mit dem Dienstprogramm **`runmqicred`** verschlüsselt wurde.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

**SSLKeyResetCount = *integer* | 0**

Die Anzahl der nicht verschlüsselten Byte, die auf einem TLS-Kanal gesendet und empfangen werden, bevor der geheime Schlüssel neu vereinbart wird.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

Der Wert muss im Bereich von 0 bis 999999999 liegen.

Der Standardwert ist 0, was bedeutet, dass geheime Schlüssel nie neu verhandelt werden.

Wenn Sie einen Wert von 1 bis 32768 angeben, verwenden die TLS-Kanäle einen Rücksetzzähler für geheime Schlüssel von 32768 (32Kb). Dadurch werden überhöhte Schlüsselresets vermieden, die bei kleinen Rücksetzwerten für geheime Schlüssel auftreten würden.

## Multi TCP-Zeilengruppe der Clientkonfigurationsdatei

Verwenden Sie die TCP-Zeilengruppe, um die Konfigurationsparameter des TCP-Netzprotokolls anzugeben.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle für alle IBM MQ MQI client-Konfigurationsdateizeilengruppen finden Sie im Abschnitt Which IBM MQ attributes can be read by each client (Welche IBM MQ-Attribute von jedem Client gelesen werden können).

Die folgenden Attribute können in die TCP-Zeilengruppe aufgenommen werden:

**ClntRcvBuffSize = *number* | 0**

Die Größe des TCP/IP-Empfangspuffers (in Byte), der vom Clientende eines Clientverbindungs-Serververbindungskanals verwendet wird.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Der Wert 0 gibt an, dass die Puffergrößen vom Betriebssystem verwaltet werden, im Gegensatz zu Festlegung der Puffergrößen durch IBM MQ. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

**ClntSndBuffSize = *number* | 0**

Die Größe des TCP/IP-Sendepuffers (in Byte), der vom Clientende eines Clientverbindungs-Serververbindungskanals verwendet wird.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Der Wert 0 gibt an, dass die Puffergrößen vom Betriebssystem verwaltet werden, im Gegensatz zu Festlegung der Puffergrößen durch IBM MQ. Wenn der Wert als null festgelegt wird, werden die Standardwerte des Betriebssystems verwendet. Wenn kein Wert festgelegt ist, wird der IBM MQ-Standardwert (32768) verwendet.

**Connect_Timeout = *number***
Die Anzahl der Sekunden, bevor ein Versuch unternommen wird, das Socket-Zeitlimit zu verbinden.

Wenn **ConnectTimeout** = 0 ist und SOCK_NONBLOCK vor einem asynchronen connect () -Aufruf ausgegeben wird, ist der Aufruf nicht blockiert. Das Standardzeitlimit von 20 Sekunden (CONNECT_WAIT_MAX) gilt für die Überprüfung des Verbindungsstatus.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-Clients gelesen werden.

IBM MQ-Kanalprozesse stellen Verbindungen über nicht blockierende Sockets her. Wenn das andere Ende des Sockets daher nicht bereit ist, wird die Verbindung () sofort mit *EINPROGRESS* oder *EWOULDBLOCK* zurückgegeben. Anschließend gibt es keinen Versuch, die Verbindung wiederherzustellen.

Wenn "Connect_Timeout" auf einen Wert ungleich null gesetzt ist, wartet IBM MQ für den angegebenen Zeitraum auf select(), um den Socket bereit zu machen. Dies erhöht die Erfolgschancen eines nachfolgenden connect () -Aufrufs. Diese Option kann in Situationen nützlich sein, in denen eine Verbindung aufgrund einer hohen Auslastung im Netz einige Wartezeiten erfordern würde.

Es gibt keine Beziehung zwischen den Parametern Connect_Timeout, ClntSndBuffSize und ClntRcvBuffSize.

**IPAddressVersion = MQIPADDR_IPV4 | MQIPADDR_IPV6**
Gibt das IP-Protokoll an, das für eine Kanalverbindung verwendet werden soll.

Dieses Attribut kann von C-, nicht verwalteten .NET-, verwalteten .NET- und verwalteten XMS .NET-Clients gelesen werden.

Sie verfügt über die möglichen Zeichenfolgewerte von MQIPADDR_IPV4 oder MQIPADDR_IPV6. Diese Werte haben dieselbe Bedeutung wie IPV4 und IPV6 in **ALTER QMGR IPADDRV** und die Umgebungsvariable **MQIPADDRV** .

**KeepAlive = JA | NEIN**
Schalten Sie die KeepAlive-Funktion ein oder aus. KeepAlive=YES bewirkt, dass TCP/IP in regelmäßigen Abständen überprüft, ob das andere Ende der Verbindung noch verfügbar ist. Ist dies nicht der Fall, wird der Kanal geschlossen.

Dieses Attribut kann von C-, unverwalteten .NET, IBM MQ classes for Java, IBM MQ classes for JMS, verwalteten .NET, und verwalteten XMS .NET-Klienten gelesen werden.

Windows **Library1 = *DLLName* | WSOCK32**
(Nur Windows) Der Name der TCP/IP-Sockets-DLL.

Dieses Attribut kann von C- und nicht verwalteten .NET-Clients gelesen werden.

## Windows Linux V 9.3.3 Zeilengruppe 'Trace' der Clientkonfigurationsdatei

Mithilfe der Zeilengruppe 'Trace' können Sie den Trace für die IBM MQ .NET -und XMS .NET -Clientbibliotheken aktivieren.

Die folgenden Attribute können in die Zeilengruppe TRACE eingeschlossen werden:

**MQDotnetTraceLevel=0 (Standardwert) |1|2**
Das Attribut **MQDotnetTraceLevel** wird zum Starten oder Stoppen des IBM MQ .NET -Trace verwendet:

- 0: Stoppt die Traceerstellung - dies ist der Standardwert.

- 1: Startet die Traceerstellung mit weniger Details.
- 2: Die Traceerstellung wird mit vollständigen Details gestartet (empfohlen).

Dieses Attribut kann vom verwalteten IBM MQ .NET -Client gelesen werden.

**MQDotnetTracePath =*Pfadname***
Das Attribut **MQDotnetTracePath** verweist auf einen Ordner, in dem die IBM MQ .NET -Tracedateien erstellt werden. Das aktuelle Verzeichnis der Anwendung wird verwendet, wenn der Pfad leer ist oder die Eigenschaft **MQDotnetTracePath** nicht definiert ist.

Dieses Attribut kann vom verwalteten IBM MQ .NET -Client gelesen werden.

**MQDotnetErrorPath =*Pfadname***
Das Attribut **MQDotnetErrorPath** verweist auf einen Ordner, in dem Fehlerprotokolldateien für den IBM MQ .NET -Trace erstellt werden. Das aktuelle Verzeichnis der Anwendung wird verwendet, wenn der Pfad leer oder das Attribut **MQDonetErrorPath** nicht definiert ist.

Dieses Attribut kann vom verwalteten IBM MQ .NET -Client gelesen werden.

**XMSDotnetTraceLevel=0 (Standardwert) |1|2**
Das Attribut **XMSDotnetTraceLevel** wird zum Starten oder Stoppen des XMS .NET -Trace verwendet:

- 0: Stoppt die Traceerstellung - dies ist der Standardwert.
- 1: Startet die Traceerstellung im Basisformat.
- 2: Startet die Traceerstellung mit dem erweiterten Format.

Dieses Attribut kann vom verwalteten XMS .NET -Client gelesen werden.

**XMSDotnetTraceFilePath=*Dateiname***
Wenn für das Attribut **XMSDotnetTraceFilePath** kein Wert festgelegt ist oder wenn dieses Attribut vorhanden ist, aber eine leere Zeichenfolge enthält, wird die Tracedatei für XMS .NET in das aktuelle Verzeichnis gestellt. Wenn Sie die Tracedatei in einem benannten Verzeichnis speichern möchten, geben Sie den Verzeichnisnamen in **XMSDotnettraceFilePath**an, z. B. `XMSDotnetTraceFile-Path="c:\somepath"`.

Dieses Attribut kann vom verwalteten XMS .NET -Client gelesen werden.

**XMSDotnetTraceSpecification =*ComponentName=type=Status***
Das Attribut **XMSDotnetTraceSpecification** gibt den Namen der Klasse an, für die Sie einen Trace erstellen möchten, sowie den Typ des Trace, den Sie für XMS .NETbenötigen.

- *Komponentenname* ist der Name der Klasse, für die Sie einen Trace erstellen möchten. In diesem Namen können Sie ein Platzhalterzeichen (*) verwenden. Beispiel: `*=all=enabled` gibt an, dass Sie für alle Klassen einen Trace erstellen möchten, und `IBM.XMS.impl.*=all=enabled` gibt an, dass Sie nur einen API-Trace anfordern.
- *Typ* kann einer der folgenden Tracetypen sein: all, debug, event, EntryExit.
- *Status* kann aktiviert oder inaktiviert sein.

Sie können mehrere Traceelemente verbinden, indem Sie einen Begrenzer ':' (Doppelpunkt) verwenden.

Dieses Attribut kann vom verwalteten XMS .NET -Client gelesen werden.

**XMSDotnetTraceFileSize=*Größe***
Das Attribut **XMSDotnetTraceFileSize** gibt die maximale Größe der Tracedatei an, die für XMS .NETgeneriert werden soll. Der Standardwert für das Maximum ist 20 MB. Dieser Wert wird als `XMSDotnetTraceFileSize=20`angegeben.

Dieses Attribut kann vom verwalteten XMS .NET -Client gelesen werden.

**XMSDotnetTraceFileNumber=*Anzahl***

Das Attribut **`XMSDotnetTraceFileNumber`** gibt die Anzahl der Tracedateien an, die für XMS .NETaufbewahrt werden sollen. Der Standardwert ist 4 (eine aktive Datei und drei Archivdateien). Die zulässige Mindestanzahl ist 2.

Dieses Attribut kann vom verwalteten XMS .NET -Client gelesen werden.

**Zugehörige Tasks**

Traceerstellung für IBM MQ .NET-Anwendungen mit mqclient.ini

Traceerstellung für XMS .NET-Anwendungen mit mqclient.ini

## Multi Aktivitätstracekonfigurationsdatei mqat.ini

Die Aktivitätstrace-Konfigurationsdatei `mqat.ini`wird verwendet, um das Verhalten des Aktivitätstrace zu konfigurieren. Diese Datei wird verwendet, um die Stufe und Häufigkeit der Tracedaten für die Berichtsaktivität zu definieren. Die Datei bietet außerdem die Möglichkeit, Regeln zu definieren, mit deren Hilfe der Aktivitätstrace auf der Basis des Namens einer Anwendung aktiviert und inaktiviert werden kann.

Die Datei `mqat.ini` folgt demselben Format aus Zeilengruppenschlüssel und Parameter-Wert-Paar wie die Dateien `mqs.ini` und `qm.ini` . Die Datei besteht aus einer einzelnen Zeilengruppe, AllActivityTrace, die verwendet wird, um die Stufe und Häufigkeit der Berichterstellung für Aktivitätstracedaten standardmäßig für alle Aktivitätstraces zu konfigurieren. Die Datei kann auch mehrere ApplicationTrace -Zeilengruppen enthalten. Jede dieser Zeilengruppen definiert eine Regel für das Traceverhalten für eine oder mehrere Verbindungen, basierend auf dem Abgleich des Anwendungsnamens der Verbindungen mit der Regel. Weitere Informationen finden Sie unter Application activity trace und Configuring activity trace behavior using `mqat.ini`.

Der Warteschlangenmanager wendet eine Reihe von Regeln an, um festzulegen, welche Zeilengruppen für eine Verbindung verwendet werden sollen. Optional können Sie die globalen Einstellungen für Tracestufe und Häufigkeit unter der Zeilengruppe AllActivityfür die Verbindungen überschreiben, die mit einer Zeilengruppe ApplicationTrace übereinstimmen. Weitere Informationen finden Sie unter Aktivitätstraceverhalten mit `mqat.ini`konfigurieren.

### Verzeichnispositionen

Linux IBM i AIX Auf AIX and Linux -und IBM i -Systemen befindet sich `mqat.ini` im Datenverzeichnis des Warteschlangenmanagers, das mit der Datei `qm.ini` identisch ist.

Windows Auf Windows-Systemen befindet sich die Datei `mqat.ini` im Verzeichnis `C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name` mit den Warteschlangenmanagerdaten. Benutzer, die Anwendungen ausführen, für die ein Trace durchgeführt wird, benötigen die Berechtigung zum Lesen dieser Datei

## Multi Zeilengruppe AllActivityTrace der Datei mqat.ini

Die Zeilengruppe AllActivityin der Konfigurationsdatei `mqat.ini` gibt die Parameter an, die zum Konfigurieren der Tracestufen für einen Warteschlangenmanager verwendet werden.

Eine einzelne Zeilengruppe AllActivitydefiniert Einstellungen für den Aktivitätstrace, der auf alle IBM MQ -Verbindungen angewendet wird, sofern sie nicht überschrieben werden.

Einzelne Werte in der Zeilengruppe AllActivityTrace können durch spezifischere Informationen in einer ZeilengruppeApplicationTraceüberschrieben werden.

Werden mehrere AllActivity-Tracezeilengruppen angegeben, werden die Werte in der letzten Zeilengruppe verwendet. Parameter, die im ausgewählten AllActivity-Trace fehlen, übernehmen Standardwerte. Parameter und Werte aus vorherigen AllActivity-Tracezeilengruppen werden ignoriert.

**ActivityInterval**

Das Zeitintervall in Sekunden zwischen Tracenachrichten. Der Aktivitätstrace verwendet keinen Zeitgeberthread, daher wird die Tracenachricht nicht zu dem Zeitpunkt geschrieben, zu dem die Zeit

vergeht, sondern geschrieben wird, wenn die erste MQI-Operation nach Ablauf des Zeitintervalls ausgeführt wird. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung disconnects (oder wenn die Anzahl der Aktivitäten erreicht ist). Der Standardwert ist 1.

**ActivityCount**
Die Anzahl der MQI-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung die Verbindung trennt (oder wenn das Aktivitätsintervall abgelaufen ist). Der Standardwert ist 100.

**TraceLevel**
Die Menge der Parameterdetails, für die für jede Operation ein Trace erstellt wird In der Beschreibung der einzelnen Operationen wird angegeben, welche Parameter für die einzelnen Trace-Ebenen angegeben werden. Auf `LOW`, `MEDIUM`oder `HIGH`setzen. Der Standardwert ist `MEDIUM`.

**TraceMessageData**
Die Menge der Nachrichtendaten, für die ein Trace in Byte für MQGET-, MQPUT-, MQPUT1-und Callback-Operationen erstellt wird Der Standardwert ist 0.

**StopOnGetTraceMsg**
Kann auf `ON` oder `OFF`gesetzt werden. Der Standardwert ist `ON`.

**SubscriptionDelivery**
Kann auf `BATCHED` oder `IMMEDIATE`gesetzt werden. Legt fest, ob die Parameter **`ActivityInterval`** und **`ActivityCount`** verwendet werden sollen, wenn eine oder mehrere Aktivitätstrace-Subskriptionen vorhanden sind. Wenn Sie diesen Parameter auf `IMMEDIATE` setzen, werden die Werte **`ActivityInterval`** und **`ActivityCount`** mit den effektiven Werten 1 überschrieben, wenn die Tracedaten eine übereinstimmende Subskription haben. Jeder Aktivitätstracesatz wird nicht mit anderen Datensätzen aus derselben Verbindung ausgeliefert und stattdessen sofort an die Subskription gesendet, ohne dass es zu einer Verzögerung kommt. Die Einstellung `IMMEDIATE` erhöht die Leistungseinbußen bei der Erfassung von Aktivitätstracedaten. Die Standardeinstellung ist `BATCHED`.

**Zugehörige Tasks**
Aktivitätstraceverhalten mit 'mqat.ini' konfigurieren

## Multi Zeilengruppe ApplicationTrace in der Datei mqat.ini

Die Konfigurationsdatei `mqat.ini` kann mehrere Zeilengruppen ApplicationTrace enthalten. Jede dieser Zeilengruppen definiert eine Regel für das Traceverhalten für eine oder mehrere Verbindungen, basierend auf dem Abgleich des Anwendungsnamens der Verbindungen mit der Regel.

Sie können die folgenden Werte für die Zeilengruppe ApplicationTrace definieren:

**Trace**
Aktivitätstraceschalter, der auf `ON` oder `OFF`gesetzt werden kann Der Parameter **`Trace`** ist ein erforderlicher Parameter ohne Standardwert. Sie kann in der anwendungsspezifischen Zeilengruppe verwendet werden, um festzustellen, ob der Aktivitätstrace für den Geltungsbereich der aktuellen Anwendungszeilengruppe aktiv ist. Beachten Sie, dass dieser Wert die Einstellungen **ACTVTRC** und **ACTVCONO** für den Warteschlangenmanager überschreibt.

**ApplName**
Der Parameter **`ApplName`** wird als Zeichenfolge angegeben und ist ein erforderlicher Parameter ohne Standardwert. Dieser Wert wird verwendet, um festzustellen, auf welche Anwendungen die Zeilengruppe 'ApplicationTrace' angewendet wird. Sie wird mit dem Wert **`ApplName`** aus der API-Exit-Kontextstruktur abgeglichen (dies ist äquivalent zu MQMD.PutApplName). Der Inhalt des Werts **`ApplName`** variiert je nach Anwendungsumgebung.

Auf Multiplatforms nur der Dateinamensabschnitt von MQAXC.ApplName wird mit dem Wert in der Zeilengruppe abgeglichen. Zeichen links vom Pfadtrennzeichen rechts werden ignoriert, wenn der Vergleich durchgeführt wird.

Ein einzelnes Platzhalterzeichen (*) kann am Ende des Werts **`ApplName`** verwendet werden, um eine beliebige Anzahl von Zeichen nach diesem Punkt abzugleichen. Wenn der Wert **`ApplName`** auf ein einzelnes Platzhalterzeichen (*) gesetzt ist, stimmt der Wert **`ApplName`** mit allen Anwendungen überein.

IBM i **ApplFunction**

Der Parameter **ApplFunction** wird als Zeichenfolge angegeben. Der Standardwert ist *. Der Wert dieses Parameters wird verwendet, um zu qualifizieren, für welche Anwendungsprogramme die Zeilengruppe ApplicationTrace und der Wert **ApplName** gelten.

Die Zeilengruppe ist optional und nur für IBM i -Warteschlangenmanager gültig. Ein einzelnes Platzhalterzeichen (*) kann am Ende des **ApplName** -Werts verwendet werden, um eine beliebige Anzahl von Zeichen abzugleichen. Beispiel: Eine Zeilengruppe ApplicationTrace , die **ApplName** = * und **ApplFunction** = *AMQSPUT0* angibt, gilt für alle Aufrufe des Programms AMQSPUT0 von einem beliebigen Job.

**ApplClass**

Der Parameter **ApplClass** definiert die Klasse einer Anwendung und kann auf die folgenden Werte gesetzt werden:

- BENUTZER
- Nachrichtenkanalagent
- ALL (Dies ist der Standardwert)

Eine Erläuterung, wie die **AppType** -Werte IBM MQ -Verbindungen entsprechen, finden Sie in Tabelle 3 unter Aktivitätstraceverhalten mit mqat.ini.

Optional können die globalen Einstellungen für Tracestufe und Häufigkeit unter der Zeilengruppe AllActivityfür die Verbindungen überschrieben werden, die einer Zeilengruppe ApplicationTrace entsprechen.

Die folgenden Parameter können unter einer Zeilengruppe ApplicationTrace festgelegt werden. Wenn sie nicht definiert sind, wird der Wert aus den Einstellungen der AllActivity -Tracezeilengruppe übernommen:

**ActivityInterval**

Das Zeitintervall in Sekunden zwischen Tracenachrichten. Der Aktivitätstrace verwendet keinen Zeitgeberthread, daher wird die Tracenachricht nicht zu dem Zeitpunkt geschrieben, zu dem die Zeit vergeht, sondern geschrieben wird, wenn die erste MQI-Operation nach Ablauf des Zeitintervalls ausgeführt wird. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung disconnects (oder wenn die Anzahl der Aktivitäten erreicht ist). Der Standardwert ist 1.

**ActivityCount**

Die Anzahl der MQI-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung die Verbindung trennt (oder wenn das Aktivitätsintervall abgelaufen ist). Der Standardwert ist 100.

**TraceLevel**

Die Menge der Parameterdetails, für die für jede Operation ein Trace erstellt wird In der Beschreibung der einzelnen Operationen wird angegeben, welche Parameter für die einzelnen Trace-Ebenen angegeben werden. Auf LOW, MEDIUModer HIGHsetzen. Der Standardwert ist MEDIUM.

**TraceMessageData**

Die Menge der Nachrichtendaten, für die ein Trace in Byte für MQGET-, MQPUT-, MQPUT1-und Callback-Operationen erstellt wird Der Standardwert ist 0.

**StopOnGetTraceMsg**

Kann auf ON oder OFFgesetzt werden. Der Standardwert ist ON.

**Zugehörige Tasks**

Aktivitätstraceverhalten mit 'mqat.ini' konfigurieren

# Verteilte Warteschlangensteuerung konfigurieren

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### Vorbereitende Schritte

Vor der Lektüre dieses Abschnitts ist es hilfreich, ein Verständnis für Kanäle, Warteschlangen und andere Konzepte zu haben, die in Verteilte Steuerung von Warteschlangen und Clusterneingeführt wurden.

Wenn Sie zwei Warteschlangenmanager verbinden müssen, die sich in unterschiedlichen physischen Netzen befinden oder die über eine Firewall kommunizieren, kann die Verwendung von IBM MQ Internet Pass-Thru die Konfiguration vereinfachen. Weitere Informationen finden Sie unter IBM MQ Internet Pass-Thru.

### Prozedur

- Verwenden Sie die Informationen in den folgenden Unterabschnitten, um Ihre Anwendungen mit der verteilten Steuerung von Warteschlangen zu verbinden:
  - „Verteilte Warteschlangenverfahren in IBM MQ" auf Seite 207
  - „Einführung in die verteilte Warteschlangenverwaltung" auf Seite 230
  - „So senden Sie eine Nachricht an einen anderen Warteschlangenmanager" auf Seite 233
  - „Ausgelöste Kanäle" auf Seite 256
  - „Sicherheit von Nachrichten" auf Seite 253
  - ALW „Kanäle in AIX, Linux, and Windows überwachen und steuern" auf Seite 264
  - IBM i „Kanäle in IBM i überwachen und steuern" auf Seite 289

**Zugehörige Konzepte**

„IBM MQ for z/OS einrichten" auf Seite 972
Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

**Zugehörige Tasks**

„Verbindungen zwischen Client und Server konfigurieren" auf Seite 15
Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

„WS-Manager-Cluster konfigurieren" auf Seite 312
Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Sie können Cluster-Komponenten definieren und Cluster erstellen und verwalten.

„IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern" auf Seite 90
Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen, indem Sie die Informationen in den Konfigurationsdateien (`.ini`) bearbeiten. Sie können auch Konfigurationsoptionen für IBM MQ MQI clientsändern.

„Warteschlangenmanager unter z/OS erstellen" auf Seite 966
Verwenden Sie diese Anweisungen zum Konfigurieren von Warteschlangenmanagern unter IBM MQ for z/OS.

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten" auf Seite 1052
In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

## Verteilte Warteschlangenverfahren in IBM MQ

In den Unterabschnitten in diesem Abschnitt werden die Verfahren beschrieben, die bei der Planung von Kanälen verwendet werden. In diesen Unterabschnitten werden Verfahren beschrieben, mit deren Hilfe Sie planen, wie die Warteschlangenmanager miteinander verbunden werden, und den Fluss von Nachrichten zwischen Ihren Anwendungen verwalten.

Informationen zu den Planungsbeispielen für Nachrichtenkanäle finden Sie unter:

- ALW Beispiel für Nachrichtenkanalplanung für AIX, Linux, and Windows
- IBM i Beispiel für Nachrichtenkanalplanung für IBM i
- z/OS Beispiel für Nachrichtenkanalplanung für z/OS
- z/OS Beispiel für Nachrichtenkanalplanung für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange

**Zugehörige Konzepte**

Kanäle

Einführung in die Nachrichtenwarteschlangensteuerung

Verteilte Warteschlangen und Cluster

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

**Zugehörige Verweise**

Beispielkonfigurationsdaten

## Nachrichtenflusssteuerung

Die Nachrichtenflusssteuerung ist eine Task, die die Einrichtung und Verwaltung von Nachrichtenrouten zwischen Warteschlangenmanagern umfasst. Es ist wichtig für Routen, die Multi-Hop durch viele WS-Manager. In diesem Abschnitt wird beschrieben, wie Sie Warteschlangen, Aliaswarteschlangendefinitionen und Nachrichtenkanäle auf Ihrem System verwenden, um die Nachrichtenflusssteuerung zu erreichen.

Sie steuern den Nachrichtenfluss mit einer Reihe von Verfahren, die in „Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206 eingeführt wurden. Wenn sich Ihr Warteschlangenmanager in einem Cluster befindet, wird der Nachrichtenfluss unter Verwendung unterschiedlicher Verfahren gesteuert, wie in „Nachrichtenflusssteuerung” auf Seite 208 beschrieben. z/OS Wenn sich Ihre Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange befinden und eine gruppeninterne Warteschlangensteuerung (IGQ) aktiviert ist, kann der Nachrichtenfluss von IGQ-Agenten gesteuert werden. Diese Agenten werden in Gruppeninterne Warteschlangensteuerung beschrieben.

Sie können die folgenden Objekte verwenden, um die Nachrichtenflusssteuerung zu erreichen:

- Übertragungswarteschlangen
- Nachrichtenkanäle
- Definition der fernen Warteschlange
- WS-Manager-Aliasdefinition
- Aliasdefinition der Warteschlange für Antwortwarteschlange

Die WS-Manager- und Warteschlangenobjekte werden in Objekttypen beschrieben. Nachrichtenkanäle werden in Verteilte Warteschlangenkomponenten beschrieben. Die folgenden Verfahren verwenden diese Objekte, um Nachrichtenflüsse in Ihrem System zu erstellen:

- Nachrichten in ferne Warteschlangen stellen
- Routing über bestimmte Übertragungswarteschlangen
- Nachrichten empfangen
- Nachrichten über das System übergeben
- Getrennte Nachrichtenflüsse
- Nachrichtenflüsse zu einem anderen Ziel wechseln
- Auflösen des Namens der Empfangswarteschlange für Antworten auf einen Aliasnamen

## Hinweis

Alle in diesem Abschnitt beschriebenen Konzepte sind für alle Knoten in einem Netz relevant und schließen das Senden und Empfangen von Nachrichtenkanalenden ein. Aus diesem Grund ist in den meisten Beispielen nur ein Knoten dargestellt. Die Ausnahme ist die explizite Zusammenarbeit des Administrators am anderen Ende eines Nachrichtenkanals durch den Administrator.

Bevor Sie mit den einzelnen Verfahren fortfahren, ist es sinnvoll, die Konzepte der Namensauflösung und die drei Möglichkeiten der Verwendung von Definitionen ferner Warteschlangen zu verwenden. Siehe Verteilte Warteschlangensteuerung und Cluster.

**Zugehörige Konzepte**

„Warteschlangennamen in Übertragungsheader" auf Seite 209
Zielwarteschlangennamen werden mit der Nachricht in den Übertragungsheader übertragen, bis die Zielwarteschlange erreicht ist.

„Vorgehensweise zum Erstellen von Warteschlangenmanagern und Antwortaliasnamen" auf Seite 209
In diesem Thema werden die drei Möglichkeiten erläutert, wie Sie eine Definition einer fernen Warteschlange erstellen können.

### *Warteschlangennamen in Übertragungsheader*

Zielwarteschlangennamen werden mit der Nachricht in den Übertragungsheader übertragen, bis die Zielwarteschlange erreicht ist.

Der von der Anwendung verwendete Warteschlangenname, der Name der logischen Warteschlange, wird vom WS-Manager in den Namen der Zielwarteschlange aufgelöst. Mit anderen Worten: der Name der physischen Warteschlange. Dieser Zielwarteschlangenname wird mit der Nachricht in einem separaten Datenbereich, dem Übertragungsheader, übertragen, bis die Zielwarteschlange erreicht ist. Der Übertragungsheader wird dann abgestreift.

Sie ändern den Warteschlangenmanagerteil dieses Warteschlangennamens, wenn Sie parallele Serviceklassen erstellen. Denken Sie daran, den Namen des Warteschlangenmanagers an den ursprünglichen Namen zurückzugeben, wenn das Ende der Serviceklassendiversion erreicht ist.

### *Vorgehensweise zum Erstellen von Warteschlangenmanagern und Antwortaliasnamen*

In diesem Thema werden die drei Möglichkeiten erläutert, wie Sie eine Definition einer fernen Warteschlange erstellen können.

Das Definitionsobjekt für die ferne Warteschlange wird auf drei verschiedene Arten verwendet. In Tabelle 16 auf Seite 210 wird beschrieben, wie die einzelnen drei Arten definiert werden:

- Verwenden Sie eine Definition einer fernen Warteschlange, um einen lokalen Warteschlangennamen erneut zu definieren.

  Die Anwendung stellt beim Öffnen einer Warteschlange nur den Warteschlangennamen zur Verfügung, und dieser Warteschlangenname ist der Name der Definition der fernen Warteschlange.

  Die Definition der fernen Warteschlange enthält die Namen der Zielwarteschlange und des Warteschlangenmanagers. Optional kann die Definition den Namen der Übertragungswarteschlange enthalten, die verwendet werden soll. Wenn kein Übertragungswarteschlangenname angegeben wird, verwendet der Warteschlangenmanager den Namen des Warteschlangenmanagers, der aus der Definition der fernen Warteschlange entnommen wurde, für den Namen der Übertragungswarteschlange. Wenn eine Übertragungswarteschlange mit diesem Namen nicht definiert ist, aber eine Standardübertragungswarteschlange definiert ist, wird die Standardübertragungswarteschlange verwendet.
- Verwenden Sie eine Definition einer fernen Warteschlange, um einen WS-Manager-Namen erneut zu definieren.

  Das Anwendungs-oder Kanalprogramm stellt beim Öffnen der Warteschlange einen Warteschlangennamen zusammen mit dem Namen des fernen Warteschlangenmanagers bereit.

Wenn Sie eine ferne Warteschlangendefinition mit demselben Namen wie der Name des Warteschlangenmanagers angegeben haben und den Warteschlangennamen in der Definition leer gelassen haben, ersetzt der Warteschlangenmanager den Namen des Warteschlangenmanagers im offenen Aufruf mit dem Namen des Warteschlangenmanagers in der Definition.

Darüber hinaus kann die Definition den Namen der Übertragungswarteschlange enthalten, die verwendet werden soll. Wenn kein Übertragungswarteschlangenname angegeben wird, verwendet der Warteschlangenmanager den Namen des Warteschlangenmanagers, der aus der Definition der fernen Warteschlange für den Namen der Übertragungswarteschlange entnommen wurde. Wenn eine Übertragungswarteschlange mit diesem Namen nicht definiert ist, aber eine Standardübertragungswarteschlange definiert ist, wird die Standardübertragungswarteschlange verwendet.

- Verwenden Sie eine Definition einer fernen Warteschlange, um einen Namen der Empfangswarteschlange für Antworten erneut zu definieren.

  Jedes Mal, wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, kann sie den Namen einer Warteschlange für Antwortnachrichten für Antwortnachrichten angeben, jedoch mit dem Namen des Warteschlangenmanagers.

  Wenn Sie eine ferne Warteschlangendefinition mit demselben Namen wie die Empfangswarteschlange für Antworten angeben, ersetzt der lokale WS-Manager den Namen der Warteschlange für die Antwort auf die Warteschlange durch den Namen der Warteschlange aus Ihrer Definition.

  Sie können einen Warteschlangenmanagernamen in der Definition angeben, jedoch keinen Namen für die Übertragungswarteschlange.

*Tabelle 16. Drei Methoden zur Verwendung des Definitionsobjekts für ferne Warteschlangen*

| **Verwendung** | **Name des Warteschlangenmanagers** | **Warteschlangenname** | **Name der Übertragungswarteschlange** |
|---|---|---|---|
| 1. Definition der fernen Warteschlange (bei OPEN-Aufruf) | | | |
| Bereitgestellte im Aufruf | Leer oder lokaler QM | (*) erforderlich | Nicht zutreffend |
| Bereitgestellte in der Definition | erforderlich | erforderlich | optional |
| 2. WS-Manager-Aliasname (bei OPEN-Aufruf) | | | |
| Bereitgestellte im Aufruf | (*) erforderlich und nicht lokaler QM | erforderlich | Nicht zutreffend |
| Bereitgestellte in der Definition | erforderlich | Leer | optional |
| 3. Aliasname für Antwortwarteschlange (bei PUT-Aufruf) | | | |
| Bereitgestellte im Aufruf | Leer | (*) erforderlich | Nicht zutreffend |
| Bereitgestellte in der Definition | optional | optional | Leer |

**Anmerkung:** (*) bedeutet, dass dieser Name der Name des Definitionsobjekts ist.

Eine formale Beschreibung finden Sie in Warteschlangennamensauflösung .

## Einreihen von Nachrichten in ferne Warteschlangen

Sie können ferne Warteschlangendefinitionsobjekte verwenden, um einen Warteschlangennamen in eine Übertragungswarteschlange in einen benachbarten WS-Manager aufzulösen.

In einer Umgebung mit verteilter Warteschlange sind eine Übertragungswarteschlange und ein Kanal der Sammelpunkt für alle Nachrichten an eine Position, ob die Nachrichten von Anwendungen in Ihrem lokalen System stammen oder die über Kanäle von einem benachbarten System ankommen. Abbildung 6 auf Seite 211 zeigt eine Anwendung, in der Nachrichten in eine logische Warteschlange mit dem Namen 'QA_norm' gestellt werden. In der Namensauflösung wird die ferne Warteschlangendefinition 'QA_norm'

verwendet, um die Übertragungswarteschlange QMB auszuwählen. Anschließend wird ein Übertragungsheader zu den Nachrichten mit der Nachricht 'QA_norm at QMB' hinzugefügt.

Nachrichten, die vom benachbarten System in 'Channel_back' ankommen, weisen einen Übertragungsheader mit dem Namen der physischen Warteschlange 'QA_norm auf QMB' auf. Beispiel: Diese Nachrichten werden unverändert in die Übertragungswarteschlange QMB gestellt.

Der Kanal verschiebt die Nachrichten in einen benachbarten WS-Manager.

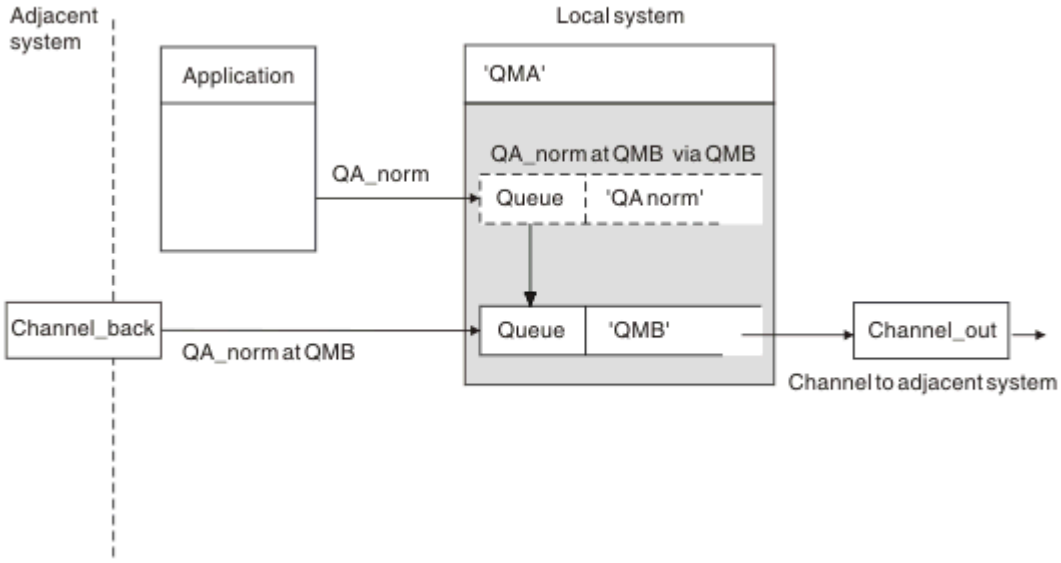



*Abbildung 6. Eine Definition einer fernen Warteschlange wird verwendet, um einen Warteschlangennamen in eine Übertragungswarteschlange in einen benachbarten WS-Manager aufzulösen.*

Wenn Sie der IBM MQ-Systemadministrator sind, müssen Sie folgende Schritte ausführen:

- Den Nachrichtenkanal aus dem benachbarten System definieren
- Definieren Sie den Nachrichtenkanal für das benachbarte System.
- Erstellen Sie die Übertragungswarteschlange QMB.
- Definieren Sie das ferne Warteschlangenobjekt 'QA_norm', um den Namen der Warteschlange aufzulösen, die von Anwendungen für den Namen der Zielwarteschlange, den Namen des Zielwarteschlangenmanagers und den Namen der Übertragungswarteschlange verwendet wird.

In einer Clustering-Umgebung müssen Sie nur einen Clusterempfängerkanal auf dem lokalen Warteschlangenmanager definieren. Es ist nicht erforderlich, eine Übertragungswarteschlange oder ein fernes Warteschlangenobjekt zu definieren. Siehe Cluster.

## Weitere Informationen zur Namensauflösung

Der Effekt der Definition einer fernen Warteschlange besteht darin, einen Namen für die physische Zielwarteschlange und den Namen des Warteschlangenmanagers zu definieren. Diese Namen werden in die Übertragungsheader von Nachrichten gestellt.

Eingehende Nachrichten von einem benachbarten System haben bereits diesen Typ von Namensauflösung, die vom ursprünglichen Warteschlangenmanager ausgeführt wurde. Daher weisen sie den Übertragungsheader mit dem Namen der physischen Zielwarteschlange und dem Namen des Warteschlangenmanagers auf. Diese Nachrichten werden von den Definitionen der fernen Warteschlange nicht beeinflusst.

**Zugehörige Verweise**
Auflösung des Warteschlangennamens

## Die Übertragungswarteschlange auswählen

Sie können eine ferne Warteschlangendefinition verwenden, um eine andere Übertragungswarteschlange zuzulassen, um Nachrichten an denselben benachbarten Warteschlangenmanager zu senden.

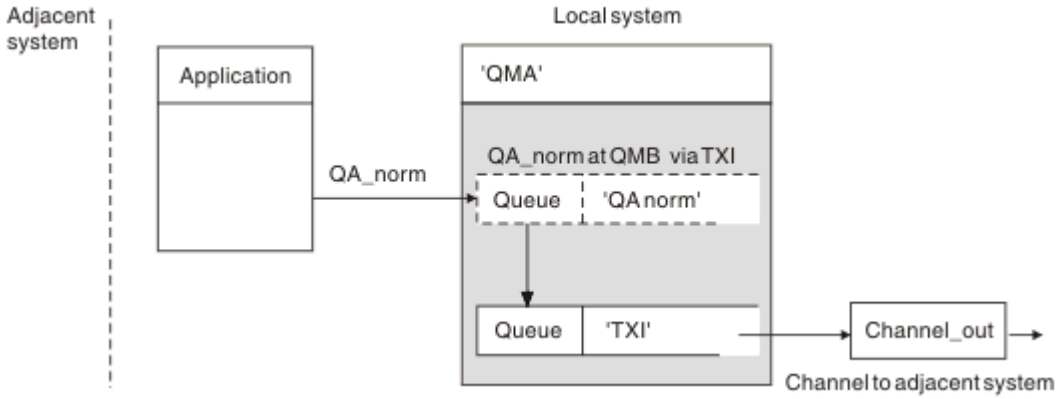



*Abbildung 7. Die Definition der fernen Warteschlange ermöglicht, dass eine andere Übertragungswarteschlange verwendet wird.*

Wenn Sie in einer Umgebung mit verteilter Warteschlange einen Nachrichtenfluss von einem Kanal in einen anderen ändern müssen, verwenden Sie dieselbe Systemkonfiguration wie in Abbildung 6 auf Seite 211 in „Einreihen von Nachrichten in ferne Warteschlangen” auf Seite 210 gezeigt. Abbildung 7 auf Seite 212 in diesem Thema zeigt, wie Sie die Definition der fernen Warteschlange verwenden, um Nachrichten über eine andere Übertragungswarteschlange und somit über einen anderen Kanal an denselben benachbarten Warteschlangenmanager zu senden.

Für die in Abbildung 7 auf Seite 212 gezeigte Konfiguration müssen Sie das ferne Warteschlangenobjekt 'QA_norm' und die Übertragungswarteschlange 'TX1' angeben. Sie müssen 'QA_norm' angeben, um die Warteschlange 'QA_norm' auf dem fernen WS-Manager, die Übertragungswarteschlange 'TX1' und den Warteschlangenmanager 'QMB_priority' auszuwählen. Geben Sie 'TX1' in der Definition des Kanals an, der an das System angrenzender Kanal ist.

Nachrichten werden in die Übertragungswarteschlange 'TX1' mit einem Übertragungsheader gestellt, der 'QA_norm bei QMB_priority' enthält, und werden über den Kanal an das benachbarte System gesendet.

Der Kanal 'channel_back' wurde aus dieser Abbildung weggelassen, da er einen WS-Manager-Aliasnamen benötigen würde.

In einer Clustering-Umgebung müssen Sie keine Übertragungswarteschlange oder eine Definition einer fernen Warteschlange definieren. Weitere Informationen finden Sie unter „Clusterwarteschlangen definieren” auf Seite 313.

## Nachrichten empfangen

Sie können den Warteschlangenmanager für den Empfang von Nachrichten von anderen Warteschlangenmanagern konfigurieren. Sie müssen sicherstellen, dass eine unbeabsichtigte Namensauflösung nicht auftritt.

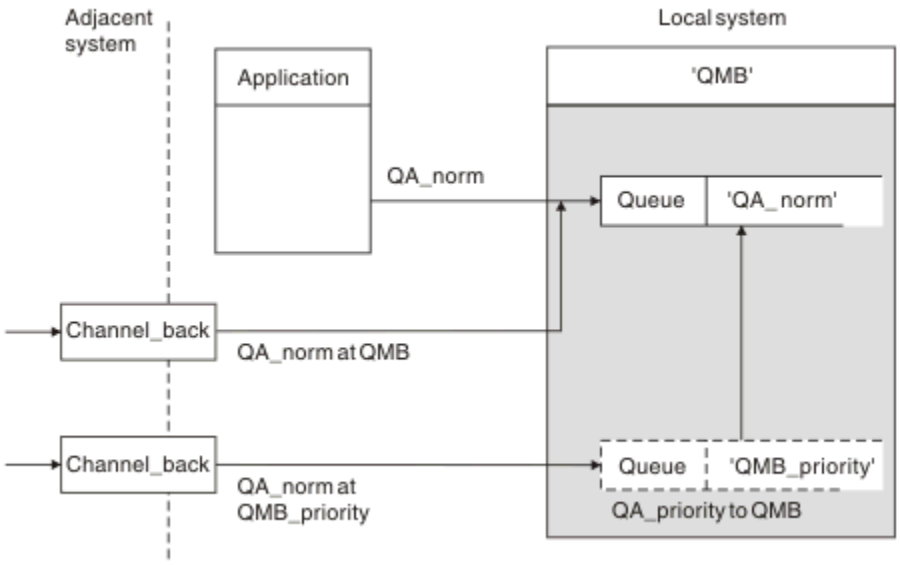



*Abbildung 8. Nachrichten direkt empfangen und Aliasnamen-WS-Manager-Namen auflösen*

Neben der Anordnung von Nachrichten, die gesendet werden sollen, muss der Systemadministrator auch die Nachrichten veranlassen, die von benachbarten Warteschlangenmanagern empfangen werden. Empfangene Nachrichten enthalten den physischen Namen des Zielwarteschlangenmanagers und die Warteschlange in der Übertragungsheader. Sie werden mit Nachrichten aus einer lokalen Anwendung behandelt, die sowohl den Namen des Warteschlangenmanagers als auch den Warteschlangennamen angibt. Aufgrund dieser Behandlung müssen Sie sicherstellen, dass Nachrichten, die in Ihr System eingegeben werden, nicht über eine unbeabsichtigte Namensauflösung verfügen. Dieses Szenario enthält Abbildung 8 auf Seite 213 .

Für diese Konfiguration müssen Sie Folgendes vorbereiten:

- Nachrichtenkanäle zum Empfangen von Nachrichten von benachbarten Warteschlangenmanagern
- Eine WS-Manager-Aliasdefinition zum Auflösen eines eingehenden Nachrichtenflusses, 'QMB_priority', in den lokalen WS-Manager-Namen 'QMB'
- Die lokale Warteschlange 'QA_norm', falls sie nicht vorhanden ist.

### Namen von Aliaswarteschlangenmanagern empfangen

Die Verwendung der Definition des WS-Manager-Aliasnamens in dieser Abbildung hat keinen anderen Zielwarteschlangenmanager ausgewählt. Nachrichten, die diesen lokalen WS-Manager durchlaufen und an 'QMB_priority' adressiert sind, sind für den Warteschlangenmanager 'WSMB' bestimmt. Der Name des Aliaswarteschlangenmanagers wird zum Erstellen des separaten Nachrichtenflusses verwendet.

## Nachrichten über das System übergeben

Sie können Nachrichten über das System auf drei Arten übergeben-unter Verwendung des Positionsnamens, unter Verwendung eines Aliasnamens für den Warteschlangenmanager oder durch Auswahl einer Übertragungswarteschlange.

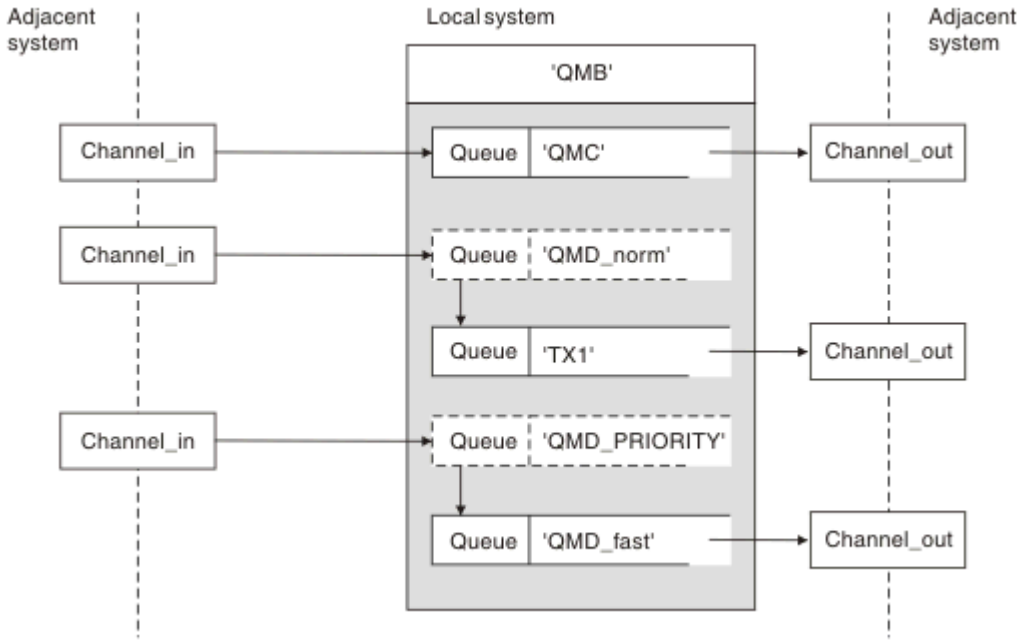



*Abbildung 9. Drei Methoden zum Übergeben von Nachrichten durch Ihr System*

Das in Abbildung 8 auf Seite 213 in „Nachrichten empfangen” auf Seite 212 gezeigte Verfahren hat gezeigt, wie ein Aliasfluss erfasst wird. Abbildung 9 auf Seite 214 veranschaulicht, wie Netze aufgebaut werden, indem die zuvor beschriebenen Verfahren zusammengeführt werden.

Die Konfiguration zeigt einen Kanal, der drei Nachrichten mit unterschiedlichen Zieladressen liefert:

1. QB an QMC
2. QB an QMD_norm
3. QB an QMD_PRIORITY

Sie müssen den ersten Nachrichtenfluss unverändert über Ihr System übergeben. Sie müssen den zweiten Nachrichtenfluss über eine andere Übertragungswarteschlange und einen anderen Kanal übergeben. Für den zweiten Nachrichtenfluss müssen Sie auch Nachrichten für den Aliasnamen WS-Managername QMD_norm in den Warteschlangenmanager QMD auflösen. Der dritte Nachrichtenfluss wählt eine andere Übertragungswarteschlange ohne jede andere Änderung aus.

In einer Clustering-Umgebung werden Nachrichten über eine Clusterübertragungswarteschlange übergeben. Normalerweise überträgt eine einzelne Übertragungswarteschlange ( SYSTEM.CLUSTER.TRANSMIT.QUEUE) alle Nachrichten an alle Warteschlangenmanager in allen Clustern, deren Mitglied der Warteschlangenmanager ist (siehe Cluster von Warteschlangenmanagern). Sie können separate Übertragungswarteschlangen für alle oder einige der WS-Manager in den Clustern definieren, zu denen der Warteschlangenmanager gehört.

Die folgenden Methoden beschreiben Verfahren, die auf eine Umgebung mit verteilten Warteschlangen anwendbar sind.

## Verwenden Sie diese Methoden.

Für diese Konfigurationen müssen Sie Folgendes vorbereiten:

- Eingabekanaldefinitionen
- Ausgabekanaldefinitionen
- Übertragungswarteschlangen:
  - QMC
  - TX1

  - QMD_fast
- Definitionen des WS-Manager-Aliasnamens
  - QMD_norm mit QMD_norm an QMD über TX1
  - QMD_PRIORITY mit QMD_PRIORITY an QMD_PRIORITY über QMD_fast

**Anmerkung:** Keiner der Nachrichtenflüsse, die im Beispiel angezeigt werden, ändert die Zielwarteschlange. Die Aliasnamen des Warteschlangenmanagers stellen die Trennung von Nachrichtenflüssen zur Verfügung.

### Methode 1: Ankommenden Standortnamen verwenden

Sie werden Nachrichten mit einem Übertragungsheader empfangen, der einen anderen Standortnamen enthält, z. B. QMC. Die einfachste Konfiguration besteht darin, eine Übertragungswarteschlange mit diesem Namen zu erstellen, QMC. Der Kanal, der die Übertragungswarteschlange bereitstellt, übergibt die Nachricht unverändert an das nächste Ziel.

### Methode 2: Aliasnamen für den WS-Manager verwenden

Die zweite Methode ist die Verwendung der WS-Manager-Aliasobjektdefinition, aber geben Sie einen neuen Standortnamen, QMD und eine bestimmte Übertragungswarteschlange an, TX1. Diese Aktion:

- Beendet den Aliasnachrichtenfluss, der durch den Aliasnamen QMD_norm des Warteschlangenmanagers, d. a. die benannte Serviceklasse QMD_norm, konfiguriert wird.
- Ändert die Übertragungsheader in diesen Nachrichten von QMD_norm in QMD.

### Methode 3: Wählen Sie eine Übertragungswarteschlange aus.

Die dritte Methode besteht darin, dass ein WS-Manager-Aliasobjekt mit demselben Namen wie die Zielposition QMD_PRIORITY definiert ist. Verwenden Sie die Definition des WS-Manager-Aliasnamens, um eine bestimmte Übertragungswarteschlange, QMD_fast und somit einen anderen Kanal auszuwählen. Die Übertragungsheader in diesen Nachrichten bleiben unverändert.

## Getrennte Nachrichtenflüsse

Sie können einen WS-Manager-Aliasnamen verwenden, um separate Nachrichtenflüsse zu erstellen, um Nachrichten an denselben WS-Manager zu senden.

### Gründe für die Trennung von Nachrichten in unterschiedliche Nachrichtenflüsse

In einer Umgebung mit verteilten Warteschlangen kann die Notwendigkeit, Nachrichten an denselben WS-Manager in verschiedene Nachrichtenflüsse zu trennen, aus einer Reihe von Gründen auftreten. For example:

- Möglicherweise müssen Sie einen separaten Nachrichtenfluss für große, mittlere und kleine Nachrichten bereitstellen. Diese Notwendigkeit gilt auch in einer Clustering-Umgebung, und in diesem Fall können Sie Cluster erstellen, die sich überschneiden. Es gibt eine Reihe von Gründen, die Sie vielleicht tun könnten, z. B.:
  - Damit andere Organisationen ihre eigene Verwaltung haben können.
  - Damit unabhängige Anwendungen separat verwaltet werden können.
  - So erstellen Sie eine Serviceklasse. Sie könnten z. B. einen Cluster mit dem Namen STAFF haben, der eine Untergruppe des Clusters mit dem Namen STUDENTS ist. Wenn Sie eine Nachricht in eine Warteschlange stellen, die im STAFF-Cluster zugänglich gemacht wird, wird ein eingeschränkter Kanal verwendet. Wenn Sie eine Nachricht in eine Warteschlange stellen, die im STUDENTS-Cluster zugänglich gemacht wird, kann entweder ein allgemeiner Kanal oder ein eingeschränkter Kanal verwendet werden.

  - So erstellen Sie Test-und Produktionsumgebungen.
- Es kann erforderlich sein, eingehende Nachrichten durch unterschiedliche Pfade aus dem Pfad der lokal generierten Nachrichten weiterzuleiten.
- Ihre Installation muss unter Umständen die Verschiebung von Nachrichten zu bestimmten Zeiten (z. B. über Nacht) planen, und die Nachrichten müssen dann in reservierten Warteschlangen gespeichert werden, bis sie geplant sind.

## Beispielnachrichtenfluss

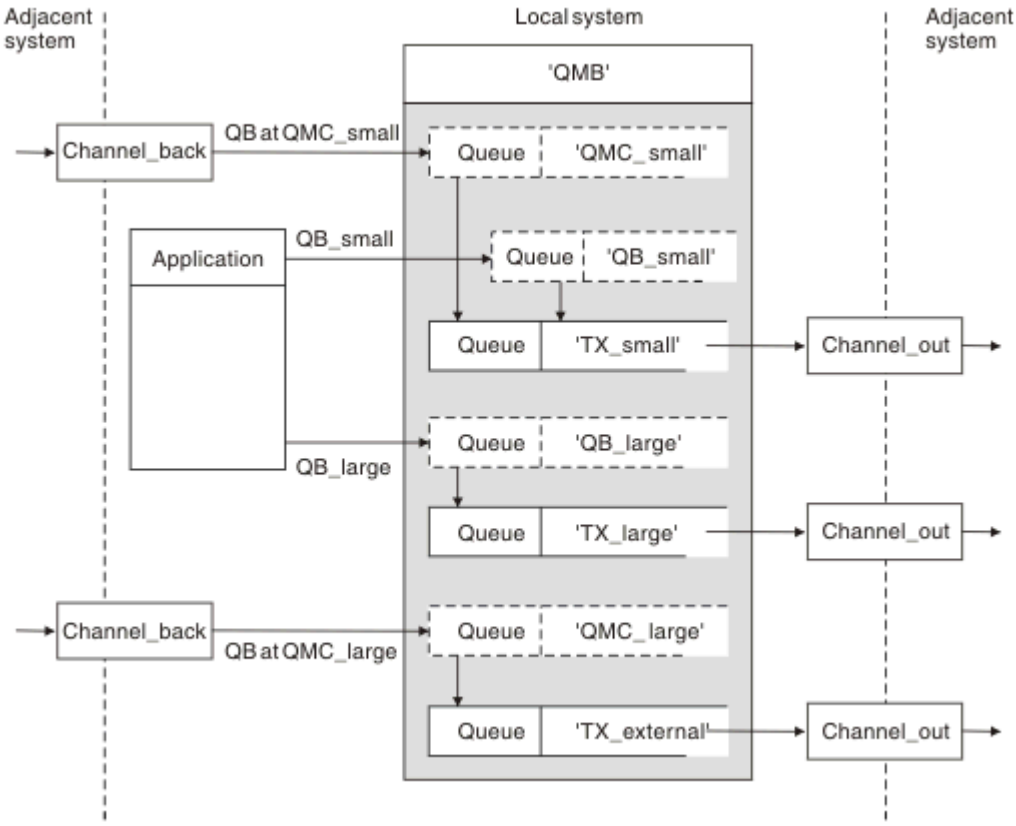



*Abbildung 10. Nachrichtenflüsse trennen*

In dem in Abbildung 10 auf Seite 216gezeigten Beispiel werden die beiden eingehenden Flüsse als Aliaswarteschlangenmanagernamen 'QMC_small' und 'QMC_large ' verwendet. Sie stellen diese Flüsse mit einer WS-Manager-Aliasdefinition bereit, um diese Nachrichtenflüsse für den lokalen WS-Manager zu erfassen. Sie verfügen über eine Anwendung, die zwei ferne Warteschlangen adressieren soll, und Sie benötigen diese Nachrichtenflüsse getrennt voneinander. Sie stellen zwei Definitionen für ferne Warteschlangen bereit, die dieselbe Position angeben, 'QMC', aber geben Sie andere Übertragungswarteschlangen an. Diese Definition behält die Abläufe bei und ist am Ende nicht mehr erforderlich, da sie denselben Namen für den Zielwarteschlangenmanager in den Übertragungsheadern haben. Sie stellen Folgendes bereit:

- Die eingehenden Kanaldefinitionen
- Die beiden fernen Warteschlangendefinitionen QB_small und QB_large
- Die beiden WS-Manager-Aliasnamendefinitionen QMC_small und QMC_large
- Die drei sendenden Kanaldefinitionen
- Drei Übertragungswarteschlangen: TX_small, TX_large und TX_external

## Koordination mit benachbarten Systemen

Wenn Sie einen WS-Manager-Aliasnamen verwenden, um einen separaten Nachrichtenfluss zu erstellen, müssen Sie diese Aktivität mit dem Systemadministrator am fernen Ende des Nachrichtenkanals koordinieren, um sicherzustellen, dass der entsprechende WS-Manager-Aliasname dort verfügbar ist.

# Konzentration von Botschaften an verschiedene Standorte

Sie können Nachrichten, die für verschiedene Orte bestimmt sind, auf einen einzigen Kanal konzentrieren.

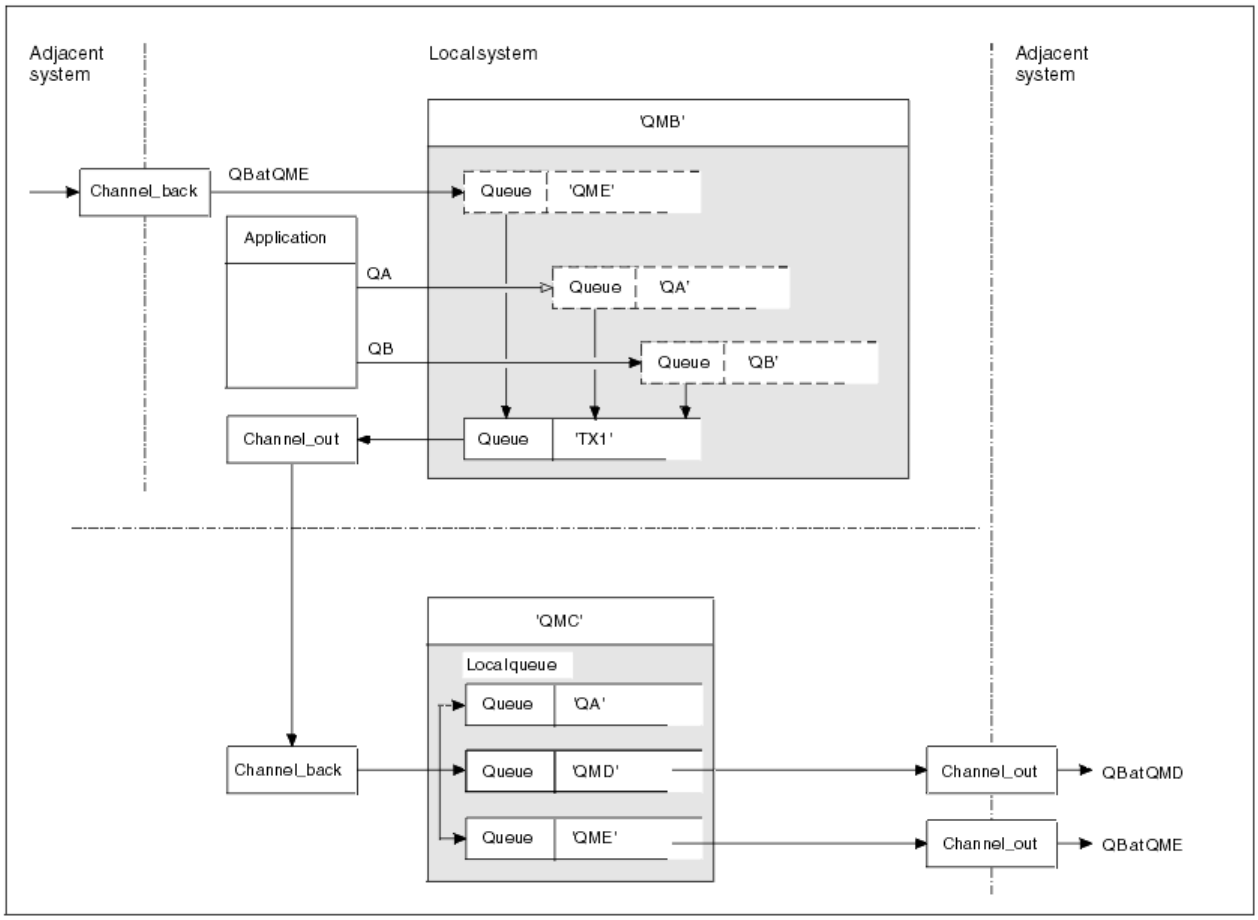



*Abbildung 11. Kombinieren von Nachrichtenflüssen auf einem Kanal*

Abbildung 11 auf Seite 217 veranschaulicht ein Verfahren zur verteilten Steuerung von Nachrichten, das für die Konzentration von Nachrichten bestimmt ist, die für verschiedene Positionen in einem Kanal bestimmt sind. Zwei Verwendungsmöglichkeiten wären denkbar:

- Nachrichtenverkehr durch ein Gateway konzentrieren
- Verwenden von Hochgeschwindigkeits-Highways zwischen Knoten

In diesem Beispiel werden Nachrichten aus verschiedenen Quellen, lokalen und benachbarten und mit unterschiedlichen Zielwarteschlangen und Warteschlangenmanagern über die Übertragungswarteschlange 'TX1' in den Warteschlangenmanager QMC übertragen. Der WS-Manager QMC stellt die Nachrichten entsprechend den Zielen bereit. Eine Übertragungswarteschlange 'QMD' wurde für die Weiterübertragung an Warteschlangenmanager QMD festgelegt. Eine andere Gruppe in eine Übertragungswarteschlange 'QME' für die Weiterleitung an Warteschlangenmanager QME gesetzt. Andere Nachrichten werden in die lokale Warteschlange 'QA' gestellt.

Sie müssen Folgendes angeben:

- Kanal- definitionen
- Übertragungswarteschlange TX1

- Definitionen ferner Warteschlangen:
  - QA mit 'QA bei QMC über TX1'
  - QB mit 'QB auf QMD bis TX1'
- Definition des WS-Manager-Aliasnamens
  - QME mit 'QME bis TX1'

Der ergänzende Administrator, der die Konfiguration von QMC konfiguriert, muss Folgendes bereitstellen:

- Kanaldefinition mit demselben Kanalnamen empfangen
- Übertragungswarteschlange QMD mit zugeordneter Sendekanaldefinition
- Übertragungswarteschlange QME mit zugeordneter Sendekanaldefinition
- Lokales Warteschlangenobjekt QA.

## Nachrichtenflüsse zu einem anderen Ziel umwählen

Sie können die Zieladresse bestimmter Nachrichten mit Hilfe von WS-Manager-Aliasnamen und -Übertragungswarteschlangen neu definieren.

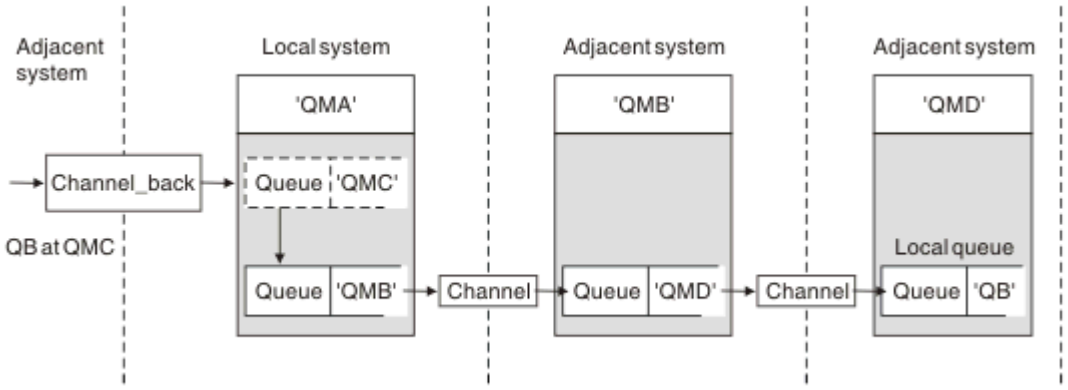



*Abbildung 12. Nachrichtenströme zu einem anderen Ziel verwählen*

In Abbildung 12 auf Seite 218 ist dargestellt, wie Sie das Ziel bestimmter Nachrichten neu definieren können. Eingehende Nachrichten an QMA sind für 'QB at QMC' bestimmt. Sie kommen normalerweise bei QMA an und werden in eine Übertragungswarteschlange mit dem Namen QMC gestellt, die Teil eines Kanals zu QMC war. Die QMA muss die Nachrichten in QMD umleiten, aber die WSMD nur über WSMB erreichen. Diese Methode ist nützlich, wenn Sie einen Service von einer Position in eine andere versetzen müssen, und es den Subskribenten ermöglichen, Nachrichten temporär zu senden, bis sie an die neue Adresse angepasst wurden.

Die Methode, eingehende Nachrichten, die für einen bestimmten Warteschlangenmanager bestimmt sind, an einen anderen Warteschlangenmanager zu umleiten:

- Ein WS-Manager-Aliasname zum Ändern des Zielwarteschlangenmanagers in einen anderen Warteschlangenmanager und zum Auswählen einer Übertragungswarteschlange für das benachbarte System.
- Eine Übertragungswarteschlange, die dem benachbarten WS-Manager dient
- Eine Übertragungswarteschlange auf dem benachbarten Warteschlangenmanager für die Weiterleitung an den Zielwarteschlangenmanager.

Sie müssen Folgendes angeben:

- Channel_back-Definition
- WS-Manager-Aliasobjektdefinition QMC mit QB auf QMD über WSMB
- Channel_out-Definition
- Die zugeordnete Übertragungswarteschlange (WSMB)

Der ergänzende Administrator, der die Konfiguration von WSMB konfiguriert, muss Folgendes bereitstellen:

- Die entsprechende Channel_back-Definition
- Die Übertragungswarteschlange, QMD
- Die zugeordnete Kanaldefinition zu QMD

Sie können Aliasnamen in einer Clustering-Umgebung verwenden. Weitere Informationen finden Sie in „WS-Manager-Aliasnamen und -Cluster” auf Seite 415.

## Nachrichten an eine Verteilerliste senden

Sie können einen einzigen MQPUT-Aufruf verwenden, damit eine Anwendung eine Nachricht an mehrere Ziele sendet.

In IBM MQ kann eine Anwendung auf allen Plattformen mit Ausnahme von z/OS eine Nachricht mit einem einzigen MQPUT-Aufruf an mehrere Ziele senden. Sie können dies sowohl in einer Umgebung mit verteilter Warteschlange als auch in einer Clusterumgebung tun. Sie müssen die Ziele in einer Verteilerliste definieren, wie in Verteilerlisten beschrieben.

Nicht alle Warteschlangenmanagern unterstützen Verteilerlisten. Wenn ein MCA eine Verbindung mit einem Partner herstellt, bestimmt er, ob der Partner Verteilerlisten unterstützt und dementsprechend eine Markierung in der Übertragungswarteschlange setzt. Wenn eine Anwendung versucht, eine Nachricht zu senden, die für eine Verteilerliste bestimmt ist, der Partner jedoch Verteilerlisten nicht unterstützt, fängt der sendende MCA die Nachricht ab und stellt sie einmal für jedes geplante Ziel in die Übertragungswarteschlange.

Durch einen empfangenden MCA wird sichergestellt, dass Nachrichten, die an eine Verteilerliste gesendet werden, sicher an allen vorgesehenen Zieladressen empfangen werden. Wenn ein Ziel fehlschlägt, stellt der MCA fest, welche Fehler aufgetreten sind. Anschließend kann sie Ausnahmebedingungsberichte für sie generieren und versuchen, die Nachrichten erneut an sie zu senden.

## Empfangswarteschlange für Antworten

Sie können eine vollständige ferne Warteschlangenverarbeitungsschleife mit Hilfe einer Warteschlange für Antwortantworten erstellen.

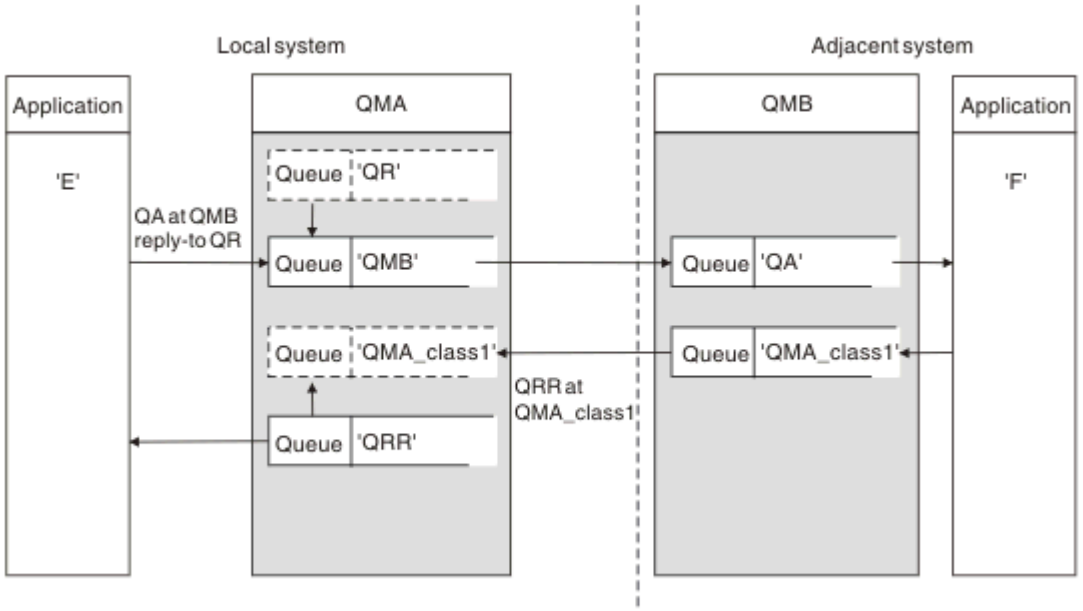



*Abbildung 13. Name der Warteschlange für Antwortwarteschlangen beim PUT-Aufruf*

In Abbildung 13 auf Seite 219 wird eine vollständige Verarbeitungsschleife der fernen Warteschlange mit Hilfe einer Warteschlange für Antwortantworten angezeigt. Diese Schleife gilt sowohl in einer Umgebung mit verteilter Warteschlange als auch in einer Clustering-Umgebung. Die Details sind wie in Tabelle 20 auf Seite 228 dargestellt.

Die Anwendung öffnet QA auf WSMB und reiht Nachrichten in diese Warteschlange ein. Die Nachrichten erhalten einen Antwortwarteschlangennamen von QR, ohne dass der Name des Warteschlangenmanagers angegeben wurde. WS-Manager QMA sucht das Antwortwarteschlangenobjekt QR und extrahiert daraus den Aliasnamen QRR und den WS-Manager-Namen QMA_class1. Diese Namen werden in die Antwortfelder der Nachrichten gestellt.

Antwortnachrichten von Anwendungen auf WSMB werden an QRR in QMA_class1 adressiert. Die WS-Manager-Aliasnamendefinition QMA_class1 wird vom WS-Manager verwendet, um die Nachrichten an sich selbst und die Warteschlange QRR zu fließen.

In diesem Szenario wird die Art und Weise dargestellt, in der Sie Anwendungen die Möglichkeit geben, eine Serviceklasse für Antwortnachrichten auszuwählen. Die Klasse wird von der Übertragungswarteschlange QMA_class1 auf WSMB zusammen mit der WS-Manager-Aliasdefinition QMA_klasse1 bei QMA implementiert. Auf diese Weise können Sie die Antwortwarteschlange einer Anwendung so ändern, dass die Nachrichtenflüsse getrennt werden, ohne dass die Anwendung beteiligt ist. Die Anwendung wählt immer QR für diese bestimmte Serviceklasse aus. Sie haben die Möglichkeit, die Serviceklasse mit der Antwort-Warteschlange-QR-Definition zu ändern.

Sie müssen Folgendes erstellen:

- Antwortwarteschlangendefinition QR
- Übertragungswarteschlangenobjekt (WSMB
- Channel_out-Definition
- Channel_back-Definition
- WS-Manager-Aliasnamendefinition QMA_class1
- Lokales Warteschlangenobjekt QRR, wenn es nicht vorhanden ist

Der ergänzende Administrator auf dem benachbarten System muss Folgendes erstellen:

- Kanaldefinition wird empfangen
- Übertragungswarteschlangenobjekt QMA_class1
- Zugeordneter Sende-Channel
- Lokales Warteschlangenobjekt QA.

Ihre Anwendungsprogramme verwenden:

- Name der Reply-to-Warteschlange QR in Anrufen
- Warteschlangenname QRR in get-Aufrufen

Auf diese Weise können Sie die Serviceklasse nach Bedarf ändern, ohne dass die Anwendung einbezogen wird. Sie ändern den Antwortalias 'QR' zusammen mit der Übertragungswarteschlange 'QMA_class1' und dem WS-Manager-Aliasnamen 'QMA_klasse1'.

Wenn beim Einlegen der Nachricht in die Warteschlange kein Objekt "reply-to alias" gefunden wird, wird der Name des lokalen Warteschlangenmanagers in das Feld für den Namen des leeren Antwortwarteschlangenmanagers (Name des Warteschlangenmanagers) eingefügt. Der Name der Empfangswarteschlange für Antworten bleibt unverändert.

## Einschränkung für Namensauflösung

Da die Namensauflösung für die Warteschlange für Antwortnachrichten bei 'QMA' ausgeführt wurde, als die ursprüngliche Nachricht gestellt wurde, ist keine weitere Namensauflösung bei 'WSMB' zulässig. Die Nachricht wird mit dem physischen Namen der Empfangswarteschlange für Antworten von der Anwendung "Antwort" in die Warteschlange gestellt.

Die Anwendungen müssen sich bewusst sein, dass der Name, den sie für die Empfangswarteschlange für Antworten verwenden, sich von dem Namen der tatsächlichen Warteschlange unterscheidet, in der die Rückgabenachrichten zu finden sind.

Wenn beispielsweise zwei Serviceklassen für die Verwendung von Anwendungen mit den Aliasnamen 'C1_alias' für Antwortwarteschlangen und 'C2_alias' zur Verfügung gestellt werden, verwenden die Anwendungen diese Namen als Antwort-in-Warteschlange-Namen in den Nachrichteneinträgen. Tatsächlich erwarten die Anwendungen jedoch, dass Nachrichten in den Warteschlangen 'C1' für 'C1_alias' und 'C2' für 'C2_alias' angezeigt werden.

Eine Anwendung kann jedoch in der Warteschlange für Antwortnachrichten einen Anfragenaufruf vornehmen, um den Namen der realen Warteschlange, die zum Abrufen der Antwortnachrichten verwendet werden muss, selbst zu überprüfen.

**Zugehörige Konzepte**

„Vorgehensweise zum Erstellen von Warteschlangenmanagern und Antwortaliasnamen” auf Seite 209
In diesem Thema werden die drei Möglichkeiten erläutert, wie Sie eine Definition einer fernen Warteschlange erstellen können.

„Beispiel für Antwortwarteschlangenalias” auf Seite 221
Dieses Beispiel veranschaulicht die Verwendung eines Antwortalias für Antworten, um eine andere Route (Übertragungswarteschlange) für zurückgegebene Nachrichten auszuwählen. Für die Verwendung dieser Funktion ist es erforderlich, dass der Name der Empfangswarteschlange für Antworten in Zusammenarbeit mit den Anwendungen geändert wird.

„Funktionsweise des Beispiels” auf Seite 223
Eine Erläuterung des Beispiels und die Verwendung des Aliasnamens der Empfangswarteschlange für Antworten durch den Warteschlangenmanager.

„Durchlauf der Warteschlange für Antwortwarteschlangen” auf Seite 224
Ein Walkthrough des Prozesses von einer Anwendung, die eine Nachricht in eine ferne Warteschlange einreihen soll, bis zu derselben Anwendung, die die Antwortnachricht aus der Warteschlange für Aliasantwortnachrichten entfernt.

## *Beispiel für Antwortwarteschlangenalias*

Dieses Beispiel veranschaulicht die Verwendung eines Antwortalias für Antworten, um eine andere Route (Übertragungswarteschlange) für zurückgegebene Nachrichten auszuwählen. Für die Verwendung dieser Funktion ist es erforderlich, dass der Name der Empfangswarteschlange für Antworten in Zusammenarbeit mit den Anwendungen geändert wird.

Wie in Abbildung 14 auf Seite 222 gezeigt, muss die Rückgaberoute für die Antwortnachrichten, einschließlich der Übertragungswarteschlange, des Kanals und des Aliasnamens des Warteschlangenmanagers, verfügbar sein.

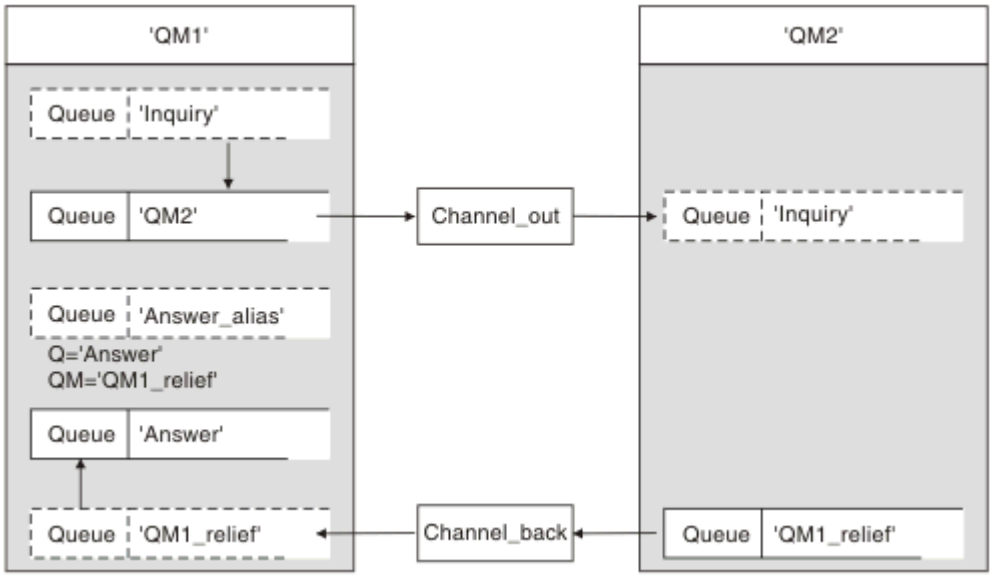



*Abbildung 14. Beispiel für Antwortwarteschlangenalias*

Dieses Beispiel gilt für Requesteranwendungen in 'QM1', die Nachrichten an Serveranwendungen in 'QM2' senden. Die Nachrichten auf dem Server sollen über einen alternativen Kanal unter Verwendung der Übertragungswarteschlange 'QM1_relief' zurückgegeben werden. (Der Standardrückgabekanal wird mit einer Übertragungswarteschlange 'QM1' bereitgestellt).

Der Aliasname der Empfangswarteschlange für Antworten ist eine bestimmte Verwendung der Definition der fernen Warteschlange mit dem Namen 'Answer_alias'. Anwendungen auf WSM1 enthalten diesen Namen, 'Answer_alias', im Feld 'reply-to' aller Nachrichten, die sie in die Warteschlange 'Abfrage' stellen.

Die Antwortwarteschlangendefinition 'Answer_alias' wird als 'Answer at QM1_relief' definiert. Anwendungen auf WSM1 erwarten, dass ihre Antworten in der lokalen Warteschlange mit dem Namen 'Answer' angezeigt werden.

Serveranwendungen bei QM2 verwenden das Antwortfeld der empfangenen Nachrichten, um die Namen der Warteschlange und des Warteschlangenmanagers für die Antwortnachrichten an den Requester auf QM1 zu erhalten.

## Definitionen, die in diesem Beispiel auf QM1 verwendet werden

Der IBM MQ-Systemadministrator bei QM1 muss sicherstellen, dass die Empfangswarteschlange für Antworten 'Antwort' zusammen mit den anderen Objekten erstellt wird. Der Name des WS-Manager-Aliasnamens, der mit einem Stern (*) gekennzeichnet ist, muss mit dem Namen des Warteschlangenmanagers in der Aliasdefinition für die Antwortwarteschlange übereinstimmen, die ebenfalls mit einem Stern (*) markiert ist.

| **Objekt** | **Definition** | |
|---|---|---|
| Lokale Übertragungswarteschlange | QM2 | |
| Definition der fernen Warteschlange | Objektname | Anfrage |
| | Name des fernen Warteschlangenmanagers | QM2 |
| | Name der fernen Warteschlange | Anfrage |
| | Name der Übertragungswarteschlange | WSM2 (DEFAULT) |

| Objekt | Definition | |
|---|---|---|
| Aliasname des Warteschlangenmanagers | Objektname | QM1_relief * |
| | Name des Warteschlangenmanagers | QM1 |
| | Warteschlangenname | (leer) |
| Aliasname der Antwortwarteschlange | Objektname | Antwortaliasname |
| | Name des fernen Warteschlangenmanagers | QM1_relief * |
| | Name der fernen Warteschlange | Antworte |

## Definition des Einreihens auf QM1

Anwendungen füllen die Antwortfelder mit dem Aliasnamen der Warteschlange für Antwortwarteschlangen aus und lassen das Feld für den Namen des WS-Managers leer.

| Feld | Inhalt |
|---|---|
| Warteschlangenname | Anfrage |
| Name des Warteschlangenmanagers | (leer) |
| Name der Empfangswarteschlange für Antworten | Antwortaliasname |
| Warteschlangenmanager für Antwortwarteschlange | (leer) |

## Definitionen, die in diesem Beispiel auf WSM2 verwendet werden

Der IBM MQ-Systemadministrator bei QM2 muss sicherstellen, dass die lokale Warteschlange für die eingehenden Nachrichten vorhanden ist und dass die ordnungsgemäß benannte Übertragungswarteschlange für die Antwortnachricht verfügbar ist.

| Objekt | Definition |
|---|---|
| Lokale Warteschlange | Anfrage |
| Übertragungswarteschlange | QM1_relief |

## Definition in WSM2 definieren

Anwendungen auf QM2 rufen den Namen der Empfangswarteschlange für Antworten und den Namen des WS-Managers aus der ursprünglichen Nachricht ab und verwenden sie, wenn sie die Antwortnachricht in die Warteschlange für Antwortnachrichten einreihen.

| Feld | Inhalt |
|---|---|
| Warteschlangenname | Antworte |
| Name des Warteschlangenmanagers | QM1_relief |

### *Funktionsweise des Beispiels*

Eine Erläuterung des Beispiels und die Verwendung des Aliasnamens der Empfangswarteschlange für Antworten durch den Warteschlangenmanager.

In diesem Beispiel verwenden Requester-Anwendungen auf WSM1 immer 'Answer_alias' als Antwort-Warteschlange in das relevante Feld des put-Aufrufs. Sie rufen ihre Nachrichten immer aus der Warteschlange mit dem Namen 'Answer' ab.

Die Definitionen der Warteschlange für Antwortwarteschlangen sind für die Verwendung durch den QM1-Systemadministrator verfügbar, um den Namen der Antwort-Warteschlange 'Answer' und der Rückgaberoute 'QM1_relief' zu ändern.

Die Änderung des Warteschlangennamens 'Answer' ist in der Regel nicht hilfreich, da die Anwendungen QM1 ihre Antworten in dieser Warteschlange erwarten. Der Systemverwalter QM1 kann jedoch die Rückgaberoute (Serviceklasse) je nach Bedarf ändern.

## Verwendung des Aliasnamens "reply-to queue" des Warteschlangenmanagers

Warteschlangenmanager QM1 ruft die Definitionen aus dem Aliasnamen der Empfangswarteschlange für Antworten ab, wenn der Name der Antwortwarteschlange, die in den von der Anwendung aufgerufenen Aufruf eingeschlossen ist, mit dem Aliasnamen für die Antwortwarteschlange identisch ist, und der Teil des Warteschlangenmanagers leer ist.

Der WS-Manager ersetzt den Namen der Empfangswarteschlange für Antworten in dem Aufruf der Warteschlange mit dem Namen der Warteschlange aus der Definition. Er ersetzt den leeren WS-Manager-Namen im put-Aufruf mit dem Namen des WS-Managers aus der Definition.

Diese Namen werden mit der Nachricht im Nachrichtendeskriptor übertragen.

*Tabelle 17. Aliasname der Antwortwarteschlange*

| Feldname | Put-Aufruf | Übertragungsheader |
|---|---|---|
| Name der Empfangswarteschlange für Antworten | Antwortaliasname | Antworte |
| Name des Antwortwarteschlangenmanagers | (leer) | QM1_relief |

### *Durchlauf der Warteschlange für Antwortwarteschlangen*

Ein Walkthrough des Prozesses von einer Anwendung, die eine Nachricht in eine ferne Warteschlange einreihen soll, bis zu derselben Anwendung, die die Antwortnachricht aus der Warteschlange für Aliasantwortnachrichten entfernt.

Um dieses Beispiel zu vervollständigen, lassen Sie uns den Prozess betrachten.

1. Die Anwendung öffnet eine Warteschlange mit dem Namen 'Abfrage' und reiht Nachrichten in diese Warteschlange ein. Die Anwendung setzt die Felder "reply-to" des Nachrichtendeskriptors auf:

   | Name der Empfangswarteschlange für Antworten | Antwortaliasname |
   |---|---|
   | Name des Antwortwarteschlangenmanagers | (leer) |

2. Der Warteschlangenmanager 'QM1' antwortet auf den Namen des leeren Warteschlangenmanagers, indem er die Definition einer fernen Warteschlangendefinition mit dem Namen 'Answer_alias' (Antwortalias) überprüft. Wenn keine gefunden wird, stellt der Warteschlangenmanager seinen eigenen Namen 'QM1' in das Feld für den Antwortwarteschlangenmanager des Nachrichtendeskriptors.
3. Wenn der Warteschlangenmanager eine Definition einer fernen Warteschlange mit dem Namen 'Answer_alias' findet, extrahiert er den Warteschlangennamen und die Namen des WS-Managers aus der Definition (Warteschlangenname= 'Answer' und WS-Manager-Name = 'QM1_relief'). Anschließend werden sie in die Antwortfelder des Nachrichtendeskriptors gestellt.
4. Der Warteschlangenmanager 'QM1' verwendet die Definition 'Inquiry' für die ferne Warteschlange, um festzustellen, ob sich die beabsichtigte Zielwarteschlange im Warteschlangenmanager 'QM2' befindet, und die Nachricht wird in die Übertragungswarteschlange 'QM2' gestellt. 'QM2' ist der Name

der Standardübertragungswarteschlange für Nachrichten, die für Warteschlangen beim Warteschlangenmanager 'QM2' bestimmt sind.

5. Wenn der WS-Manager 'QM1' die Nachricht in die Übertragungswarteschlange einreiht, wird der Nachricht ein Übertragungsheader hinzugefügt. Dieser Header enthält den Namen der Zielwarteschlange, 'Inquiry' und den Zielwarteschlangenmanager 'QM2'.
6. Die Nachricht wird beim WS-Manager 'QM2' empfangen und in die lokale Warteschlange 'Inquiry' gestellt.
7. Eine Anwendung ruft die Nachricht aus dieser Warteschlange ab und verarbeitet die Nachricht. Die Anwendung bereitet eine Antwortnachricht vor und reiht diese Antwortnachricht aus dem Nachrichtendeskriptor der ursprünglichen Nachricht in den Namen der Antwort auf die Antwort ein:

   | Name der Empfangswarteschlange für Antworten | Antworte |
   |---|---|
   | Name des Antwortwarteschlangenmanagers | QM1_relief |

8. WS-Manager 'QM2' führt den Befehl put aus. Wenn der Warteschlangenmanagername 'QM1_relief' ein ferner Warteschlangenmanager ist, wird die Nachricht in die Übertragungswarteschlange mit dem gleichen Namen 'QM1_relief' versetzt. Die Nachricht erhält einen Übertragungsheader, der den Namen der Zielwarteschlange, 'Answer' und den Zielwarteschlangenmanager 'QM1_relief' enthält.
9. Die Nachricht wird an WS-Manager 'QM1' übertragen. Der Warteschlangenmanager erkennt, dass der WS-Manager-Name 'QM1_relief' ein Aliasname ist. Er extrahiert aus der Aliasdefinition 'QM1_relief' den Namen des physischen Warteschlangenmanagers 'QM1'.
10. WS-Manager 'QM1' reiht dann die Nachricht in den Namen der Warteschlange ein, die im Übertragungsheader 'Answer' enthalten ist.
11. Die Anwendung extrahiert ihre Antwortnachricht aus der Warteschlange 'Answer'.

## Überlegungen zum Netzbetrieb

In einer Umgebung mit verteilten Warteschlangensteuerung, da Nachrichtenziele mit nur einem Warteschlangennamen und einem Warteschlangenmanagernamen adressiert werden, gelten bestimmte Regeln.

1. Wenn der Name des Warteschlangenmanagers angegeben wird und der Name sich von dem Namen des lokalen WS-Managers unterscheidet:
   - Es muss eine Übertragungswarteschlange mit dem gleichen Namen verfügbar sein. Diese Übertragungswarteschlange muss Teil eines Nachrichtenkanals sein, der Nachrichten in einen anderen WS-Manager versetzt, oder
   - Eine WS-Manager-Aliasnamendefinition muss vorhanden sein, um den Namen des Warteschlangenmanagers in denselben Namen oder einen anderen WS-Manager-Namen und eine optionale Übertragungswarteschlange aufzulösen.
   - Wenn der Name der Übertragungswarteschlange nicht aufgelöst werden kann und eine Standardübertragungswarteschlange definiert wurde, wird die Standardübertragungswarteschlange verwendet.
2. Wird nur der Warteschlangenname angegeben, muss eine Warteschlange eines beliebigen Typs mit demselben Namen auf dem lokalen WS-Manager verfügbar sein. Bei dieser Warteschlange kann es sich um eine Definition einer fernen Warteschlange handeln, die in eine Übertragungswarteschlange in einen benachbarten Warteschlangenmanager, einen WS-Manager-Namen und eine optionale Übertragungswarteschlange aufgelöst wird.

Informationen dazu, wie dies in einer Clusterumgebung funktioniert, finden Sie unter Cluster.

z/OS Wenn die Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange (QSG) ausgeführt werden und die gruppeninterne Warteschlangensteuerung (IGQ) aktiviert ist, können Sie Warteschlange SYSTEM.QSG.TRANSMIT.QUEUE verwenden. Weitere Informationen finden Sie unter Gruppeninterne Warteschlangensteuerung.

Betrachten Sie das Szenario eines Nachrichtenkanals, der Nachrichten von einem WS-Manager in einen anderen in einer Umgebung mit verteilten Warteschlangen versetzt.

Die Nachrichten, die verschoben werden, stammen von einem anderen Warteschlangenmanager im Netz, und einige Nachrichten, die einen unbekannten Warteschlangenmanagernamen als Ziel haben, kommen möglicherweise an. Dieses Problem kann auftreten, wenn ein WS-Manager-Name geändert wurde oder beispielsweise aus dem System entfernt wurde.

Das Kanalprogramm erkennt diese Situation, wenn es keine Übertragungswarteschlange für diese Nachrichten finden kann, und stellt die Nachrichten in die Warteschlange für nicht zustellbare Nachrichten (dead-letter). Es liegt in Ihrer Verantwortung, nach diesen Nachrichten zu suchen und die Weiterleitung an die richtige Zieladresse zu veranlassen. Alternativ können Sie sie an den Absender zurückgeben, wo der Urheber ermittelt werden kann.

Ausnahmebedingungsberichte werden unter diesen Umständen generiert, wenn Berichtsnachrichten in der ursprünglichen Nachricht angefordert wurden.

### Namensauflösungskonvention

Die Namensauflösung, die die Identität der Zielwarteschlange (d. a. logisch in eine Änderung des physischen Namens) ändert, tritt nur einmal und nur im Ursprungswarteschlangenmanager auf.

Die nachfolgende Verwendung der verschiedenen Alias-Möglichkeiten darf nur bei der Trennung und Kombination von Nachrichtenflüssen verwendet werden.

## Routing zurückgeben

Nachrichten können eine Rückgabeadresse in Form des Namens einer Warteschlange und eines Warteschlangenmanagers enthalten. Dieses Formular für die Rückgabeadresse kann sowohl in einer Umgebung mit verteilter Warteschlange als auch in einer Clusterumgebung verwendet werden.

Diese Adresse wird normalerweise von der Anwendung angegeben, die die Nachricht erstellt. Sie kann von jeder Anwendung geändert werden, die dann die Nachricht verarbeitet, einschließlich Benutzerexitanwendungen.

Unabhängig von der Quelle dieser Adresse kann jede Anwendung, die die Nachricht bearbeitet, diese Adresse verwenden, um Antworten, Status oder Berichtsnachrichten an die Ursprungsanwendung zurückzugeben.

Die Art und Weise, wie diese Antwortnachrichten weitergeleitet werden, unterscheidet sich nicht von der Art und Weise, wie die ursprüngliche Nachricht weitergeleitet wird. Sie müssen wissen, dass die Nachrichtenflüsse, die Sie zu anderen Warteschlangenmanagern erstellen, entsprechende Rückkehrabläufe benötigen.

### Konflikte mit physischen Namen

Der Name der Zielwarteschlange für die Antwort auf die Warteschlange wurde in den Namen der physischen Warteschlange des ursprünglichen Warteschlangenmanagers aufgelöst. Er darf beim antwortenden Warteschlangenmanager nicht erneut aufgelöst werden.

Es ist eine wahrscheinliche Möglichkeit für Namenskonfliktprobleme, die nur durch eine netzwerkweite Vereinbarung über Namen von physischen und logischen Warteschlangen verhindert werden können.

## Übersetzungen von Warteschlangennamen verwalten

Wenn Sie eine Warteschlangenmanageraliasdefinition oder eine Definition einer fernen Warteschlange erstellen, wird die Namensauflösung für jede Nachricht ausgeführt, die diesen Namen trägt. Diese Situation muss verwaltet werden.

Diese Beschreibung wird für Anwendungsdesigner und Kanalplaner mit einem einzelnen System bereitgestellt, das Nachrichtenkanäle zu benachbarten Systemen hat. Sie nimmt eine lokale Ansicht der Kanalplanung und -steuerung vor.

Wenn Sie eine Warteschlangenmanageraliasdefinition oder eine Definition einer fernen Warteschlange erstellen, wird die Namensauflösung für jede Nachricht ausgeführt, die diesen Namen trägt, unabhängig von der Quelle der Nachricht. Um diese Situation zu überwachen, die möglicherweise eine große Anzahl von Warteschlangen in einem WS-Manager-Netz enthält, behalten Sie die folgenden Tabellen bei:

- Die Namen der Quellenwarteschlangen und der Quellenwarteschlangenmanager in Bezug auf aufgelöste Warteschlangennamen, aufgelöste Warteschlangenmanagernamen und aufgelöste Übertragungswarteschlangennamen mit der Methode der Auflösung.
- Die Namen der Quellenwarteschlangen in Bezug auf:
  - Namen der aufgelösten Zielwarteschlangen
  - Namen von Zielwarteschlangenmanagern aufgelöst
  - Übertragungswarteschlangen
  - Nachrichtenkanalnamen
  - Angrenzende Systemnamen
  - Namen der Antwortwarteschlange

**Anmerkung:** Die Verwendung des Begriffs *Quelle* in diesem Kontext bezieht sich auf den Warteschlangennamen oder den Namen des Warteschlangenmanagers, der von der Anwendung bereitgestellt wird, oder ein Kanalprogramm, wenn eine Warteschlange zum Einreihen von Nachrichten geöffnet wird.

Ein Beispiel für jede dieser Tabellen finden Sie in Tabelle 18 auf Seite 227, Tabelle 19 auf Seite 227 und Tabelle 20 auf Seite 228.

Die Namen in diesen Tabellen werden aus den Beispielen in diesem Abschnitt abgeleitet, und diese Tabelle ist nicht als praktisches Beispiel für die Warteschlangennamensauflösung in einem Knoten gedacht.

*Tabelle 18. Warteschlangennamensauflösung auf WS-Manager QMA*

| **Quellenwarteschlange angegeben, wenn Warteschlange geöffnet wird** | **Der Quellenwarteschlangenmanager wurde beim Öffnen der Warteschlange angegeben.** | **Aufgelöster Warteschlangenname** | **Aufgelöster Name des Warteschlangenmanagers** | **Name der aufgelösten Übertragungswarteschlange** | **Auflösungstyp** |
|---|---|---|---|---|---|
| QA_norm | - | QA_norm | WSMB | WSMB | Ferne Warteschlange |
| (beliebig) | WSMB | - | - | WSMB | (keine) |
| QA_norm | - | QA_norm | WSMB | TX1 | Ferne Warteschlange |
| QB | QMC | QB | WSMD | WSMB | Aliasname des Warteschlangenmanagers |

*Tabelle 19. Warteschlangennamensauflösung auf WS-Manager WSMB*

| **Quellenwarteschlange angegeben, wenn Warteschlange geöffnet wird** | **Der Quellenwarteschlangenmanager wurde beim Öffnen der Warteschlange angegeben.** | **Aufgelöster Warteschlangenname** | **Aufgelöster Name des Warteschlangenmanagers** | **Name der aufgelösten Übertragungswarteschlange** | **Auflösungstyp** |
|---|---|---|---|---|---|
| QA_norm | - | QA_norm | WSMB | - | (keine) |

*Tabelle 19. Warteschlangennamensauflösung auf WS-Manager WSMB (Forts.)*

| **Quellenwarteschlange angegeben, wenn Warteschlange geöffnet wird** | **Der Quellenwarteschlangenmanager wurde beim Öffnen der Warteschlange angegeben.** | **Aufgelöster Warteschlangenname** | **Aufgelöster Name des Warteschlangenmanagers** | **Name der aufgelösten Übertragungswarteschlange** | **Auflösungstyp** |
|---|---|---|---|---|---|
| QA_norm | WSMB | QA_norm | WSMB | - | (keine) |
| QA_norm | QMB_PRIORITÄT | QA_norm | WSMB | - | Aliasname des Warteschlangenmanagers |
| (beliebig) | QMC | (beliebig) | QMC | QMC | (keine) |
| (beliebig) | QMD_norm | (beliebig) | QMD_norm | TX1 | Aliasname des Warteschlangenmanagers |
| (beliebig) | QMD_PRIORITÄT | (beliebig) | QMD_PRIORITÄT | QMD_schnell | Aliasname des Warteschlangenmanagers |
| (beliebig) | QMC_small | (beliebig) | QMC_small | TX_klein | Aliasname des Warteschlangenmanagers |
| (beliebig) | QMC_large | (beliebig) | QMC_large | TX_extern | Aliasname des Warteschlangenmanagers |
| QB_small | QMC | QB_small | QMC | TX_klein | Ferne Warteschlange |
| QB_large | QMC | QB_large | QMC | TX_groß | Ferne Warteschlange |
| (beliebig) | QME | (beliebig) | QME | TX1 | Aliasname des Warteschlangenmanagers |
| QA | QMC | QA | QMC | TX1 | Ferne Warteschlange |
| QB | WSMD | QB | WSMD | TX1 | Ferne Warteschlange |

*Tabelle 20. Umsetzung der Namen der Warteschlange für Antwortwarteschlangen auf WS-Manager QMA*

| **Anwendungsdesign** | | **Antwort-Aliasdefinition** | |
|---|---|---|---|
| **Lokaler QMGR** | **Warteschlangenname für Nachrichten** | **Aliasname der Antwort-Warteschlange** | **Redefined to** |
| QMA | QRR | QR | QRR bei QMA_class1 |

## Nummerierung der Kanalnachrichtenfolge

Der Kanal verwendet Folgenummern, um zu überprüfen, ob Nachrichten in derselben Reihenfolge zugestellt werden, in der sie aus der Übertragungswarteschlange stammen.

Kanalfolgenummern werden überprüft, wenn ein Kanal gestartet wird, und wenn eine Abweichung auftritt, impliziert dies, dass persistente Synchronisationsdaten auf beiden Seiten des Kanals verloren gegangen sind, z. B. eine Disaster-Recovery-Konfiguration (DR), oder dass das Ende der Stapelverarbeitung unterbrochen wurde, als der Kanal unbestätigt war.

Das Zurücksetzen oder Ignorieren von Folgenummernabweichungen, siehe **IgnoreSeqNumberMismatch** in der Zeilengruppe *Channels der Datei qm.ini*, riskiert keinen Verlust oder eine Duplizierung eines Nachrichtenstapels und setzt den unbestätigten Status eines Kanals nicht zurück.

Diese Informationen können mit DISPLAY CHSTATUS angezeigt werden. Die Folgenummer und eine Kennung, die als LUWID bezeichnet wird, werden im persistenten Speicher für die letzte Nachricht gespeichert, die in einem Stapel übertragen wurde. Diese Werte werden beim Kanalstart verwendet, um sicherzustellen, dass beide Enden des Links übereinstimmen, welche Nachrichten erfolgreich übertragen wurden.

### Sequenzielles Abrufen von Nachrichten

Wenn eine Anwendung eine Folge von Nachrichten in dieselbe Zielwarteschlange einreiht, können diese Nachrichten in Folge durch eine ***Single*** -Anwendung mit einer Folge von MQGET-Operationen abgerufen werden, wenn die folgenden Bedingungen erfüllt sind:

- Alle put-Anforderungen wurden von derselben Anwendung ausgeführt.
- Alle put-Anforderungen waren entweder von derselben Arbeitseinheit oder alle gestellten Anforderungen wurden außerhalb einer UO- Unit gestellt.
- Die Nachrichten haben alle dieselbe Priorität.
- Die Nachrichten weisen alle dieselbe Persistenz auf.
- Bei der fernen Warteschlangensteuerung ist die Konfiguration so, dass es nur einen Pfad von der Anwendung gibt, die die put-Anforderung stellt, über ihren Warteschlangenmanager über die übergreifende Kommunikation mit dem Zielwarteschlangenmanager und der Zielwarteschlange.
- Die Nachrichten werden nicht in eine Warteschlange für dead-letter gestellt (z. B. wenn eine Warteschlange temporär voll ist).
- Die Anwendung, die die Nachricht erhält, ändert nicht absichtlich die Reihenfolge des Abrufs, z. B. durch die Angabe eines bestimmten *MsgId* oder *CorrelId* oder durch die Verwendung von Nachrichtenprioritäten.
- Es werden nur eine Anwendung get-Operationen ausgeführt, um die Nachrichten aus der Zielwarteschlange abzurufen. Wenn mehr als eine Anwendung vorhanden ist, müssen diese Anwendungen so konzipiert sein, dass sie alle Nachrichten in jeder Sequenz, die von einer sendenden Anwendung gestellt wird, abrufen können.

**Anmerkung:** Nachrichten von anderen Aufgaben und Arbeitseinheiten können mit der Sequenz interspergt werden, selbst wenn die Sequenz aus einer einzigen Arbeitseinheit einge-stellt wurde.

Wenn diese Bedingungen nicht erfüllt werden können und die Reihenfolge der Nachrichten in der Zielwarteschlange wichtig ist, kann die Anwendung so codiert werden, dass sie ihre eigene Nachrichtenfolgenummer als Teil der Nachricht verwendet, um die Reihenfolge der Nachrichten zu gewährleisten.

### Sequenz des Abrufs von schnellen, nicht persistenten Nachrichten

Nicht persistente Nachrichten in einem schnellen Kanal können persistente Nachrichten auf demselben Kanal überdauern und so aus der Reihenfolge kommen. Der empfangende MCA reiht die nicht persistenten Nachrichten sofort in die Zielwarteschlange ein und macht sie sichtbar. Persistente Nachrichten werden bis zum nächsten Synchronisationspunkt nicht sichtbar gemacht.

## Loopback-Tests

*Loopback-Tests* ist eine Technik auf Nicht-z/OS-Plattformen, mit der Sie eine Datenübertragungsverbindung testen können, ohne tatsächlich eine Verbindung zu einer anderen Maschine herstellen zu müssen.

Sie haben eine Verbindung zwischen zwei WS-Managern eingerichtet, als ob sie sich auf separaten Maschinen befinden, aber Sie testen die Verbindung, indem Sie eine Schleife zu einem anderen Prozess auf derselben Maschine zurückführen. Dieses Verfahren bedeutet, dass Sie Ihren Kommunikationscode testen können, ohne dass ein aktives Netz erforderlich ist.

Die Art und Weise, in der Sie dies tun, hängt davon ab, welche Produkte und Protokolle Sie verwenden.

Auf Windows-Systemen können Sie den "loopback"-Adapter verwenden.

Weitere Informationen finden Sie in der Dokumentation zu den Produkten, die Sie verwenden.

### Trace-und Aktivitätsaufzeichnung weiterleiten

Sie können die Weiterleitung einer Nachricht durch eine Reihe von Warteschlangenmanagern auf zwei Arten bestätigen.

Sie können die IBM MQ-Anwendung zur Routenanzeige, die über den Steuerbefehl **dspmqrte** verfügbar ist, oder die Aktivitätsaufzeichnung verwenden. Beide Themen werden im Abschnitt Referenzinformationen zur Überwachung beschrieben.

## Einführung in die verteilte Warteschlangenverwaltung

DQM (Distributed Queue Management) wird zum Definieren und Steuern der Kommunikation zwischen Warteschlangenmanagern verwendet.

Verwaltung verteilter Warteschlangen:

- Ermöglicht es Ihnen, Kommunikationskanäle zwischen Warteschlangenmanagern zu definieren und zu steuern.
- Stellt einen Nachrichtenkanaldienst zur Verfügung, mit dem Nachrichten von einem Typ von *lokaler Warteschlange* , der als Übertragungswarteschlange bezeichnet wird, in Kommunikationsverbindungen auf einem lokalen System und von Kommunikationsverbindungen zu lokalen Warteschlangen an einem Zielwarteschlangenmanager verschoben werden können.
- Stellt Funktionen zur Überwachung des Betriebs von Kanälen und zur Diagnose von Problemen mit Hilfe von Anzeigen, Befehlen und Programmen bereit.

Kanaldefinitionen ordnen Kanalnamen zu Übertragungswarteschlangen, Kommunikationsverbindungskennungen und Kanalattributen zu. Kanaldefinitionen werden auf verschiedenen Plattformen auf unterschiedliche Weise implementiert. Das Senden und Empfangen von Nachrichten wird von Programmen gesteuert, die als *Nachrichtenkanalagenten* (MCAs) bezeichnet werden, die die Kanaldefinitionen zum Starten und Steuern der Kommunikation verwenden.

Die MCAs wiederum werden von DQM selbst gesteuert. Die Struktur ist plattformabhängig, enthält jedoch in der Regel Empfangsprogramme und Auslösemonitore sowie Bedienerbefehle und Anzeigen.

Ein *Nachrichtenkanal* ist eine Einwegpipe für das Versetzen von Nachrichten von einem WS-Manager in einen anderen. Somit weist ein Nachrichtenkanal zwei Endpunkte auf, die durch ein MCAs-Paar dargestellt werden. Jeder Endpunkt verfügt über eine Definition seines Endes des Nachrichtenkanals. Beispiel: Ein Ende würde einen Sender, das andere Ende einen Empfänger definieren.

Weitere Informationen zum Definieren von Kanälen finden Sie unter:

- ALW „Kanäle in AIX, Linux, and Windows überwachen und steuern” auf Seite 264
- z/OS „Kanäle in z/OS überwachen und steuern” auf Seite 1055
- IBM i „Kanäle in IBM i überwachen und steuern” auf Seite 289

Informationen zu den Planungsbeispielen für Nachrichtenkanäle finden Sie unter:

- ALW Beispiel für Nachrichtenkanalplanung für AIX, Linux, and Windows
- IBM i Beispiel für Nachrichtenkanalplanung für IBM i

- z/OS Beispiel für Nachrichtenkanalplanung für z/OS
- z/OS Beispiel für Nachrichtenkanalplanung für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange

Informationen zu Kanalexits finden Sie unter Channel-Exit-Programme für Messaging-Kanäle .

**Zugehörige Konzepte**

„Senden und Empfangen von Nachrichten” auf Seite 231
Die folgende Abbildung zeigt das Management des verteilten Warteschlangenmanagements mit Details zu den Beziehungen zwischen Entitäten, wenn Nachrichten übertragen werden. Es zeigt auch den Ablauf der Steuerung an.

„Kanalsteuerfunktion” auf Seite 239
Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

„Was passiert, wenn eine Nachricht nicht zugestellt werden kann?” auf Seite 254
Wenn eine Nachricht nicht zugestellt werden kann, kann der MCA sie auf mehrere Arten verarbeiten. Sie kann es erneut versuchen, sie kann an den Absender zurückkehren oder sie in die Warteschlange für dead-Mail setzen.

„Initialisierungs- und Konfigurationsdateien” auf Seite 260
Die Verarbeitung von Kanalinitialisierungsdaten hängt von Ihrer IBM MQ-Plattform ab.

„Datenkonvertierung für Nachrichten” auf Seite 261
IBM MQ-Nachrichten erfordern möglicherweise eine Datenkonvertierung, wenn sie zwischen Warteschlangen in verschiedenen Warteschlangenmanagern gesendet werden.

„Schreiben eigener Nachrichtenkanalagenten” auf Seite 261
IBM MQ ermöglicht es Ihnen, Ihre eigenen Nachrichtenkanalagenten-Programme (MCA) zu schreiben oder ein Programm von einem unabhängigen Softwareanbieter zu installieren.

„Andere Aspekte, die für die verteilte Warteschlangenverwaltung zu berücksichtigen sind” auf Seite 262
Weitere Themen, die bei der Vorbereitung von IBM MQ für die verteilte Warteschlangenverwaltung zu berücksichtigen sind. In diesem Abschnitt finden Sie Informationen zu Nicht zugestellten Nachrichtenwarteschlangen, Warteschlangen in Verwendung, Systemerweiterungen und Benutzerexitprogrammen sowie zur Ausführung von Kanälen und Empfangsprogrammen als vertrauenswürdige Anwendungen.

**Zugehörige Verweise**

Beispielkonfigurationsdaten

## Senden und Empfangen von Nachrichten

Die folgende Abbildung zeigt das Management des verteilten Warteschlangenmanagements mit Details zu den Beziehungen zwischen Entitäten, wenn Nachrichten übertragen werden. Es zeigt auch den Ablauf der Steuerung an.

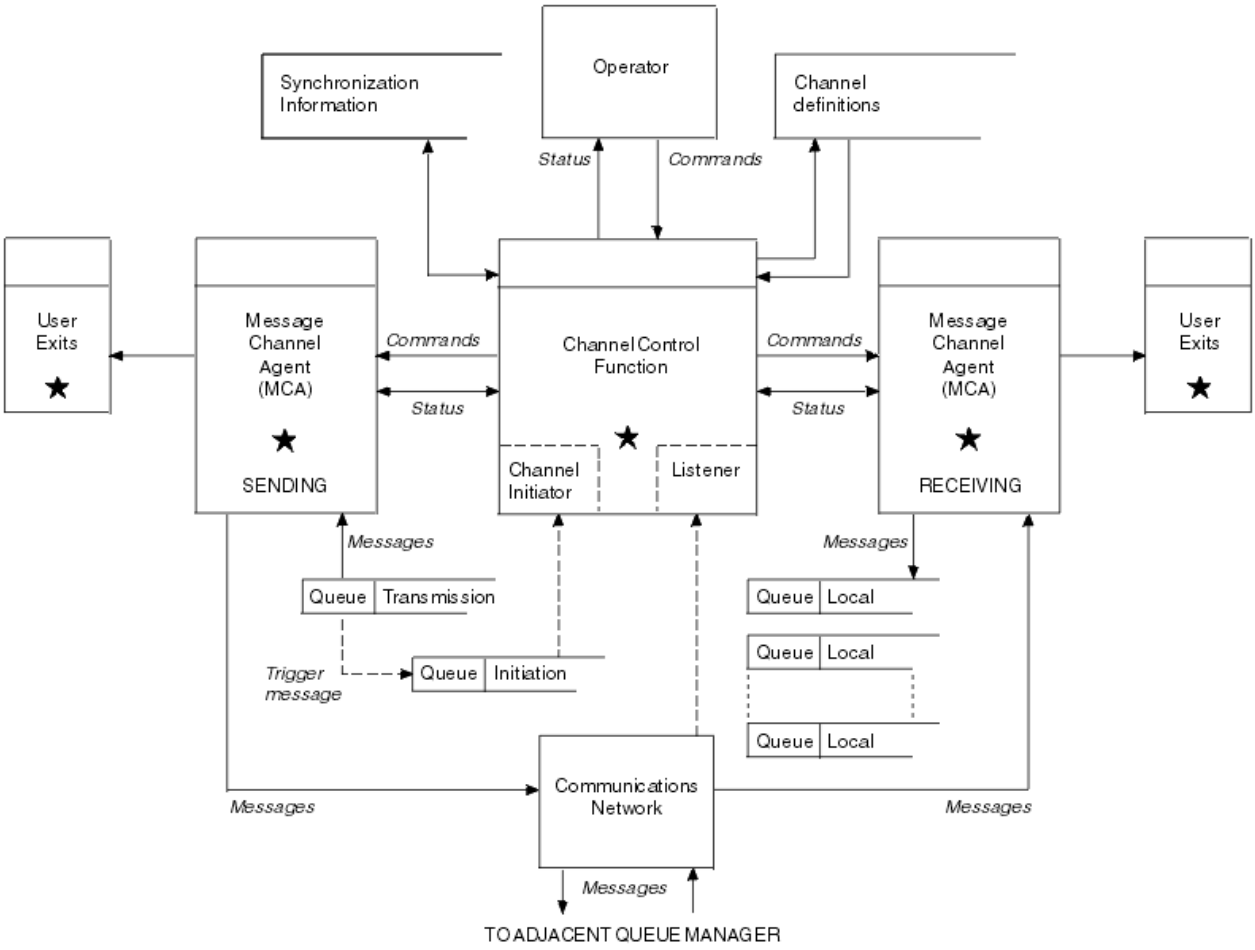



*Abbildung 15. Verteiltes Warteschlangenmanagementmodell*

**Anmerkung:**

1. Abhängig von der Plattform gibt es je einen MCA pro Kanal. Es können eine oder mehrere Kanalsteuerungsfunktionen für einen bestimmten Warteschlangenmanager vorhanden sein.
2. Die Implementierung von MCAs und Kanalsteuerungsfunktionen ist stark plattformabhängig. Sie können Programme oder Prozesse oder Threads sein, und sie können eine einzelne Entität oder viele verschiedene unabhängige oder verknüpfte Teile umfassen.
3. Alle Komponenten, die mit einem Stern gekennzeichnet sind, können die MQI verwenden.

## Kanalparameter

Ein MCA erhält seine Parameter auf eine der folgenden Arten:

- Wird der Kanalname durch einen Befehl gestartet, wird er in einem Datenbereich übergeben. Der MCA liest dann die Kanaldefinition direkt, um die zugehörigen Attribute abzurufen.
- Der MCA kann für den Sender und in einigen Fällen Serverkanäle automatisch vom Warteschlangenmanager des Warteschlangenmanagers gestartet werden. Der Kanalname wird, sofern zutreffend, aus der Auslöserprozessdefinition abgerufen und an den MCA übergeben. Die übrige Verarbeitung ist mit der zuvor beschriebenen Verarbeitung identisch. Serverkanäle müssen nur so konfiguriert werden, dass sie ausgelöst werden, wenn sie vollständig qualifiziert sind, d. a. sie geben einen CONNAME an, zu dem eine Verbindung hergestellt werden soll.
- Wenn der Kanalname über Remotezugriff von einem Sender-, Server-, Requester-oder Clientanschluss gestartet wird, wird der Kanalname in den Anfangsdaten vom Partner-Nachrichtenkanalagenten übergeben. Der MCA liest die Kanaldefinition direkt, um die zugehörigen Attribute abzurufen.

Bestimmte Attribute, die in der Kanaldefinition nicht definiert sind, sind ebenfalls verhandelbar:

**Nachrichten teilen**
Wenn ein Ende keine geteilten Nachrichten unterstützt, werden die geteilten Nachrichten nicht gesendet.

**Konvertierungsfunktion**
Wenn ein Ende die erforderliche Codepagekonvertierung oder die Konvertierung der numerischen Codierung bei Bedarf nicht ausführen kann, muss das andere Ende sie verarbeiten. Wenn das Ende nicht unterstützt wird, kann der Kanal bei Bedarf nicht gestartet werden.

**Unterstützung Verteilerliste**
Wenn ein Ende Verteilerlisten nicht unterstützt, setzt der Partner-MCA ein Flag in seiner Übertragungswarteschlange, so dass es Nachrichten abfangen kann, die für mehrere Ziele bestimmt sind.

### Kanalstatus und Folgenummern

Nachrichtenkanalagentenprogramme führen Datensätze der aktuellen Folgenummer und der logischen Arbeitseinheit für jeden Kanal und des allgemeinen Status des Kanals auf. Auf einigen Plattformen können Sie diese Statusinformationen anzeigen, um die Kanäle zu steuern.

## So senden Sie eine Nachricht an einen anderen Warteschlangenmanager

In diesem Abschnitt wird die einfachste Methode zum Senden einer Nachricht zwischen Warteschlangenmanagern beschrieben, einschließlich der erforderlichen Voraussetzungen und Berechtigungen. Es können auch andere Methoden verwendet werden, um Nachrichten an einen fernen Warteschlangenmanager zu senden.

Bevor Sie eine Nachricht von einem WS-Manager an einen anderen Warteschlangenmanager senden, müssen Sie die folgenden Schritte ausführen:

1. Überprüfen Sie, ob Ihr ausgewähles Kommunikationsprotokoll verfügbar ist.
2. Starten Sie die Warteschlangenmanager.
3. Starten Sie die Kanalinitiatoren.
4. Starten Sie die Empfangsprogramme.

Sie müssen außerdem auch die richtige IBM MQ-Sicherheitsberechtigung haben, um die erforderlichen Objekte zu erstellen.

Gehen Sie wie folgt vor, um Nachrichten von einem WS-Manager an einen anderen zu

- Definieren Sie die folgenden Objekte auf dem Quellenwarteschlangenmanager:
  - Senderkanal
  - Definition der fernen Warteschlange
  - Initialisierungswarteschlange ( z/OS erforderlich unter z/OS, ansonsten optional)
  - Übertragungswarteschlange
  - Warteschlange für nicht zustellbare Nachrichten
- Definieren Sie die folgenden Objekte auf dem Ziel-WS-Manager:
  - Empfängerkanal
  - Zielwarteschlange
  - Warteschlange für nicht zustellbare Nachrichten

Je nach Ihrer IBM MQ-Plattform können Sie zum Definieren dieser Objekte mehrere verschiedene Methoden verwenden:

- Auf allen Plattformen können Sie die IBM MQ-Scriptbefehle (MQSC) verwenden, die im Abschnitt MQSC-Befehle beschrieben sind, die im Abschnitt Verwaltungstasks automatisieren beschriebenen Befehle des programmierbaren Befehlsformats (PCF) oder den IBM MQ-Explorer.

- z/OS Unter z/OS können Sie auch die im Abschnitt IBM MQ for z/OS verwalten beschriebenen Betriebs- und Steuerkonsolen verwenden.
- IBM i Unter IBM i können Sie auch die Anzeigenschnittstelle verwenden.

Weitere Informationen zum Erstellen der Komponenten zum Senden von Nachrichten an einen anderen WS-Manager finden Sie in den folgenden Unterabschnitten:

**Zugehörige Konzepte**

„Verteilte Warteschlangenverfahren in IBM MQ" auf Seite 207
In den Unterabschnitten in diesem Abschnitt werden die Verfahren beschrieben, die bei der Planung von Kanälen verwendet werden. In diesen Unterabschnitten werden Verfahren beschrieben, mit deren Hilfe Sie planen, wie die Warteschlangenmanager miteinander verbunden werden, und den Fluss von Nachrichten zwischen Ihren Anwendungen verwalten.

„Einführung in die verteilte Warteschlangenverwaltung" auf Seite 230
DQM (Distributed Queue Management) wird zum Definieren und Steuern der Kommunikation zwischen Warteschlangenmanagern verwendet.

„Ausgelöste Kanäle" auf Seite 256
IBM MQ stellt eine Funktion bereit, um eine Anwendung automatisch zu starten, wenn bestimmte Bedingungen in einer Warteschlange erfüllt sind. Diese Funktion wird als Triggerung bezeichnet.

„Sicherheit von Nachrichten" auf Seite 253
Zusätzlich zu den typischen Wiederherstellungsfunktionen von IBM MQ stellt das verteilte Warteschlangenmanagement sicher, dass Nachrichten ordnungsgemäß zugestellt werden, indem eine Synchronisationspunktprozedur verwendet wird, die zwischen den beiden Enden des Nachrichtenkanals koordiniert wird. Wenn diese Prozedur einen Fehler feststellt, wird der Kanal geschlossen, so dass Sie das Problem untersuchen und die Nachrichten sicher in der Übertragungswarteschlange behalten können, bis der Kanal erneut gestartet wird.

**Zugehörige Tasks**

„Warteschlangenmanager auf Multiplatforms erstellen" auf Seite 7
Bevor Sie Nachrichten und Warteschlangen verwenden können, müssen Sie mindestens einen WS-Manager und die zugehörigen Objekte erstellen und starten. Ein Warteschlangenmanager verwaltet die Ressourcen, die ihm zugeordnet sind, insbesondere die Warteschlangen, die er besitzt. Er stellt Warteschlangenservices für Anwendungen für MQI-Aufrufe (Message Queuing Interface) und Befehle zum Erstellen, Ändern, Anzeigen und Löschen von IBM MQ-Objekten bereit.

„Kanäle in AIX, Linux, and Windows überwachen und steuern" auf Seite 264
Für DQM müssen Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Sie können Kanäle mit Befehlen, Programmen, IBM MQ Explorer, Dateien für die Kanaldefinitionen und einem Speicherbereich für Synchronisationsinformationen steuern.

„Kanäle in IBM i überwachen und steuern" auf Seite 289
Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Jeder WS-Manager verfügt über ein DQM-Programm zur Steuerung von Verbindungen zu kompatiblen fernen Warteschlangenmanagern.

„Verbindungen zwischen Client und Server konfigurieren" auf Seite 15
Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

„WS-Manager-Cluster konfigurieren" auf Seite 312
Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Sie können Cluster-Komponenten definieren und Cluster erstellen und verwalten.

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten" auf Seite 1052
In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

### *Kanäle definieren*

Wenn Sie Nachrichten von einem WS-Manager an einen anderen Warteschlangenmanager senden möchten, müssen Sie zwei Kanäle definieren. Sie müssen einen Kanal auf dem Quellenwarteschlangenmanager und einen Kanal auf dem Zielwarteschlangenmanager definieren.

**Auf dem Quellenwarteschlangenmanager**

Definieren Sie einen Kanal mit dem Kanaltyp SENDER. Sie müssen Folgendes angeben:

- Der Name der Übertragungswarteschlange, die verwendet werden soll (das Attribut XMITQ).
- Der Verbindungsname des Partnersystems (Attribut CONNAME).
- Der Name des Kommunikationsprotokolls, das Sie verwenden (Attribut TRPTYPE). Unter IBM MQ for z/OS muss das Protokoll TCP oder LU6.2 sein. Auf anderen Plattformen müssen Sie dies nicht angeben. Sie können den Wert aus der Standardkanaldefinition übernehmen lassen.

Details zu allen Kanalattributen werden in Kanalattribute angegeben.

**Auf dem Zielwarteschlangenmanager**

Definieren Sie einen Kanal mit einem Kanaltyp von RECEIVER und denselben Namen wie der Senderkanal.

Den Namen des von Ihnen verwendeten Übertragungsprotokolls (Attribut TRPTYPE) angeben. Unter IBM MQ for z/OS muss das Protokoll TCP oder LU6.2 sein. Auf anderen Plattformen müssen Sie dies nicht angeben. Sie können den Wert aus der Standardkanaldefinition übernehmen lassen.

Empfängerkanaldefinitionen können generisch sein. Dies bedeutet Folgendes: Wenn Sie mehrere Warteschlangenmanager mit demselben Empfänger kommunizieren, können die sendenden Kanäle alle denselben Namen für den Empfänger angeben, und es gilt eine Empfängerdefinition für alle.

Wenn Sie den Kanal definiert haben, können Sie ihn mit dem Befehl PING CHANNEL testen. Mit diesem Befehl wird eine spezielle Nachricht vom Senderkanal an den Empfängerkanal gesendet und überprüft, ob die Nachricht zurückgegeben wird.

**Anmerkung:** Der Wert des Parameters TRPTYPE wird vom antwortenden Nachrichtenkanalagenten ignoriert. Ein TRPTYPE mit dem Wert 'TCP' in der Senderkanaldefinition beispielsweise wird erfolgreich mit einem TRPTYPE-Wert 'LU62' in der Empfängerkanaldefinition als Partner gestartet.

### *Definieren der Warteschlangen*

Wenn Sie Nachrichten von einem Warteschlangenmanager an einen anderen Warteschlangenmanager senden möchten, müssen Sie bis zu sechs Warteschlangen definieren. Sie müssen bis zu vier Warteschlangen auf dem Quellenwarteschlangenmanager und bis zu zwei Warteschlangen auf dem Ziel-WS-Manager definieren.

**Auf dem Quellenwarteschlangenmanager**

- Definition der fernen Warteschlange

  Geben Sie in dieser Definition Folgendes an:

  **Name des fernen Warteschlangenmanagers**
  Der Name des Zielwarteschlangenmanagers.

  **Name der fernen Warteschlange**
  Der Name der Zielwarteschlange auf dem Ziel-WS-Manager.

  **Name der Übertragungswarteschlange**
  Der Name der Übertragungswarteschlange. Sie müssen diesen Namen der Übertragungswarteschlange nicht angeben. Ist dies nicht der Fall, wird eine Übertragungswarteschlange mit demselben Namen wie der Zielwarteschlangenmanager verwendet. Wenn diese Option nicht vorhanden ist, wird die Standardübertragungswarteschlange verwendet. Es wird empfohlen, der Übertragungswarteschlange denselben Namen wie der Zielwarteschlangenmanager zu geben, so dass die Warteschlange standardmäßig gefunden wird.

- Definition der Initialisierungswarteschlange

z/OS Dies ist erforderlich. Sie müssen die Initialisierungswarteschlange mit dem Namen SYSTEM.CHANNEL.INITQ. verwenden.

Multi Dies ist optional. Überlegen Sie die Benennung der Initialisierungswarteschlange SYSTEM.CHANNEL.INITQ.

- Definition der Übertragungswarteschlange

  Eine lokale Warteschlange, bei der das Attribut USAGE auf XMITQ gesetzt ist. IBM i Wenn Sie die native IBM MQ for IBM i-Schnittstelle verwenden, lautet das USAGE-Attribut *TMQ.
- Definition der Warteschlange für nicht zustellbare

  Definieren Sie eine Warteschlange für dead-Mail, in die unzustellbare Nachrichten geschrieben werden können.

**Auf dem Zielwarteschlangenmanager**

- Definition der lokalen Warteschlange

  Die Zielwarteschlange. Der Name dieser Warteschlange muss mit dem Namen dieser Warteschlange übereinstimmen, der im Feld für den fernen Warteschlangennamen der Definition der fernen Warteschlange im Quellenwarteschlangenmanager angegeben ist.
- Definition der Warteschlange für nicht zustellbare

  Definieren Sie eine Warteschlange für dead-Mail, in die unzustellbare Nachrichten geschrieben werden können.

**Zugehörige Konzepte**

„Übertragungswarteschlange erstellen” auf Seite 236
Bevor ein Kanal (außer einem Requesterkanal) gestartet werden kann, muss die Übertragungswarteschlange wie in diesem Abschnitt beschrieben definiert werden. Die Übertragungswarteschlange muss in der Kanaldefinition angegeben werden.

„Übertragungswarteschlange unter IBM i erstellen” auf Seite 237
Sie können eine Übertragungswarteschlange auf der IBM i-Plattform erstellen, indem Sie die Anzeige "MQM-Warteschlange erstellen" verwenden.

*Übertragungswarteschlange erstellen*

Bevor ein Kanal (außer einem Requesterkanal) gestartet werden kann, muss die Übertragungswarteschlange wie in diesem Abschnitt beschrieben definiert werden. Die Übertragungswarteschlange muss in der Kanaldefinition angegeben werden.

Definieren Sie für jeden sendenden Nachrichtenkanal eine lokale Warteschlange mit dem Attribut USAGE, das auf XMITQ gesetzt ist. Wenn Sie eine bestimmte Übertragungswarteschlange in Ihren fernen Warteschlangendefinitionen verwenden möchten, erstellen Sie eine ferne Warteschlange wie angezeigt.

Um eine Übertragungswarteschlange zu erstellen, verwenden Sie die IBM MQ-Befehle (MQSC), wie in den folgenden Beispielen gezeigt:

**Beispiel für die Erstellung einer Übertragungswarteschlange**

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

**Beispiel für eine ferne Warteschlange erstellen**

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

Erwäge die Benennung der Übertragungswarteschlange den Namen des WS-Managers auf dem fernen System, wie in den Beispielen gezeigt.

*Übertragungswarteschlange unter IBM i erstellen*
Sie können eine Übertragungswarteschlange auf der IBM i-Plattform erstellen, indem Sie die Anzeige "MQM-Warteschlange erstellen" verwenden.

Sie müssen für jeden sendenden Nachrichtenkanal eine lokale Warteschlange mit dem Attribut "Nutzungsfeld" definieren, das auf *TMQ gesetzt ist.

Sollen ferne Warteschlangendefinitionen verwendet werden, denselben Befehl verwenden, um eine Warteschlange mit der Art *RMT und die Verwendung von *NORMAL zu erstellen.

Wenn Sie eine Übertragungswarteschlange erstellen möchten, verwenden Sie den Befehl CRTMQMQ in der Befehlszeile, um die erste Anzeige für die Warteschlangenerstellung anzuzeigen. Weitere Informationen finden Sie im Abschnitt Abbildung 16 auf Seite 237.

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . .

Queue type . . . . . .  ____     *ALS, *LCL, *MDL, *RMT

Message Queue Manager name . . .  *DFT_____________________________
_____









Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
+
```

*Abbildung 16. Erstellen Sie eine Warteschlange (1).*

Geben Sie den Namen der Warteschlange ein, und geben Sie den Typ der Warteschlange an, die Sie erstellen wollen: Lokal, Fern oder Alias. Geben Sie für eine Übertragungswarteschlange die Option Lokal ( *LCL) in dieser Anzeige an, und drücken Sie die Eingabetaste.

Sie werden mit der zweiten Seite der Anzeige 'Create MQM Queue' (MQM-Warteschlange erstellen) angezeigt (siehe Abbildung 17 auf Seite 238).

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . > HURS.2.HURS.PRIORIT

Queue type . . . . . > *LCL     *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . .  *DFT
Replace . . . . . .  *NO      *NO, *YES
Text 'description' . . . . . .  '                    '
Put enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Default message priority . . . .  0       0-9, *SYSDFTQ
Default message persistence . .  *NO      *SYSDFTQ, *NO, *YES
Process name . . . . . .  '                    '
Triggering enabled . . . . . .  *NO      *SYSDFTQ, *NO, *YES
Get enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Sharing enabled . . . . . .  *YES     *SYSDFTQ, *NO, *YES



More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 17. Erstellen Sie eine Warteschlange (2).*

Ändern Sie einen der angezeigten Standardwerte. Drücken Sie die Taste zum Blättern, um zur nächsten Anzeige zu blättern (siehe Abbildung 18 auf Seite 238).

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Default share option . . . . . .  *YES     *SYSDFTQ, *NO, *YES
Message delivery sequence . . .  *PTY     *SYSDFTQ, *PTY, *FIFO
Harden backout count . . . . . .  *NO      *SYSDFTQ, *NO, *YES
Trigger type . . . . . .  *FIRST    *SYSDFTQ, *FIRST, *ALL...
Trigger depth . . . . . .  1       1-999999999, *SYSDFTQ
Trigger message priority . . . .  0       0-9, *SYSDFTQ
Trigger data . . . . . .  '                    '
Retention interval . . . . . .  999999999   0-999999999, *SYSDFTQ
Maximum queue depth . . . . . .  5000     1-24000, *SYSDFTQ
Maximum message length . . . . .  4194304    0-4194304, *SYSDFTQ
Backout threshold . . . . . .  0       0-999999999, *SYSDFTQ
Backout requeue queue . . . . .  '                    '
Initiation queue . . . . . .  '                    '



More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 18. Erstellen Sie eine Warteschlange (3).*

Geben Sie *TMQ für die Übertragungswarteschlange im Feld Verwendung in dieser Anzeige ein, und ändern Sie die Standardwerte, die in den anderen Feldern angezeigt werden.

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Usage . . . . . . . . . . . . . .  *TMQ     *SYSDFTQ, *NORMAL, *TMQ
Queue depth high threshold . . .  80      0-100, *SYSDFTQ
Queue depth low threshold . . .  20      0-100, *SYSDFTQ
Queue full events enabled . . .  *YES     *SYSDFTQ, *NO, *YES
Queue high events enabled . . .  *YES     *SYSDFTQ, *NO, *YES
Queue low events enabled . . . .  *YES     *SYSDFTQ, *NO, *YES
Service interval . . . . . . . .  999999999   0-999999999, *SYSDFTQ
Service interval events . . . .  *NONE     *SYSDFTQ, *HIGH, *OK, *NONE
Distribution list support . . .  *NO      *SYSDFTQ, *NO, *YES
Cluster Name . . . . . . . . . .  *SYSDFTQ
Cluster Name List . . . . . . .  *SYSDFTQ
Default Binding . . . . . . . .  *SYSDFTQ   *SYSDFTQ, *OPEN, *NOTFIXED




Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 19. Erstellen Sie eine Warteschlange (4).*

Wenn Sie sicher sind, dass die Felder die richtigen Daten enthalten, drücken Sie die Eingabetaste, um die Warteschlange zu erstellen.

### *Kanal starten*

Wenn Sie Nachrichten in die ferne Warteschlange einlegen, die auf dem Quellenwarteschlangenmanager definiert ist, werden sie in der Übertragungswarteschlange gespeichert, bis der Kanal gestartet wird. Wenn der Kanal gestartet wurde, werden die Nachrichten an die Zielwarteschlange auf dem fernen WS-Manager zugestellt.

Starten Sie den Kanal auf dem sendenden Warteschlangenmanager mit dem Befehl START CHANNEL. Wenn Sie den sendenden Kanal starten, wird der empfangende Kanal automatisch gestartet (durch den Listener) und die Nachrichten werden an die Zielwarteschlange gesendet. Beide Enden des Nachrichtenkanals müssen aktiv sein, damit Nachrichten übertragen werden können.

Da sich die beiden Kanalenden auf verschiedenen Warteschlangenmanagern befinden, konnten sie mit anderen Attributen definiert werden. Um Unterschiede zu beheben, gibt es beim Start des Kanals eine erste Datenverhandlung zwischen den beiden Enden. Im Allgemeinen arbeiten die beiden Enden des Kanals mit den Attributen, die die weniger Ressourcen benötigen. Auf diese Weise können größere Systeme die geringeren Ressourcen kleinerer Systeme am anderen Ende des Nachrichtenkanals aufnehmen.

Der sendende MCA teilt große Nachrichten auf, bevor er sie über den Kanal sendet. Sie werden auf dem fernen Warteschlangenmanager erneut assembliert assemblitiert. Dies ist für den Benutzer nicht erkennbar.

Ein MCA kann Nachrichten mit mehreren Threads übertragen. Dieser Prozess, der als *Pipelining* bezeichnet wird, ermöglicht es dem MCA, Nachrichten effizienter zu übertragen, wobei weniger Wartestatus vorhanden sind. Pipelining verbessert die Kanalleistung.

## Kanalsteuerfunktion

Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

Befehle werden über Anzeigen, Programme oder von einer Befehlszeile an die Kanalsteuerfunktion ausgegeben. In der Anzeige-Schnittstelle werden auch Kanalstatus-und Kanaldefinitionsdaten angezeigt. Sie können Programmable Command Formats oder diese IBM MQ-Befehle (MQSC) und Steuerbefehle verwenden, die in „Kanäle in AIX, Linux, and Windows überwachen und steuern” auf Seite 264 ausführlich beschrieben sind.

Die Befehle fallen in die folgenden Gruppen:

- Kanalverwaltung
- Kanalsteuerung
- Kanalstatusüberwachung

Kanalverwaltungsbefehle befassen sich mit den Definitionen der Kanäle. Sie ermöglichen Ihnen Folgendes:

- Kanaldefinition erstellen
- Kanaldefinition kopieren
- Kanaldefinition ändern
- Kanaldefinition löschen

Kanalsteuerungsbefehle verwalten den Betrieb der Kanäle. Sie ermöglichen Ihnen Folgendes:

- Kanal starten
- Kanal stoppen
- Erneutes Synchronisieren mit Partner (in einigen Implementierungen)
- Nachrichtenfolgenummern zurücksetzen
- Auflösen eines unbestäschen Nachrichtenstaufes von Nachrichten
- Ping; Senden einer Testkommunikation über den Kanal

Die Kanalüberwachung zeigt den Status der Kanäle an, z. B.:

- Aktuelle Kanaleinstellungen
- Ob der Kanal aktiv oder inaktiv ist
- Gibt an, ob der Kanal in einem synchronisierten Status beendet wurde

**Zugehörige Konzepte**
Wo finden Sie Informationen zur Problembestimmung?

### *Vorbereiten von Kanälen*

Bevor Sie versuchen, einen Nachrichtenkanal oder einen MQI-Kanal zu starten, müssen Sie den Kanal vorbereiten. Sie müssen sicherstellen, dass alle Attribute der lokalen und fernen Kanaldefinitionen korrekt und kompatibel sind.

In Kanalattribute werden die Kanaldefinitionen und Attribute beschrieben.

Obwohl Sie explizite Kanaldefinitionen konfiguriert haben, können die Kanalverhandlungen, die bei einem Kanalstart ausgeführt werden, einen oder einen anderen der definierten Werte überschreiben. Dieses Verhalten ist normal und nicht für den Benutzer erkennbar und wurde auf diese Weise so angeordnet, dass ansonsten inkompatible Definitionen zusammenarbeiten können.

## Automatische Definition von Empfänger-und Serververbindungskanälen

Wenn keine geeignete Kanaldefintion vorhanden ist, wird in IBM MQ auf allen Plattformen außer z/OS für einen Empfänger- oder Serververbindungskanal mit aktivierter automatischer Definition automatisch eine Definition erstellt. Die Definition wird wie folgt erstellt:

1. Die entsprechende Modellkanaldefinition, SYSTEM.AUTO.RECEIVER oder SYSTEM.AUTO.SVRCONN. Die Modellkanaldefinitionen für die automatische Definition sind mit den Systemstandardwerten SYSTEM.DEF.RECEIVER und SYSTEM.DEF.SVRCONN identisch, mit Ausnahme des Beschreibungsfelds, das

"Auto-defined by" ist, gefolgt von 49 Leerzeichen. Der Systemadministrator kann einen beliebigen Teil der bereitgestellten Modellkanaldefinitionen ändern.

2. Informationen aus dem Partnersystem. Die Werte aus dem Partner werden für den Kanalnamen und den Folgenummernumbruch verwendet.
3. Ein Kanalexitprogramm, das Sie zum Ändern der Werte verwenden können, die von der automatischen Definition erstellt wurden. Siehe Exitprogramm für die automatische Kanaldefinition (Channel Auto-Definition)

Anschließend wird die Beschreibung geprüft, um festzustellen, ob sie durch einen Exit für die automatische Definition geändert wurde oder weil die Modelldefinition geändert wurde. Wenn die ersten 44 Zeichen immer noch "Automatisch definiert durch" gefolgt von 29 Leerzeichen sind, wird der Name des WS-Managers hinzugefügt. Wenn die letzten 20 Zeichen noch immer alle Leerzeichen sind, werden die lokale Uhrzeit und das Datum hinzugefügt.

Wenn die Definition erstellt und gespeichert wurde, beginnt der Kanalstart so, als ob die Definition immer vorhanden war. Die Stapelgröße, die Übertragungsgröße und die Nachrichtengröße werden mit dem Partner ausgehandelt.

## Andere Objekte definieren

Bevor ein Nachrichtenkanal gestartet werden kann, müssen beide Enden auf ihren Warteschlangenmanagern definiert sein (oder für die automatische Definition aktiviert sein). Die Übertragungswarteschlange, die sie bereitstellen soll, muss für den Warteschlangenmanager auf der sendenden Seite definiert sein. Die Kommunikationsverbindung muss definiert und verfügbar sein. Möglicherweise müssen Sie noch weitere IBM MQ-Objekte erstellen, z. B. Definitionen ferner Warteschlangen, Definitionen von Warteschlangenmanager-Aliasnamen und Empfangswarteschlangen für Antworten, um die unter „Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206 beschriebenen Szenarios zu implementieren.

Informationen zum Definieren von MQI-Kanälen finden Sie in „Definieren von MQI-Kanälen” auf Seite 31.

## Mehrere Nachrichtenkanäle pro Übertragungswarteschlange

Es ist möglich, mehr als einen Kanal pro Übertragungswarteschlange zu definieren, aber nur einer dieser Kanäle kann zu einem beliebigen Zeitpunkt aktiv sein. Berücksichtigen Sie diese Option bei der Bereitstellung alternativer Routen zwischen WS-Managern für die Fehlerberichtigung bei der Datenverkehrsverteilung und dem Verbindungsfehler. Eine Übertragungswarteschlange kann nicht von einem anderen Kanal verwendet werden, wenn der vorherige Kanal beendet wurde und einen Stapel von Nachrichten im unbestätbaren Fall an der sendenden Seite hinterlässt. Weitere Informationen finden Sie unter „Handhabung unbestätigter Kanäle” auf Seite 252.

## Kanal starten

Ein Kanal kann dazu veranlasst werden, Nachrichten auf eine von vier Arten zu übertragen. Es kann sein:

- Gestartet durch einen Operator (nicht Empfänger, Clusterempfänger oder Serververbindungskanäle).
- Ausgelöst aus der Übertragungswarteschlange. Diese Methode gilt nur für Senderkanäle und vollständig qualifizierte Serverkanäle (die Kanäle, die einen CONNAME angeben). Sie müssen die erforderlichen Objekte für die Auslösung von Kanälen vorbereiten.
- Gestartet von einem Anwendungsprogramm (nicht Empfänger, Clusterempfänger oder Serververbindungskanäle).
- Fernes Starten aus dem Netz durch einen Sender-, Cluster-Sender-, Requester-, Server-oder Client-Kanal-Kanal. Empfänger-, Cluster-Empfänger-und möglicherweise Server-und Requesterkanalübertragungen werden auf diese Weise gestartet; es handelt sich also um Serververbindungskanäle. Die Kanäle selbst müssen bereits gestartet sein (d. a. aktiviert).

**Anmerkung:** Da ein Kanal 'gestartet' ist, sendet er nicht unbedingt Nachrichten. Stattdessen kann es 'aktiviert' sein, die Übertragung zu starten, wenn eine der vier zuvor beschriebenen Ereignisse eintritt. Die Aktivierung und Deaktivierung eines Kanals wird mit den Bedienerbefehlen START und STOP erzielt.

### *Kanalstatus*

Ein Kanal kann zu einem beliebigen Zeitpunkt in einem von vielen Status sein. Einige Staaten haben auch Unterzustände. Aus einem bestimmten Zustand kann ein Kanal in andere Zustände übergehen.

Abbildung 20 auf Seite 242 zeigt die Hierarchie aller möglichen Kanalstatus und die Unterzustände, die für jeden Kanalstatus gelten.

Abbildung 21 auf Seite 243 zeigt die Links zwischen Kanalstatus. Diese Links gelten für alle Arten von Nachrichtenkanal-und Serververbindungskanälen.

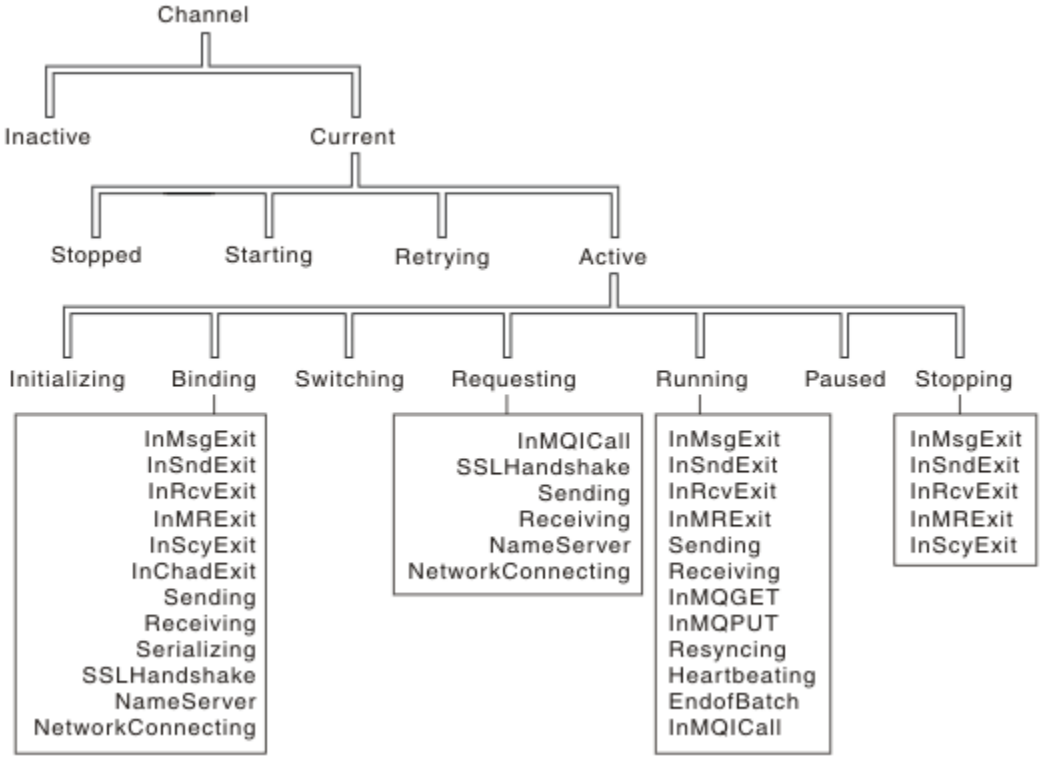



*Abbildung 20. Kanalstatus und Unterzustände*

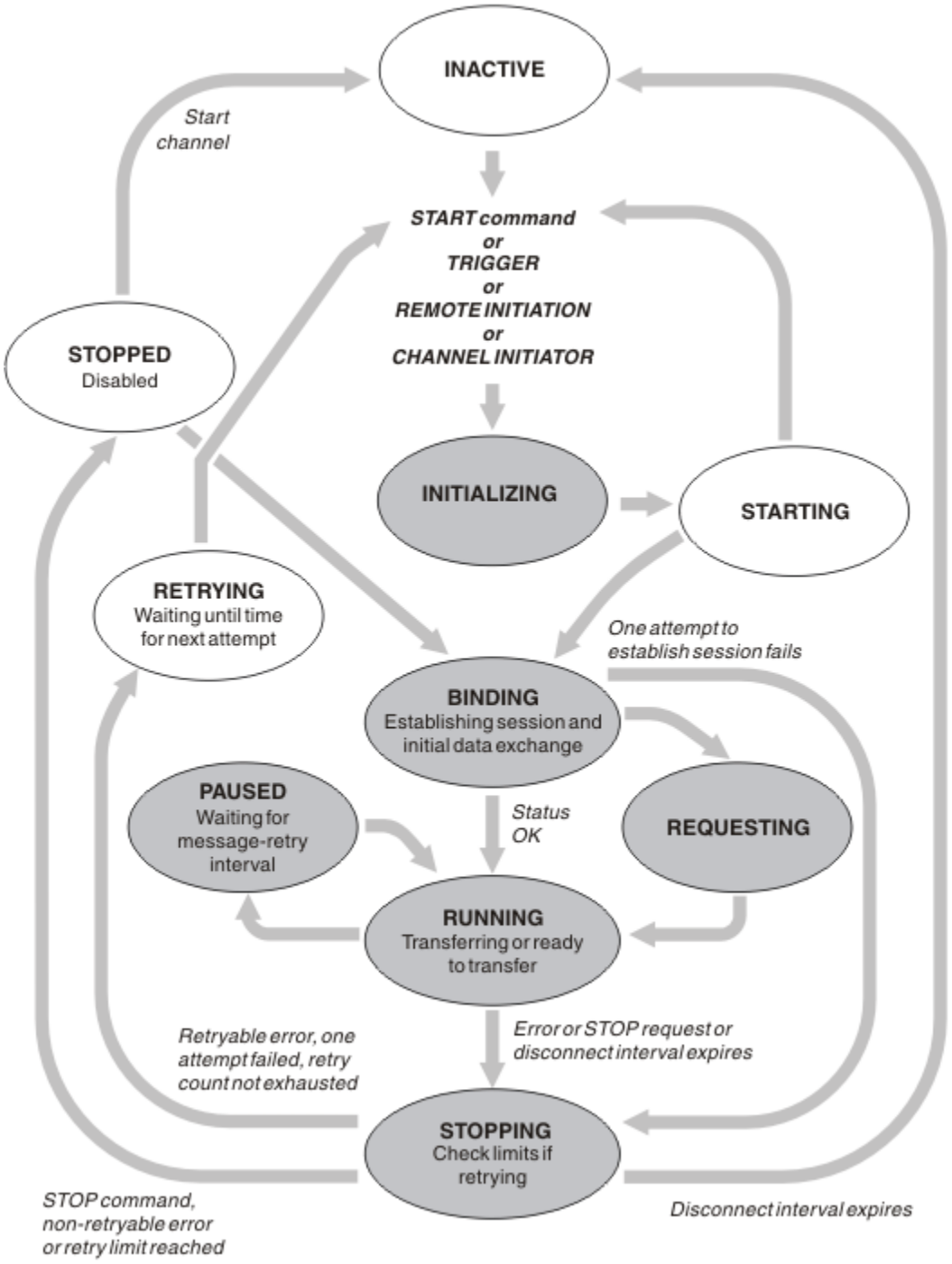



*Abbildung 21. Flows zwischen Kanalstatus*

## Aktuell und aktiv

Ein Kanal ist *aktuell* , wenn er sich in einem anderen Status als "inaktiv" befindet. Ein aktueller Kanal ist *aktiv* , es sei denn, er befindet sich im Status RETRYING, STOPPED oder STARTING. Wenn ein Kanal aktiv ist, verbraucht er Ressourcen, und ein Prozess oder Thread wird ausgeführt. Die sieben möglichen Status eines aktiven Kanals (INITIALIZING, BINDING, SWITCHING, REQUESTING, RUNNING, PAUSED oder STOPPING) werden in Abbildung 21 auf Seite 243 hervorgehoben.

Ein aktiver Kanal kann auch einen Substatus anzeigen, der mehr Details zu dem, was der Kanal gerade tut, gibt. Die Unterzustände für die einzelnen Status werden in Abbildung 20 auf Seite 242 angezeigt.

*Aktuell und aktiv*
Der Kanal ist " aktuell " , wenn er sich in einem anderen Status als 'inaktiv' befindet. Ein aktueller Kanal ist " aktiv " , es sei denn, er befindet sich im Status RETRYING, STOPPED oder STARTING.

Wenn ein Kanal "aktiv" ist, kann er auch einen Substatus anzeigen, der mehr Details zu dem, was der Kanal gerade tut, gibt.

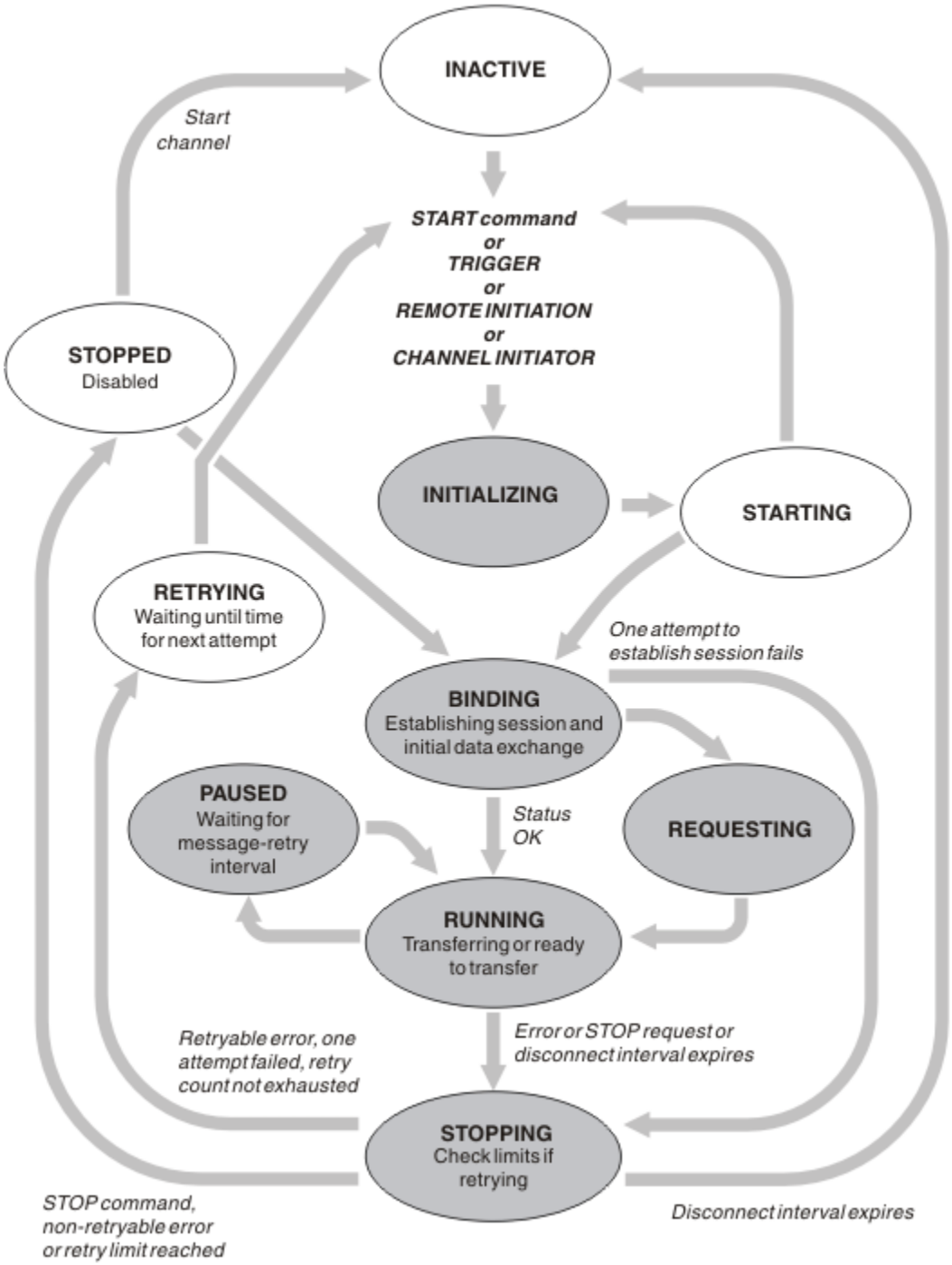



*Abbildung 22. Flows zwischen Kanalstatus*

**Anmerkung:**

1. Wenn sich ein Kanal in einem der sechs in Abbildung 22 auf Seite 244 hervorgehobenen Status befindet (INITIALIZING, BINDING, REQUESTING, RUNNING, PAUSED oder STOPPING), wird die Ressource verbraucht, und ein Prozess oder Thread wird ausgeführt; der Kanal ist *aktiv* .

2. Wenn sich ein Kanal im Status STOPPED befindet, kann die Sitzung aktiv sein, da der nächste Status noch nicht bekannt ist.

## Angeben der maximalen Anzahl aktueller Kanäle

Sie können die maximale Anzahl Kanäle angeben, die gleichzeitig ausgeführt werden können. Diese Zahl gibt die Anzahl der Kanäle an, die Einträge in der Kanalstatustabelle enthalten, einschließlich der Kanäle, die erneut versucht werden, und Kanäle, die gestoppt wurden. Geben Sie dies für Ihre Plattform an:

- z/OS Verwenden Sie den Befehl `ALTER QMGR MAXCHL` .
- IBM i Bearbeiten Sie die Initialisierungsdatei des Warteschlangenmanagers.
- Linux AIX Bearbeiten Sie die Konfigurationsdatei des Warteschlangenmanagers.
- Verwenden Sie die IBM MQ Explorer.

Weitere Informationen zu den Werten, die mit der Initialisierung oder der Konfigurationsdatei festgelegt wurden, finden Sie im Abschnitt Zeilengruppen für Konfigurationsdateien für verteilte Steuerung von Warteschlangen . Weitere Informationen zur Angabe der maximalen Anzahl an Kanälen finden Sie in den folgenden Abschnitten:

- ALW IBM MQ verwalten
- IBM i IBM MQ for IBM i verwalten
- z/OS IBM MQ for z/OS verwalten

**Anmerkung:**

1. Serververbindungskanäle sind in dieser Nummer enthalten.
2. Ein Kanal muss aktiv sein, bevor er aktiv werden kann. Wenn ein Kanal gestartet wird, aber nicht aktuell werden kann, schlägt der Start fehl.

## Angeben der maximalen Anzahl aktiver Kanäle

Sie können auch die maximale Anzahl aktiver Kanäle angeben, um zu verhindern, dass Ihr System von vielen Startkanälen überlastet wird. Wenn Sie diese Methode verwenden, legen Sie das Attribut "Unterbrechungsintervall" auf einen niedrigen Wert fest, damit die Wartezeitkanäle gestartet werden können, sobald andere Kanäle beendet werden.

Jedes Mal, wenn ein Kanal versucht, eine Verbindung zu seinem Partner herzustellen, muss er zu einem aktiven Kanal werden. Wenn der Versuch fehlschlägt, bleibt es ein aktueller Kanal, der nicht aktiv ist, bis es Zeit für den nächsten Versuch ist. Die Anzahl der Wiederholungen eines Kanalretries und die Häufigkeit, die durch den Wiederholungszähler und die Kanalattribute des Wiederholungsintervalls festgelegt wird. Es gibt kurze und lange Werte für diese beiden Attribute. Weitere Informationen finden Sie unter Kanalattribute .

Wenn ein Kanal zu einem aktiven Kanal werden muss (da ein Startbefehl abgesetzt wurde oder weil er ausgelöst wurde oder weil es Zeit für einen anderen Wiederholungsversuch ist), kann dies jedoch nicht möglich sein, da die Anzahl der aktiven Kanäle bereits den Maximalwert hat, wartet der Kanal, bis einer der aktiven Bereiche von einer anderen Kanalinstanz, die nicht mehr aktiv ist, freigegeben wird. Wenn jedoch ein Kanal gestartet wird, weil er über Remotezugriff eingeleitet wird und zu diesem Zeitpunkt keine aktiven Steckplätze zur Verfügung stehen, wird die ferne Initialisierung zurückgewiesen.

Wenn ein anderer Kanal als ein Requesterkanal versucht, aktiv zu werden, geht er in den Status STARTING. Dieser Status tritt auch dann auf, wenn ein aktiver Steckplatz sofort verfügbar ist, obwohl er nur im Status STARTING für einen kurzen Zeitraum vorhanden ist. Wenn der Kanal jedoch auf einen aktiven Steckplatz warten muss, befindet er sich im Status STARTING, während er wartet.

Requesterkanäle werden nicht in den Status STARTING (STARTING) gestartet. Wenn ein Requesterkanal nicht gestartet werden kann, weil die Anzahl der aktiven Kanäle bereits an der Grenze liegt, wird der Kanal abnormal beendet.

Wenn ein Kanal (bei dem es sich nicht um einen Requesterkanal handelt) nicht in der Lage ist, einen aktiven Zeitabschnitt zu erhalten, und daher auf einen Abschnitt wartet, wird eine Nachricht in das Protokoll z/OS oder in die z/OS-Konsole geschrieben und ein Ereignis wird generiert. Wenn ein Slot später freigegeben wird und der Kanal ihn anfordern kann, werden eine weitere Nachricht und ein weiteres Ereignis generiert. Keines dieser Ereignisse und Nachrichten wird generiert, wenn der Kanal in der Lage ist, einen Slot direkt zu erhalten.

Wenn ein Befehl STOP CHANNEL abgesetzt wird, während der Kanal darauf wartet, aktiv zu werden, wird der Kanal in den Status STOPPED (STOPPED) wechselt. Ein Ereignis Channel-Stopped wird ausgelöst.

Serververbindungskanäle sind in der maximalen Anzahl aktiver Kanäle enthalten.

Weitere Informationen zur Angabe der maximalen Anzahl aktiver Kanäle finden Sie in den folgenden Abschnitten:

- ALW IBM MQ verwalten
- IBM i IBM MQ for IBM i verwalten
- z/OS IBM MQ for z/OS verwalten

*Kanalfehler*
Fehler auf Kanälen bewirken, dass der Kanal weitere Übertragungen stoppenkann. Wenn es sich bei dem Kanal um einen Sender oder einen Server handelt, geht er in den Status WIEDERHOLUNG, da es möglich ist, dass sich das Problem möglicherweise selbst abbildet. Wenn der Kanal nicht in den Status RETRY wechseln kann, wechselt der Kanal in den Status STOPPED.

Für sendende Kanäle wird die zugeordnete Übertragungswarteschlange auf GET (DISABLED) gesetzt und die Auslösung ist ausgeschaltet. (Ein Befehl STOP mit STATUS (STOPPED) übernimmt die Seite, die sie in den Status STOPPED ausgegeben hat; nur das Auslaufen des Unterbrechungsintervalls oder ein STOP-Befehl mit STATUS (INACTIVE) macht sie normal beendet und wird inaktiv.) Kanäle, die sich im Status STOPPED befinden, benötigen Bedienereingriffe, bevor sie erneut gestartet werden können (siehe „Gestoppte Kanäle erneut starten” auf Seite 251 ).

**Anmerkung:** Für IBM i IBM i, AIX, Linux, and Windows-Systeme muss ein Kanalinitiator ausgeführt werden, damit die Wiederholung versucht werden kann. Wenn der Kanalinitiator nicht verfügbar ist, wird der Kanal inaktiv und muss manuell erneut gestartet werden. Wenn Sie ein Script zum Starten des Kanals verwenden, stellen Sie sicher, dass der Kanalinitiator aktiv ist, bevor Sie versuchen, das Script auszuführen.

Länger Wiederholungszähler (LONGRTY) beschreibt, wie die Wiederholung funktioniert. Wenn der Fehler gelöscht wird, wird der Kanal automatisch erneut gestartet, und die Übertragungswarteschlange wird erneut aktiviert. Wenn der Wiederholungsgrenzwert ohne die Fehlerbereinigung erreicht wird, wechselt der Kanal in den Status STOPPED (STOPPED). Ein gestoppte Kanal muss manuell durch den Bediener erneut gestartet werden. Ist der Fehler weiterhin vorhanden, wiederholt er die Operation nicht erneut. Wenn die Übertragung erfolgreich gestartet wird, wird die Übertragungswarteschlange erneut aktiviert.

z/OS Wenn der Kanalinitiator gestoppt wird, während sich ein Kanal im Status RETRYING oder STOPPED befindet, wird der Kanalstatus gespeichert, wenn der Kanalinitiator erneut gestartet wird. Der Kanalstatus für den Kanaltyp SVRCONN wird jedoch zurückgesetzt, wenn der Kanalinitiator gestoppt wird, während sich der Kanal im Status STOPPED befindet.

Multi Wenn der Warteschlangenmanager gestoppt wird, während sich ein Kanal im Status RETRYING oder STOPPED befindet, wird der Kanalstatus gespeichert, wenn der Warteschlangenmanager erneut gestartet wird. Ab IBM MQ 8.0 gilt dies auch für SVRCONN-Kanäle. Zuvor wurde der Kanalstatus für den Kanaltyp SVRCONN zurückgesetzt, wenn der Kanalinitiator gestoppt wurde, während sich der Kanal im Status STOPPED befand.

Wenn ein Kanal keine Nachricht in die Zielwarteschlange einlegen kann, weil die Warteschlange voll ist oder unterdrückt wird, kann der Kanal die Operation in einem Zeitintervall (das im Attribut für den Nachrichtenwiederholungsintervall angegeben ist) wiederholt versuchen, die Operation zu wiederholen

(die im Attribut für die Anzahl der Wiederholungen für Nachrichten angegeben ist). Alternativ können Sie einen eigenen Nachrichtenwiederholungsexit schreiben, der festlegt, welche Umstände eine Wiederholung bewirken, und die Anzahl der Versuche, die ausgeführt wurden. Der Kanal wechselt in den Status PAUSED, während er darauf wartet, dass das Nachrichtenwiederholungsintervall beendet wird.

Informationen zu den Kanalattributen finden Sie unter Kanalattribute . Informationen zum Exit für Nachrichtenwiederholungen finden Sie unter Kanalexitprogramme für Nachrichtenkanäle .

## *Grenzwerte für Serververbindungskanäle*

Sie können Grenzwerte für Serververbindungskanäle festlegen, um zu verhindern, dass Clientanwendungen die Kanalressourcen des Warteschlangenmanagers mit dem Parameter **MAXINST** ausschöpfen, und um zu verhindern, dass eine einzelne Clientanwendung die Kapazität des Serververbindungskanals mit dem Parameter **MAXINSTC** ausschöpft.

Sie legen **MAXINST** und **MAXINSTC** mit dem Befehl **DEFINE CHANNEL** fest.

Eine maximale Gesamtanzahl an Kanälen kann zu einem beliebigen Zeitpunkt auf einem einzelnen Warteschlangenmanager aktiv sein. Die Gesamtzahl der Serververbindungskanalinstanzen ist in der maximalen Anzahl aktiver Kanäle enthalten.

Wenn Sie nicht die maximale Anzahl simultaner Instanzen eines Serververbindungskanals angeben, der gestartet werden kann, ist es möglich, dass eine einzelne Clientanwendung eine Verbindung zu einem einzigen Serververbindungskanal herstellen kann, um die maximale Anzahl aktiver Kanäle zu erschöpfen, die verfügbar sind. Wenn die maximale Anzahl aktiver Kanäle erreicht ist, wird verhindert, dass andere Kanäle auf dem WS-Manager gestartet werden. Um diese Situation zu vermeiden, müssen Sie die Anzahl der simultanen Instanzen eines einzelnen Serververbindungskanals begrenzen, der unabhängig von dem Client gestartet werden kann, der gestartet wurde.

Wenn der Wert des Grenzwerts auf unter die aktuell laufende Anzahl der Instanzen des Serververbindungskanals reduziert wird, selbst auf null, sind die aktiven Kanäle nicht betroffen. Neue Instanzen können erst gestartet werden, wenn genügend vorhandene Instanzen nicht mehr ausgeführt werden, so dass die Anzahl der Instanzen, die momentan ausgeführt werden, kleiner als der Grenzwert ist.

Außerdem können viele verschiedene Client-Verbindungskanäle eine Verbindung zu einem einzelnen Serververbindungskanal herstellen. Der Grenzwert für die Anzahl simultaner Instanzen eines einzelnen Serververbindungskanals, der gestartet werden kann, unabhängig davon, welcher Client sie gestartet hat, verhindert, dass ein Client die maximale Kapazität des aktiven Kanals des Warteschlangenmanagers anstrengt. Wenn Sie nicht auch die Anzahl der simultanen Instanzen eines einzelnen Serververbindungskanals begrenzen, der von einem einzelnen Client aus gestartet werden kann, ist es möglich, dass eine einzelne fehlerhafte Clientanwendung so viele Verbindungen öffnet, dass sie die Kanalkapazität ausschöpft, die einem einzelnen Serververbindungskanal zugeordnet ist, und verhindert daher, dass andere Clients, die den Kanal verwenden müssen, keine Verbindung zu ihm herstellen müssen. Um diese Situation zu vermeiden, müssen Sie die Anzahl der simultanen Instanzen eines einzelnen Serververbindungskanals begrenzen, der von einem einzelnen Client aus gestartet werden kann.

Wenn der Wert der individuellen Clientgrenze unter die Anzahl der Instanzen des Serververbindungskanals reduziert wird, die momentan von einzelnen Clients ausgeführt werden, selbst auf null, sind die aktiven Kanäle nicht betroffen. Neue Instanzen des Serververbindungskanals können jedoch nicht von einem einzelnen Client gestartet werden, der den neuen Grenzwert überschreitet, bis genügend vorhandene Instanzen von diesem Client nicht mehr ausgeführt werden, so dass die Anzahl der Instanzen, die momentan ausgeführt werden, kleiner als der Wert dieses Parameters ist.

**Zugehörige Verweise**

Kanalattribute und Kanaltypen
CHANNEL DEFINE CHANNEL

## *Überprüfen, ob das andere Ende des Kanals noch verfügbar ist*

Sie können das Intervall der Überwachungssignale, das Intervall für die Nutzung des Überwachungszeitlimits und das Empfangszeitlimit verwenden, um zu überprüfen, ob das andere Ende des Kanals verfügbar ist.

## Überwachungssignale

Sie können das Attribut "heartbeat interval channel" verwenden, um anzugeben, dass Flüsse von dem sendenden Nachrichtenkanalsystem übergeben werden sollen, wenn keine Nachrichten in der Übertragungswarteschlange vorhanden sind, wie im Heartbeat-Intervall (HBINT) beschrieben.

## Keep Alive

z/OS Wenn Sie unter z/OSTCP/IP als Transportprotokoll verwenden, können Sie auch einen Wert für das Attribut **Keepalive** des Intervallkanals (**KAINT**) angeben. Es wird empfohlen, für das **Keepalive** -Intervall einen höheren Wert als das Heartbeatintervall und einen kleineren Wert als den Unterbrechungswert anzugeben. Sie können dieses Attribut verwenden, um einen Zeitüberschreitungswert für jeden Kanal anzugeben, wie in Keepalive Interval (KAINT) beschrieben.

Multi Auf IBM i-, AIX, Linux, and Windows -Systemen können Sie `keepalive=yes`festlegen, wenn Sie TCP als Transportprotokoll verwenden. Wenn Sie diese Option angeben, prüft TCP regelmäßig, ob das andere Ende der Verbindung noch verfügbar ist. Ist dies nicht der Kanal, wird der Kanal beendet. Diese Option wird in Keepalive Interval (KAINT) beschrieben.

Wenn Sie über unzuverlässige Kanäle verfügen, die TCP-Fehler melden, verwenden Sie die Option **Keepalive** , um die Wahrscheinlichkeit zu beheben, dass die Kanäle wiederhergestellt werden.

Sie können Zeitintervalle angeben, um das Verhalten der Option **Keepalive** zu steuern. Wenn Sie das Zeitintervall ändern, werden nur die TCP/IP-Kanäle gestartet, die nach der Änderung gestartet wurden. Stellen Sie sicher, dass der Wert, den Sie für das Zeitintervall auswählen, kleiner ist als der Wert des Unterbrechungsintervalls für den Kanal.

Weitere Informationen zur Verwendung der Option **Keepalive** finden Sie unter dem Parameter **KAINT** im Befehl **DEFINE CHANNEL** .

## Zeitlimit für Empfang

Wenn Sie TCP als Transportprotokoll verwenden, wird auch das empfangende Ende einer inaktiven Nicht-MQI-Kanalverbindung geschlossen, wenn für einen Zeitraum keine Daten empfangen werden. Dieser Zeitraum, der Wert für *Empfangszeitlimit* , wird entsprechend dem Wert für **HBINT** (Überwachungssignalintervall) bestimmt.

In IBM MQ für IBM i, AIX, Linux, and Windows-Systeme, wird der Wert *Zeitlimitüberschreitung bei Empfang* wie folgt eingestellt:

1. Bei einer anfänglichen Anzahl von Datenflüssen ist der *Empfangszeitlimitwert* doppelt so hoch wie der Wert **HBINT** aus der Kanaldefinition, bevor eine Vereinbarung stattfindet.
2. Nachdem die Kanäle einen Wert für **HBINT** vereinbart haben, wird der Wert für *receive timeout* auf den doppelten Wert gesetzt, wenn **HBINT** auf weniger als 60 Sekunden gesetzt ist. Wenn **HBINT** auf 60 Sekunden oder mehr gesetzt ist, wird der Wert für *Empfangszeitlimit* auf 60 Sekunden größer als der Wert von HBINT gesetzt.

z/OS Unter z/OSwird der Wert für *Empfangszeitlimit* wie folgt festgelegt:

1. Bei einer anfänglichen Anzahl von Datenflüssen ist der *Empfangszeitlimitwert* doppelt so hoch wie der Wert **HBINT** aus der Kanaldefinition, bevor eine Vereinbarung stattfindet.
2. Wenn **RCVTIME** festgelegt ist, wird das Zeitlimit abhängig vom Parameter **RCVTTYPE** auf einen der folgenden Werte gesetzt und unterliegt einem von **RCVTMIN** festgelegten Grenzwert, sofern dieser gilt:
   - das ausgehandelte HBINT multipliziert mit einer Konstanten
   - das vereinbarte HBINT plus eine konstante Anzahl von Sekunden
   - eine konstante Anzahl von Sekunden

   **RCVTMIN** gilt nicht, wenn **RCVTTYPE(EQUAL)** konfiguriert ist. Wenn Sie den konstanten Wert **RCVTIME** verwenden und ein Überwachungssignalintervall verwenden, geben Sie keinen Wert für **RCVTIME** an,

der kleiner als das Überwachungssignalintervall ist. Details zu den Attributen **RCVTIME**, **RCVTMIN** und **RCVTTYPE** finden Sie in der Beschreibung des Befehls **ALTER QMGR** .

**Anmerkung:**

1. Wenn einer der Werte null ist, gibt es kein Zeitlimit.
2. Für Verbindungen, die keine Überwachungssignale unterstützen, wird der Wert **HBINT** in Schritt 2 auf null festgelegt. Daher gibt es kein Zeitlimit, sodass Sie TCP/IP KEEPALIVEverwenden müssen.
3. Für Clientverbindungen, die gemeinsame Dialoge verwenden, können Heartbeats über den Kanal (von beiden Enden) die gesamte Zeit fließen, nicht nur, wenn ein MQGET-Aufruf aussteht.
4. Bei Clientverbindungen, bei denen die gemeinsamen Dialoge nicht verwendet werden, werden Überwachungssignale vom Server nur dann aus dem Server fließen, wenn der Client einen MQGET-Aufruf mit einem Wartestatus ausgibt. Daher wird es nicht empfohlen, das Intervall der Überwachungssignale für Clientkanäle zu klein zu setzen. Wenn das Überwachungssignal beispielsweise auf 10 Sekunden gesetzt wird, schlägt ein MQCMIT-Aufruf fehl (mit MQRC_CONNECTION_BROKEN), wenn die Festschreibung länger als 20 Sekunden dauert, da während dieser Zeit keine Daten geflossen sind. Dies kann bei großen Arbeitseinheiten der Fall sein. Es tritt jedoch nicht auf, wenn geeignete Werte für das Intervall der Überwachungssignale ausgewählt werden, da nur MQGET mit Wartezeit einen signifikanten Zeitraum in Anspruch nimmt.

   Wenn **SHARECNV** ungleich null ist, verwendet der Client eine Vollduplexverbindung, d. h., der Client kann während aller MQI-Aufrufe Überwachungssignale (und führt sie aus)
5. Das Abbrechen der Verbindung nach dem doppelten Intervall der Überwachungssignale ist gültig, da mindestens in jedem Intervall der Überwachungssignale ein Daten-oder Überwachungssignalfluss erwartet wird. Das Festlegen des Überwachungssignalintervalls ist zu klein, kann jedoch zu Problemen führen, insbesondere wenn Sie Kanalexits verwenden. Wenn der Wert für **HBINT** beispielsweise eine Sekunde beträgt und ein Sende-oder Empfangsexit verwendet wird, wartet die Empfangsseite nur 2 Sekunden, bevor der Kanal abgebrochen wird. Wenn der MCA eine Task wie das Verschlüsseln der Nachricht ausführt, ist dieser Wert möglicherweise zu kurz.

## Empfohlene Einstellungen

### z/OS IBM MQ for z/OS

Als Startpunkt können Sie Folgendes verwenden:

```
/cpf ALTER QMGR TCPKEEP(YES) RCVTTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

Dabei steht cpf für das Befehlspräfix für das Subsystem des Warteschlangenmanagers.

Weitere Informationen zu den verschiedenen Parametern finden Sie unter **ALTER QMGR** und IBM MQ Netzverfügbarkeit .

Wenn die IP-Adresse des Senders in mehrere Adressen übersetzt werden kann, müssen Sie möglicherweise **ADOPTCHK** auf QMName anstelle von ALLsetzen.

### Multi IBM MQ for Multiplatforms

Fügen Sie in qm.ini die folgenden Informationen hinzu:

```
TCP:
KeepAlive=Yes
CHANNELS:
AdoptNewMCA=ALL
AdoptNewMCACheck=ALL
```

Weitere Informationen finden Sie unter **ALTER QMGR**, Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangenund „Zeilengruppe 'Channels' in der Datei 'qm.ini'" auf Seite 125 .

Wenn die IP-Adresse des Absenders in mehrere Adressen umgesetzt werden kann, müssen Sie **AdoptNewMCACheck** möglicherweise auf QMNAME anstatt auf **ALL**setzen.

## *MCA-Adopting*

Die Funktion "Adopt MCA" ermöglicht es IBM MQ, einen Empfängerkanal abzubrechen und an seiner Stelle einen neuen zu starten.

Wenn ein Kanal keinen Kontakt mehr hat, kann der Empfängerkanal in einem Zustand 'Kommunikation empfangen' zurückgelassen werden. Wenn die Kommunikation erneut aufgebaut wird, versucht der Senderkanal, die Verbindung wieder herzustellen. Wenn der ferne WS-Manager feststellt, dass der Empfängerkanal bereits ausgeführt wird, lässt er nicht zu, dass eine andere Version desselben Empfängerkanals gestartet wird. Für dieses Problem ist ein Benutzereingriff erforderlich, um das Problem oder die Verwendung des Systemkeepalive zu beheben.

Mit der Funktion "Adopt MCA" wird das Problem automatisch gelöst. Sie ermöglicht es IBM MQ, einen Empfängerkanal abzubrechen und an seiner Stelle einen neuen zu starten.

**Zugehörige Tasks**

IBM MQ verwalten

IBM MQ for z/OS verwalten

IBM MQ for IBM i verwalten

## *Kanäle stoppen und in den Quiescemodus versetzt*

Sie können einen Kanal stoppen und in den Quiescemodus versetzt, bevor das Unterbrechungszeitintervall abläuft.

Nachrichtenkanäle sind so konzipiert, dass sie lange Verbindungen zwischen Warteschlangenmanagern mit ordnungsgemässegesteuerter Beendigung, die nur durch das Kanalattribut "Unterbrechungsintervall" gesteuert werden, bestehen Dieser Mechanismus funktioniert gut, es sei denn, der Bediener muss den Kanal beenden, bevor das Unterbrechungszeitintervall abläuft. Dieser Bedarf kann in den folgenden Situationen auftreten:

- Systemstilllegung
- Ressourcenschonung
- Unilaterale Aktion an einem Ende eines Kanals

In diesem Fall können Sie den Kanal stoppen. Dazu können Sie folgende Optionen verwenden:

- MQSC-Befehl STOP CHANNEL
- Befehl "Stop Channel PCF"
- den IBM MQ Explorer
- z/OS IBM i andere plattformspezifische Mechanismen, wie folgt:

  z/OS **Für z/OS:**
  Die Anzeige 'Kanal stoppen'

  IBM i **Für IBM i:**
  Der CL-Befehl ENDMQMCHL oder die Option END in der Anzeige WRKMQMCHL

Es gibt drei Optionen zum Stoppen von Kanälen mit den folgenden Befehlen:

**QUIESCE**
Die Option QUIESCE versucht, den aktuellen Stapel von Nachrichten zu beenden, bevor der Kanal gestoppt wird.

**FORCE**
Die Option FORCE versucht, den Kanal sofort zu stoppen, und möglicherweise muss der Kanal beim Neustart erneut synchronisiert werden, da der Kanal möglicherweise unbestätigt bleibt.

z/OS Unter IBM MQ for z/OS unterbricht FORCE jegliche laufende Nachrichtenneuzuordnung, wodurch möglicherweise BIND_NOT_FIXED-Nachrichten teilweise neu zugeteilt werden oder sich nicht mehr in der vorgegebenen Reihenfolge befinden.

**TERMINATE**
Die Option TERMINATE versucht, den Kanal sofort zu stoppen, und beendet den Thread oder den Prozess des Kanals.

z/OS Unter IBM MQ for z/OS unterbricht TERMINATE jegliche laufende Nachrichtenneuzuordnung, wodurch möglicherweise BIND_NOT_FIXED-Nachrichten teilweise neu zugeteilt werden oder sich nicht mehr in der vorgegebenen Reihenfolge befinden.

Alle diese Optionen verlassen den Kanal in einem STOPPED-Status, der einen Bedienereingriff erfordert, um ihn erneut zu starten.

Das Stoppen des Kanals auf der Senderseite ist zwar wirksam, erfordert jedoch einen Bedienereingriff, um den Neustart zu starten. Am empfangenden Ende des Kanals sind die Dinge sehr viel schwieriger, da der MCA auf Daten von der sendenden Seite wartet und es keine Möglichkeit gibt, einen *ordnungsgemäßen* Abschluss des Kanals von der empfangenden Seite zu initiieren; der Stoppbefehl steht an, bis der MCA aus seinem Wartestatus für Daten zurückkehrt.

Abhängig von den erforderlichen Betriebsmerkmalen gibt es daher drei empfohlene Methoden für die Verwendung von Kanälen:

- Wenn die Kanäle lange ausgeführt werden sollen, beachten Sie, dass die Beendigung nur von der sendenden Seite aus ordnungsgemäß beendet werden kann. Wenn Kanäle unterbrochen werden, d. a. gestoppt sind, ist ein Bedienereingriff (ein Befehl START CHANNEL) erforderlich, um sie erneut starten zu können.
- Wenn die Kanäle nur dann aktiv sein sollen, wenn Nachrichten für die Übertragung vorhanden sind, setzen Sie das Unterbrechungsintervall auf einen relativ niedrigen Wert. Die Standardeinstellung ist hoch und wird daher nicht für Kanäle empfohlen, in denen diese Steuerungsstufe erforderlich ist. Da es schwierig ist, den empfangenden Kanal zu unterbrechen, ist es die wirtschaftlichste Option, den Kanal automatisch zu trennen und die Verbindung zu den Workloadanforderungen wieder herzustellen. Für die meisten Kanäle kann die entsprechende Einstellung des Unterbrechungsintervalls heuristisch festgelegt werden.
- Sie können das Attribut "heartbeat-interval" verwenden, um zu bewirken, dass der sendende MCA einen Überwachungssignalfluss an den empfangenden MCA sendet, wenn er keine Nachrichten zum Senden hat. Diese Aktion gibt den empfangenden MCA aus seinem Wartestatus frei und gibt ihm die Möglichkeit, den Kanal in den Quiescemodus zu setzen, ohne zu warten, bis das Unterbrechungsintervall abgelaufen ist. Geben Sie dem Überwachungssignalintervall einen niedrigeren Wert als den Wert des Unterbrechungsintervalls an.

  **Anmerkung:**

  1. Es wird empfohlen, das Unterbrechungsintervall für Serverkanäle auf einen niedrigen Wert oder auf Überwachungssignale zu setzen. Dieser niedrige Wert soll den Fall zulassen, dass der Requesterkanal abnormal beendet wird (z. B. weil der Kanal abgebrochen wurde), wenn keine Nachrichten für den zu sengenden Serverkanal vorhanden sind. Wenn das Unterbrechungsintervall hoch ist und keine Überwachungssignale verwendet werden, erkennt der Server nicht, dass der Anforderer beendet wurde (was er nur beim nächsten Versuch, eine Nachricht an den Requester zu senden), beendet hat. Während der Server noch aktiv ist, hält er die Übertragungswarteschlange für exklusive Eingabe bereit, um weitere Nachrichten zu erhalten, die in die Warteschlange eintreffen. Wenn versucht wird, den Kanal vom anfordernden Benutzer erneut zu starten, empfängt die Startanforderung einen Fehler, da der Server immer noch die Übertragungswarteschlange für die exklusive Eingabe geöffnet hat. Es ist erforderlich, den Serverkanal zu stoppen und den Kanal anschließend erneut vom Requester erneut zu starten.

## *Gestoppte Kanäle erneut starten*

Wenn ein Kanal in den Status STOPPED wechselt, müssen Sie den Kanal manuell erneut starten.

## Informationen zu diesem Vorgang

Für Sender-oder Serverkanäle wurde die zugeordnete Übertragungswarteschlange auf GET (DISABLED) gesetzt und die Auslösung wurde inaktiviert, wenn der Kanal in den Status STOPPED (STOPPED) eingetreten ist. Wenn die Startanforderung empfangen wird, werden diese Attribute automatisch zurückgesetzt.

z/OS Wenn der Kanalinitiator gestoppt wird, während sich ein Kanal im Status RETRYING oder STOPPED befindet, wird der Kanalstatus gespeichert, wenn der Kanalinitiator erneut gestartet wird. Der Kanalstatus für den Kanaltyp SVRCONN wird jedoch zurückgesetzt, wenn der Kanalinitiator gestoppt wird, während sich der Kanal im Status STOPPED befindet.

Multi Wenn der Warteschlangenmanager gestoppt wird, während sich ein Kanal im Status RETRYING oder STOPPED befindet, wird der Kanalstatus gespeichert, wenn der Warteschlangenmanager erneut gestartet wird. Ab IBM MQ 8.0 gilt dies auch für SVRCONN-Kanäle. Zuvor wurde der Kanalstatus für den Kanaltyp SVRCONN zurückgesetzt, wenn der Kanalinitiator gestoppt wurde, während sich der Kanal im Status STOPPED befand.

## Prozedur

- Starten Sie den Kanal auf eine der folgenden Arten erneut:
  - Durch Verwendung des Befehls START CHANNEL MQSC .
  - Durch Verwendung des Befehls PCF für Kanalstart .
  - Mit dem IBM MQ Explorer
  - z/OS Unter z/OS über die Anzeige Kanal starten.
  - IBM i Unter IBM i entweder über die Befehlszeile mit dem Befehl STRMQMCHL oder mit der Option START in der Anzeige WRKMQMCHL.

### *Handhabung unbestätigter Kanäle*

Ein unbestäsender Kanal ist ein Kanal, der sich im Zweifel mit einem fernen Kanal befindet, über den Nachrichten gesendet und empfangen wurden.

## Informationen zu diesem Vorgang

Beachten Sie die Unterscheidung zwischen diesem und einem Warteschlangenmanager, der Zweifel daran hat, welche Nachrichten in einer Warteschlange festgeschrieben werden sollen.

Sie können die Möglichkeit verringern, dass ein Kanal unbestätigt wird, indem Sie den Kanalparameter 'Batch Heartbeat' (**BATCHHB**) verwenden. Wenn ein Wert für diesen Parameter angegeben wird, überprüft ein Senderkanal, ob der ferne Kanal noch aktiv ist, bevor weitere Aktionen ausgeführt werden. Wird keine Antwort empfangen, wird davon ausgegangen, dass der Empfängerkanal nicht mehr aktiv ist. Die Nachrichten können rückgängig gemacht und erneut weitergeleitet werden, und der Sender-Channel wird nicht in Frage gestellt. Dadurch wird die Zeit verringert, in der der Kanal in Zweifel zum Zeitraum zwischen dem Senderkanal gestellt werden konnte, der verifiziert, dass der Empfängerkanal noch aktiv ist, und zu überprüfen, ob der Empfängerkanal die gesendeten Nachrichten empfangen hat. Weitere Informationen zum Parameter für den Stapelüberwachungssignalwert finden Sie unter Kanalattribute .

Unbestätigte Kanalprobleme werden in der Regel automatisch aufgelöst. Auch wenn die Kommunikation verloren geht und ein Kanal im Zweifelsfall mit einem Nachrichtenstatsch im Sender in Zweifel steht, wenn der Empfangsstatus unbekannt ist, wird die Situation behoben, wenn die Kommunikation erneut aufgebaut wird. Die Folgenummer und die LUWID-Datensätze werden zu diesem Zweck aufbewahrt. Der Kanal steht im Zweifel, bis LUWID-Informationen ausgetauscht wurden und nur ein Nachrichtenstelfenstapel für den Kanal unbestätigt sein kann.

Sie können den Kanal bei Bedarf manuell resynchronisieren. Der Begriff "manuell" schließt die Verwendung von Operatoren oder Programmen ein, die IBM MQ -Systemverwaltungsbefehle enthalten. Der manuelle Resynchronisationsprozess funktioniert wie folgt. Diese Beschreibung verwendet MQSC-Befehle, aber Sie können auch die PCF-Entsprechungen verwenden.

## Vorgehensweise

1. Verwenden Sie den Befehl **DISPLAY CHSTATUS** , um die ID der zuletzt festgeschriebenen logischen Arbeitseinheit (LUWID) für jede Seite des Kanals zu suchen.

   Verwenden Sie dazu die folgenden Befehle:

   - Für die unbestätigte Seite des Kanals:

     ```
     DISPLAY CHSTATUS(name) SAVED CURLUWID
     ```

     Mit den Parametern **CONNAME** und **XMITQ** können Sie den Kanal genauer angeben.
   - Für die Empfängerseite des Kanals:

     ```
     DISPLAY CHSTATUS( name ) SAVED LSTLUWID
     ```

     Sie können den Parameter **CONNAME** verwenden, um den Kanal genauer zu identifizieren.

   **Anmerkung:** Die Befehle sind unterschiedlich, da nur die sendende Seite des Kanals unbestätigt sein kann. Die empfangende Seite ist nie im Zweifel.

   IBM i Unter IBM ikann der Befehl **DISPLAY CHSTATUS** aus einer Datei mit dem Befehl **STRMQMMQSC** oder dem CL-Befehl "Mit MQM-Kanalstatus arbeiten" **WRKMQMCHST**ausgeführt werden.
2. Wenn die beiden LUWIDs identisch sind, verwenden Sie den Befehl **RESOLVE CHANNEL** , um die unbestätigten Nachrichten festzuschreiben.

   Wenn die beiden LUWIDs identisch sind, hat die empfangende Seite die Arbeitseinheit festgeschrieben, die vom Absender als unbestätigt betrachtet wird. Die sendende Seite kann nun die unbestätigen Nachrichten aus der Übertragungswarteschlange entfernen und sie erneut aktivieren. Dazu wird der folgende **RESOLVE CHANNEL** -Befehl verwendet:

   ```
   RESOLVE CHANNEL(name) ACTION(COMMIT)
   ```
3. Wenn sich die beiden LUWIDs unterscheiden, verwenden Sie den Befehl **RESOLVE CHANNEL** , um die unbestätigten Nachrichten zurückzusetzen.

   Wenn die beiden LUWIDs unterschiedlich sind, hat die empfangende Seite die UOW nicht festgeschrieben, die der Absender als unbestätigt betrachtet. Die sendende Seite muss die unbestätigen Nachrichten in der Übertragungswarteschlange aufbewahren und sie erneut senden. Dazu wird der folgende **RESOLVE CHANNEL** -Befehl verwendet:

   ```
   RESOLVE CHANNEL( name ) ACTION(BACKOUT)
   ```

   IBM i Unter IBM ikönnen Sie den Befehl "Resolve MQM Channel" verwenden: **RSVMQMCHL**.

## Ergebnisse

Wenn dieser Prozess abgeschlossen ist, ist der Kanal nicht mehr im Zweifel. Die Übertragungswarteschlange kann jetzt, falls erforderlich, von einem anderen Kanal verwendet werden.

**Zugehörige Verweise**

DISPLAY CHSTATUS (Kanalstatus anzeigen)

RESOLVE CHANNEL (einen Kanal bitten, unbestätigte Nachrichten zu aufzulösen)

# Sicherheit von Nachrichten

Zusätzlich zu den typischen Wiederherstellungsfunktionen von IBM MQ stellt das verteilte Warteschlangenmanagement sicher, dass Nachrichten ordnungsgemäß zugestellt werden, indem eine Synchronisationspunktprozedur verwendet wird, die zwischen den beiden Enden des Nachrichtenkanals koordiniert wird. Wenn diese Prozedur einen Fehler feststellt, wird der Kanal geschlossen, so dass Sie das Problem

untersuchen und die Nachrichten sicher in der Übertragungswarteschlange behalten können, bis der Kanal erneut gestartet wird.

Die Synchronisationspunktprozedur hat einen zusätzlichen Vorteil, indem sie versucht, eine *im Zweifel*-Situation wiederherzustellen, wenn der Kanal gestartet wird. ( *Im Zweifel* ist der Status einer Arbeitseinheit mit Wiederherstellung, für die ein Synchronisationspunkt angefordert wurde, aber das Ergebnis der Anforderung ist noch nicht bekannt.) Darüber hinaus sind diese Funktion die beiden folgenden Funktionen:

1. Mit Commit oder Backout auflösen
2. Folgenummer zurücksetzen

Die Verwendung dieser Funktionen findet nur in Ausnahmefällen statt, da sich der Kanal in den meisten Fällen automatisch wiederfindet.

### Schnelle, nicht persistente Nachrichten

Mit dem Kanalattribut der nicht persistenten Nachrichtengeschwindigkeit (NPMSPEED) kann angegeben werden, dass alle nicht persistenten Nachrichten auf dem Kanal schneller zugestellt werden sollen. Weitere Informationen zu diesem Attribut finden Sie im Abschnitt Nicht persistente Nachrichtengeschwindigkeit (NPMSPEED) .

Wenn ein Kanal beendet wird, während sich schnell, nicht persistente Nachrichten im Transit befinden, gehen die Nachrichten möglicherweise verloren, und die Anwendung kann bei Bedarf für die Wiederherstellung sorgen.

Wenn der empfangende Kanal die Nachricht nicht in die Zielwarteschlange einlegen kann, wird sie in die Warteschlange für nicht zustellbare Nachrichten gestellt, wenn eine Warteschlange definiert wurde. Ist dies nicht der Fall, wird die Nachricht gelöscht.

**Anmerkung:** Wenn das andere Ende des Kanals die Option nicht unterstützt, wird der Kanal mit normaler Geschwindigkeit ausgeführt.

### Nicht zugegebene Nachrichten

Informationen darüber, was passiert, wenn eine Nachricht nicht zugestellt werden kann, finden Sie in „Was passiert, wenn eine Nachricht nicht zugestellt werden kann?" auf Seite 254.

## Was passiert, wenn eine Nachricht nicht zugestellt werden kann?

Wenn eine Nachricht nicht zugestellt werden kann, kann der MCA sie auf mehrere Arten verarbeiten. Sie kann es erneut versuchen, sie kann an den Absender zurückkehren oder sie in die Warteschlange für dead-Mail setzen.

Abbildung 23 auf Seite 255 zeigt die Verarbeitung an, die auftritt, wenn ein MCA keine Nachricht in die Zielwarteschlange einlegen kann. (Die angezeigten Optionen gelten nicht für alle Plattformen.)

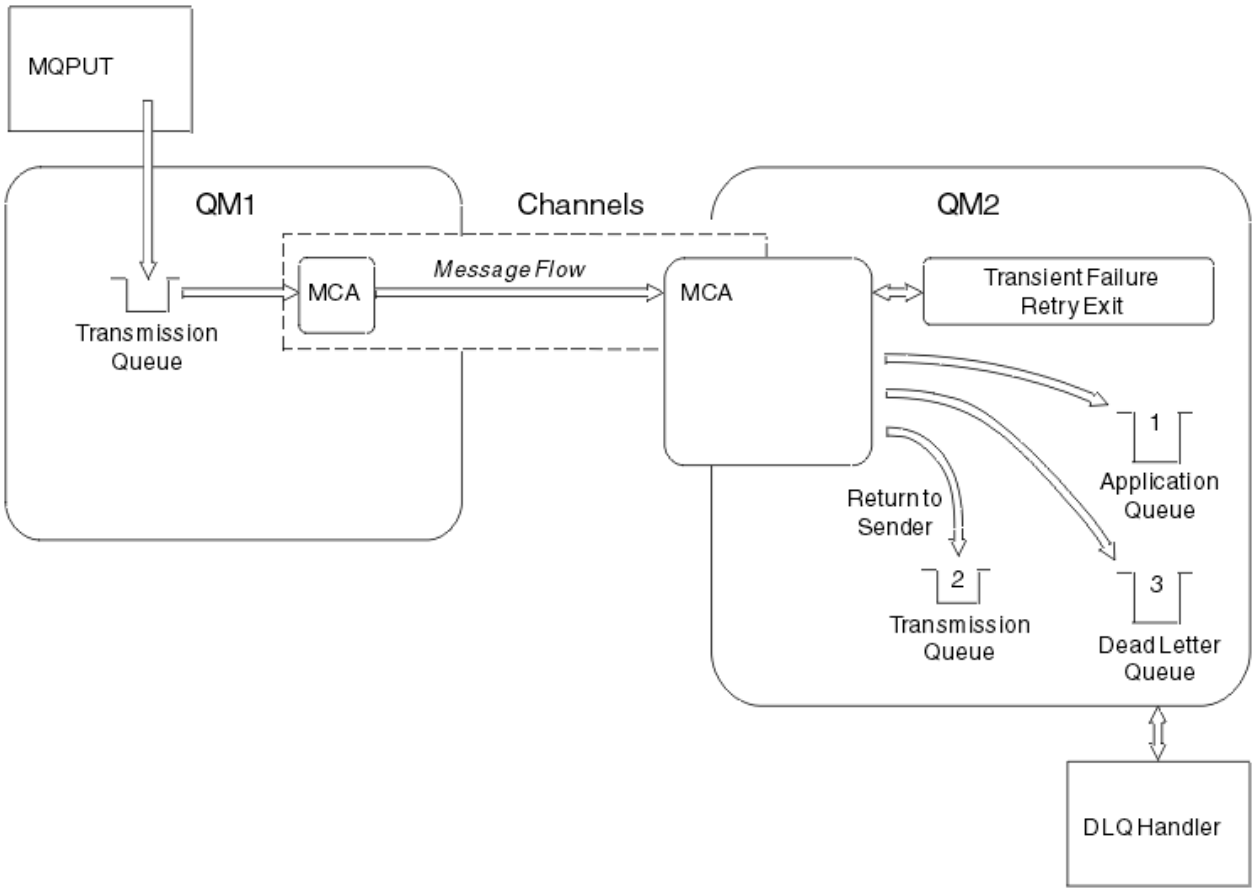



*Abbildung 23. Was passiert, wenn eine Nachricht nicht zugestellt werden kann*

Wie in der Abbildung dargestellt, kann der MCA mehrere Dinge mit einer Nachricht tun, die er nicht liefern kann. Die Aktion wird durch die Optionen bestimmt, die bei der Definition des Kanals und in den MQPUT-Berichtsoptionen für die Nachricht angegeben sind.

1. Nachrichtenwiederholung

   Wenn der Nachrichtenkanalnachrichtenkanalcode (MCA) keine Nachricht aus einem Grund in die Zielwarteschlange einlegen kann (z. B. weil die Warteschlange voll ist), kann der MCA warten und die Operation zu einem späteren späteren Vorgang wiederholen. Sie können feststellen, ob der MCA wartet, wie lange und wie viele Versuche er hat.

   - Sie können eine Nachrichtenwiederholungszeit und ein Intervall für MQPUT-Fehler angeben, wenn Sie Ihren Kanal definieren. Wenn die Nachricht nicht in die Zielwarteschlange gestellt werden kann, weil die Warteschlange voll ist oder für die Ausführung gesperrt ist, versucht der MCA die Operation, wie oft angegeben, in dem angegebenen Zeitintervall versucht wird.
   - Sie können einen eigenen Nachrichtenwiederholungsexit schreiben. Über den Exit können Sie angeben, unter welchen Bedingungen der MCA die MQPUT-oder MQOPEN-Operation wiederholen soll. Geben Sie den Namen des Exits an, wenn Sie den Kanal definieren.

2. Zurück-zu-Absender

   Wenn die Nachrichtenwiederholung nicht erfolgreich war oder ein anderer Typ von Fehler festgestellt wurde, kann der Nachrichtenkanalnachrichtenkanalfehler die Nachricht an den Absender zurücksenden. Zum Aktivieren von "return-to-sender" müssen Sie die folgenden Optionen im Nachrichtendeskriptor angeben, wenn Sie die Nachricht in die ursprüngliche Warteschlange stellen:

   - Die Berichtsoption MQRO_EXCEPTION_WITH_FULL_DATA
   - Die Berichtsoption MQRO_DISCARD_MSG

- Der Name der Empfangswarteschlange für Antworten und der Empfangswarteschlange für Antworten an den Warteschlangenmanager.

Wenn der MCA die Nachricht nicht in die Zielwarteschlange einlegen kann, generiert er einen Ausnahmebericht, der die ursprüngliche Nachricht enthält, und stellt ihn in eine Übertragungswarteschlange, die an die in der ursprünglichen Nachricht angegebene Warteschlange für Antwortnachrichten gesendet werden soll. (Wenn sich die Warteschlange für Antwortnachrichten auf demselben WS-Manager wie der Nachrichtenkanalmanager befindet, wird die Nachricht direkt in diese Warteschlange gestellt, nicht in eine Übertragungswarteschlange.)

3. Warteschlange für nicht zustellbare Nachrichten

   Wenn eine Nachricht nicht zugestellt oder zurückgegeben werden kann, wird sie in die Warteschlange für dead-letter (DLQ) gestellt. Sie können den DLQ-Handler verwenden, um die Nachricht zu verarbeiten. Diese Verarbeitung wird im Abschnitt Nachrichten in einer Warteschlange für nicht zustellbare Nachrichten verarbeiten für IBM MQ for UNIX-, Linux- und Windows-Systeme und in Steuerroutine der Warteschlange für nicht zustellbare Nachrichten (CSQUDLQH) für z/OS-Systeme beschrieben. Wenn die Warteschlange für nicht zustellbare Nachrichten nicht verfügbar ist, wird die Nachricht vom sendenden Nachrichtenkanalsystem (MCA) in der Übertragungswarteschlange und der Kanal gestoppt. Bei einem schnellen Kanal gehen nicht persistente Nachrichten verloren, die nicht in eine Warteschlange mit dead-letter geschrieben werden können.

   Wenn in IBM WebSphere MQ 7.0 keine lokale Warteschlange für nicht zustellbare Nachrichten definiert ist, die ferne Warteschlange nicht erreichbar oder nicht definiert ist und keine ferne Warteschlange für nicht zustellbare Nachrichten vorhanden ist, schaltet der Sendekanal in den RETRY-Modus um und Nachrichten werden automatisch in die Übertragungswarteschlange zurückgesetzt.

**Zugehörige Verweise**

Warteschlange für nicht zustellbare Mail verwenden (USEDLQ)

## Ausgelöste Kanäle

IBM MQ stellt eine Funktion bereit, um eine Anwendung automatisch zu starten, wenn bestimmte Bedingungen in einer Warteschlange erfüllt sind. Diese Funktion wird als Triggerung bezeichnet.

Diese Erläuterung soll als Übersicht über die Auslösekonzepte dienen. Eine vollständige Beschreibung finden Sie im Abschnitt IBM MQ-Anwendungen mit Triggern starten.

Plattformspezifische Informationen finden Sie in den folgenden Informationen:

- Für AIX, Linux, and Windows siehe „Kanäle unter AIX, Linux, and Windows auslösen” auf Seite 258
- IBM i Für IBM i siehe „Kanäle in IBM MQ for IBM i auslösen” auf Seite 258
- z/OS Für z/OS siehe „Übertragungswarteschlangen und Auslöserkanäle” auf Seite 1054

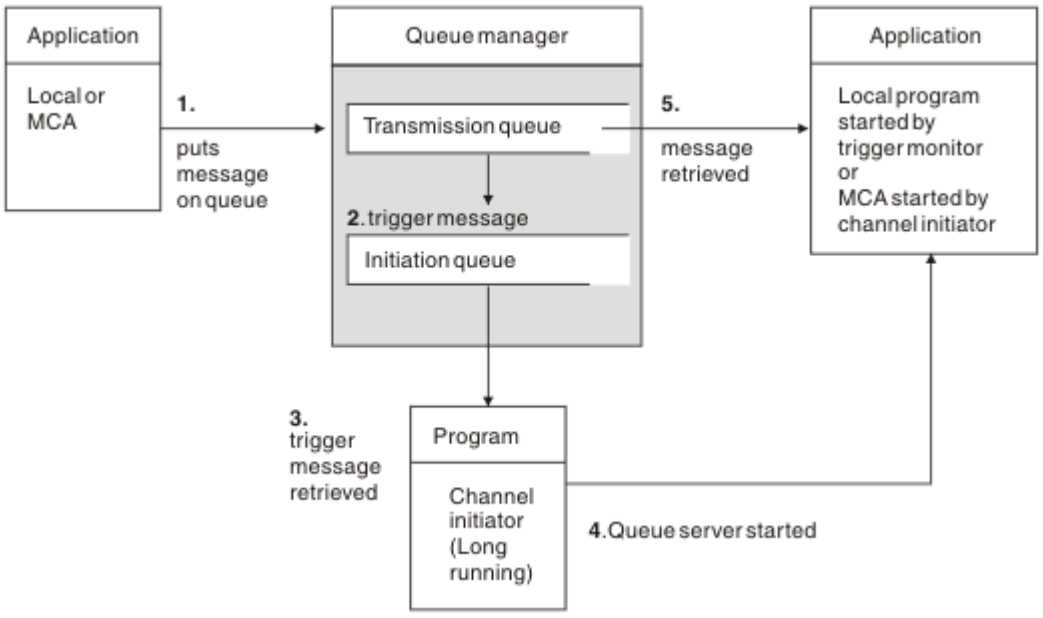



*Abbildung 24. Die Konzepte des Triggern*

Die für die Auslösung erforderlichen Objekte werden in Abbildung 24 auf Seite 257 angezeigt. Es zeigt die folgende Abfolge von Ereignissen an:

1. Der lokale WS-Manager stellt eine Nachricht von einer Anwendung oder von einem Nachrichtenkanalagenten (MCA) in die Übertragungswarteschlange.
2. Wenn die Auslöserbedingungen erfüllt sind, stellt der lokale WS-Manager eine Auslösenachricht in die Initialisierungswarteschlange.
3. Das Kanalinitiatorprogramm long-running überwacht die Initialisierungswarteschlange und ruft Nachrichten bei ihrer Ankunft ab.
4. Der Kanalinitiator verarbeitet die Auslösenachrichten in Übereinstimmung mit den darin enthaltenen Informationen. Diese Informationen können den Kanalnamen enthalten, in dem Fall, in dem der entsprechende MCA gestartet wird.
5. Die lokale Anwendung oder der MCA, der ausgelöst wurde, ruft die Nachrichten aus der Übertragungswarteschlange ab.

Um dieses Szenario zu konfigurieren, müssen Sie folgende Schritte ausführen:

- Erstellen Sie die Übertragungswarteschlange mit dem Namen der Initialisierungswarteschlange (das heißt SYSTEM.CHANNEL.INITQ) in dem entsprechenden Attribut.
- Stellen Sie sicher, dass die Initialisierungswarteschlange (SYSTEM.CHANNEL.INITQ) vorhanden ist.
- Stellen Sie sicher, dass das Kanalinitiatorprogramm verfügbar und aktiv ist. Das Kanalinitiatorprogramm muss in seinem Startbefehl mit dem Namen der Initialisierungswarteschlange angegeben werden.
  z/OS Unter z/OS ist der Name der Initialisierungswarteschlange festgelegt und wird daher nicht beim Startbefehl verwendet.
- Optional können Sie die Prozessdefinition für die Triggerung erstellen, falls sie nicht vorhanden ist, und stellen Sie sicher, dass das Feld *UserData* den Namen des Kanals enthält, der verwendet wird. Anstatt eine Prozessdefinition zu erstellen, können Sie den Kanalnamen in dem Attribut **TriggerData** der Übertragungswarteschlange angeben. IBM MQ für IBM i IBM i, AIX, Linux, and Windows-Systeme ermöglichen es, den Kanalnamen leer zu lassen; in diesem Fall wird die erste verfügbare Kanaldefinition mit dieser Übertragungswarteschlange verwendet.

- Stellen Sie sicher, dass die Definition der Übertragungswarteschlange den Namen der Prozessdefinition enthält, die sie bereitstellen soll (falls zutreffend), der Name der Initialisierungswarteschlange und die auslösenden Merkmale, die Sie fühlen, am besten geeignet sind. Das Auslösersteuerattribut ermöglicht es, dass die Auslösung aktiviert wird, oder nicht, falls erforderlich.

**Anmerkung:**

1. Das Kanalinitiatorprogramm dient als 'Auslösemonitor', der die Initialisierungswarteschlange überwacht, die zum Starten von Kanälen verwendet wird.
2. Es kann eine Initialisierungswarteschlange und ein Auslöserprozess verwendet werden, um eine beliebige Anzahl Kanäle auszulösen.
3. Es können eine beliebige Anzahl von Initialisierungswarteschlangen und Auslöserprozessen definiert werden.
4. Es wird empfohlen, einen Auslösertyp FIRST zu verwenden, um eine Überflutung des Systems mit Kanalstarts zu vermeiden.

## Kanäle unter AIX, Linux, and Windows auslösen


Sie können eine Prozessdefinition in IBM MQ erstellen und Prozesse definieren, die ausgelöst werden sollen. Verwenden Sie den WebSphere MQ-Scriptbefehl DEFINE PROCESS, um eine Prozessdefinition zu erstellen, die den Prozess benennt, der ausgelöst werden soll, wenn Nachrichten in einer Übertragungswarteschlange ankommen. Das Attribut USERDATA der Prozessdefinition enthält den Namen des Kanals, der von der Übertragungswarteschlange bedient wird.

Definieren Sie die lokale Warteschlange (QM4), und geben Sie an, dass Auslösenachrichten in die Initialisierungswarteschlange (IQ) geschrieben werden sollen, um die Anwendung auszulösen, die den Kanal startet (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

Definieren Sie die Anwendung (Prozess P1), die gestartet werden soll:

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

Alternativ können Sie bei IBM MQ for UNIX-, Linux- und Windows-Systemen auf eine Prozessdefinition verzichten und stattdessen im Attribut 'TRIGDATA' der Übertragungswarteschlange den Kanalnamen angeben.

Definieren Sie die lokale Warteschlange (QM4). Geben Sie an, dass Auslösenachrichten in die Standardinitialisierungswarteschlange SYSTEM.CHANNEL.INITQ geschrieben werden sollen, um die Anwendung (Prozess P1) auszulösen, die den Kanal startet (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

Wenn Sie keinen Kanalnamen angeben, durchsucht der Kanalinitiator die Kanaldefinitionsdateien, bis er einen Kanal findet, der der benannten Übertragungswarteschlange zugeordnet ist.

## Kanäle in IBM MQ for IBM i auslösen


Das Auslösen von Kanälen erfolgt in IBM MQ for IBM i über den Kanalinitiatorprozess. Ein Kanalinitiatorprozess für die Initialisierungswarteschlange SYSTEM.CHANNEL.INITQ wird automatisch mit dem Warteschlangenmanager gestartet, es sei denn, er wird durch Ändern des Attributs des Warteschlangenmanagers SCHINIT inaktiviert.

Richten Sie die Übertragungswarteschlange für den Kanal ein, und geben Sie SYSTEM.CHANNEL.INITQ als Initialisierungswarteschlange an, und aktivieren Sie die Auslösung für die Warteschlange. Der Kanalinitiator startet den ersten verfügbaren Kanal, der diese Übertragungswarteschlange angibt.

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)
USAGE(*TMQ)
```

Deprecated Sie können mit dem Befehl STRMQMCHLI manuell bis zu drei Kanalinitiatorprozesse starten und verschiedene Initialisierungswarteschlangen angeben. Sie können auch mehrere Kanäle angeben, die in der Lage sind, die Übertragungswarteschlange zu verarbeiten, und den zu startendem Kanal auswählen. Diese Funktionalität ist weiterhin für die Kompatibilität mit früheren Releases vorgesehen. Ihre Verwendung wird nicht weiter unterstützt.

**Anmerkung:** Nur jeweils ein Kanal kann eine Übertragungswarteschlange verarbeiten.

```
STRMQMCHLI QNAME(MYINITQ)
```

Richten Sie die Übertragungswarteschlange für den Kanal mit der Angabe TRGENBL (*YES) ein und geben Sie den Kanalnamen im Feld TRIGDATA an, um den Kanal auszuwählen, der gestartet werden soll. For example:

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(MYINITQ)
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

**Zugehörige Konzepte**

„Kanalinitiator starten und stoppen” auf Seite 259
Die Triggerung wird mithilfe des Kanalinitiatorprozesses implementiert.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

**Zugehörige Verweise**

Kanalprogramme unter AIX, Linux, and Windows

IBM i Jobs für übergreifende Kommunikation unter IBM i

IBM i Kanalzustände unter IBM i

## *Kanalinitiator starten und stoppen*

Die Triggerung wird mithilfe des Kanalinitiatorprozesses implementiert.

Dieser Kanalinitiatorprozess wird mit dem MQSC-Befehl START CHINIT gestartet. Wenn Sie nicht die Standardinitialisierungswarteschlange verwenden, geben Sie den Namen der Initialisierungswarteschlange im Befehl an. Wenn Sie beispielsweise den Befehl START CHINIT zum Starten der Warteschlange IQ für den Standardwarteschlangenmanager verwenden möchten, geben Sie Folgendes ein:

```
START CHINIT INITQ(IQ)
```

Standardmäßig wird ein Kanalinitiator automatisch mit der Standardinitialisierungswarteschlange SYSTEM.CHANNEL.INITQ. gestartet. Wenn Sie alle Kanalinitiatoren manuell starten möchten, führen Sie die folgenden Schritte aus:

1. Erstellen und starten Sie den WS-Manager.
2. Ändern Sie die Eigenschaft SCHINIT des WS-Managers in MANUAL.

3. Beenden Sie den WS-Manager und starten Sie ihn erneut

In IBM MQ for Multiplatforms-Systemen wird ein Kanalinitiator automatisch gestartet. Die Anzahl der Kanalinitiatoren, die gestartet werden können, ist begrenzt. Der Standardwert und der Maximalwert sind 3. Sie können dies mithilfe von MAXINITIATOREN in der Datei qm.ini für AIX and Linux-Systeme und in der Registry für Windows-Systeme ändern.

Details zum Ausführungskanalinitiatorbefehl **runmqchi**und zu den anderen Steuerbefehlen finden Sie unter IBM MQ -Steuerbefehle .

### Kanalinitiator stoppen

Der Standardkanalinitiator wird beim Starten eines Warteschlangenmanagers automatisch gestartet. Alle Kanalinitiatoren werden automatisch gestoppt, wenn ein WS-Manager gestoppt wird.

## Initialisierungs- und Konfigurationsdateien

Die Verarbeitung von Kanalinitialisierungsdaten hängt von Ihrer IBM MQ-Plattform ab.

### IBM MQ for z/OS

z/OS

In IBM MQ for z/OSwerden Initialisierungs-und Konfigurationsdaten mit dem MQSC-Befehl **ALTER QMGR** angegeben. Wenn Sie **ALTER QMGR** -Befehle in die Initialisierungseingabedatei CSQINP2 einreihen, werden sie bei jedem Start des Warteschlangenmanagers verarbeitet.

Um MQSC-Befehle wie **START LISTENER** bei jedem Start des Kanalinitiators auszuführen, stellen Sie sie in die Initialisierungseingabedatei CSQINPX und geben Sie die optionale Datendefinitionsanweisung CSQINPX in der Prozedur der gestarteten Task des Kanalinitiators an.

Weitere Informationen zu CSQINP2 und CSQINPX finden Sie unter Initialisierungseingabedateien anpassenund ALTER QMGR.

### IBM MQ for Multiplatforms

Multi

In IBM MQ for Multiplatformsgibt es Konfigurationsdateien für grundlegende Konfigurationsinformationen zur IBM MQ -Installation.

Es gibt zwei Konfigurationsdateien: eine gilt für die Maschine, die andere für einen einzelnen WS-Manager.

**Konfigurationsdatei IBM MQ**
Diese Datei enthält Informationen, die für alle WS-Manager auf dem IBM MQ-System relevant sind. Die Datei wird als `mqs.ini` bezeichnet. Eine Beschreibung finden Sie im Abschnitt „IBM MQ-Konfigurationsdatei, mqs.ini” auf Seite 91.

**Warteschlangenmanagerkonfigurationsdatei**

Diese Datei enthält Konfigurationsdaten, die sich auf einen bestimmten Warteschlangenmanager beziehen. Die Datei wird als `qm.ini` bezeichnet.

Sie wird während der Erstellung des Warteschlangenmanagers erstellt und kann Konfigurationsinformationen enthalten, die für alle Aspekte des Warteschlangenmanagers relevant sind. Zu den in der Datei enthaltenen Informationen gehören Details darüber, wie sich die Konfiguration des Protokolls von der Standardkonfiguration in der IBM MQ-Konfigurationsdatei unterscheidet.

Die Konfigurationsdatei des Warteschlangenmanagers wird im Stammverzeichnis der Verzeichnisstruktur, die vom Warteschlangenmanager belegt ist, gehalten. Für die **DefaultPath**-Attribute würden die Konfigurationsdateien des Warteschlangenmanagers für einen Warteschlangenmanager namens `WSMNAME` beispielsweise wie folgt lauten:

Für AIX and Linux-Systeme:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Für Windows-Systeme:

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```


Für IBM i:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

Hier ist ein Auszug aus einem `qm.ini`. Sie gibt an, dass der TCP/IP-Listener auf Port 2500 empfangsbereit ist, die maximale Anzahl der aktuellen Kanäle 200 ist und die maximale Anzahl aktiver Kanäle 100 beträgt.

```
TCP:
Port=2500
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

Sie können einen Bereich von TCP/IP-Ports angeben, die von einem abgehenden Kanal verwendet werden sollen. Eine Methode besteht darin, die `qm.ini`-Datei zu verwenden, um den Anfang und das Ende eines Bereichs von Portwerten anzugeben. Das folgende Beispiel zeigt eine `qm.ini`-Datei, die eine Reihe von Kanälen angibt:

```
TCP:
StrPort=2500
EndPort=3000
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

Wenn Sie einen Wert für **`StrPort`** oder **`EndPort`** angeben, müssen Sie einen Wert für beide angeben. Der Wert von **`EndPort`** muss immer größer als der Wert von **`StrPort`** sein.

Der Kanal versucht, die einzelnen Portwerte in dem angegebenen Bereich zu verwenden. Wenn die Verbindung erfolgreich hergestellt werden kann, ist der Portwert der Port, den der Kanal verwendet.

Weitere Informationen zu `qm.ini` -Dateien finden Sie unter „Warteschlangenmanagerkonfigurationsdateien, qm.ini” auf Seite 104.

## Datenkonvertierung für Nachrichten

IBM MQ-Nachrichten erfordern möglicherweise eine Datenkonvertierung, wenn sie zwischen Warteschlangen in verschiedenen Warteschlangenmanagern gesendet werden.

Eine IBM MQ-Nachricht besteht aus zwei Teilen:

- Steuerinformationen in einem Nachrichtendeskriptor
- Anwendungsdaten

Für beide Teile ist möglicherweise eine Datenkonvertierung erforderlich, wenn sie zwischen Warteschlangen in verschiedenen Warteschlangenmanagern gesendet wird. Informationen zur Anwendungsdatenkonvertierung finden Sie unter Anwendungsdatenkonvertierung .

## Schreiben eigener Nachrichtenkanalagenten

IBM MQ ermöglicht es Ihnen, Ihre eigenen Nachrichtenkanalagenten-Programme (MCA) zu schreiben oder ein Programm von einem unabhängigen Softwareanbieter zu installieren.

Möglicherweise möchten Sie Ihre eigenen MCA-Programme schreiben, um die Interaktionen von IBM MQ über Ihr eigenes proprietäres Kommunikationsprotokoll zu ermöglichen oder Nachrichten über ein Protokoll zu senden, das von IBM MQ nicht unterstützt wird. (Das Schreiben eines eigenes MCA-Programm für die Interaktion mit einem von IBM MQ bereitgestellten MCA am anderen Ende ist nicht möglich.)

Wenn Sie sich für die Verwendung eines nicht von IBM MQ bereitgestellten MCA entscheiden, müssen Sie die folgenden Punkte berücksichtigen.

**Senden und Empfangen von Nachrichten**
Sie müssen eine sendende Anwendung schreiben, die Nachrichten von überall dort abruft, wo Ihre Anwendung sie stellt, z. B. aus einer Übertragungswarteschlange, und sendet sie an ein Protokoll, mit dem Sie kommunizieren möchten. Sie müssen außerdem eine empfangende Anwendung schreiben, die Nachrichten aus diesem Protokoll aufnimmt und sie in Zielwarteschlangen stellt. Die sendenden und empfangenden Anwendungen verwenden die MQI-Aufrufe (Message Queue Interface, Nachrichtenwarteschlangenschnittstelle) und nicht alle speziellen Schnittstellen.

Sie müssen sicherstellen, dass Nachrichten nur einmal zugestellt werden. Die Synchronisationspunktkoordination kann zur Unterstützung bei dieser Zustellung verwendet werden.

**Kanalsteuerfunktion**
Sie müssen Ihre eigenen Verwaltungsfunktionen bereitstellen, um Kanäle zu steuern. Sie können die IBM MQ-Kanalverwaltungsfunktionen weder für die Konfiguration (z. B. den Befehl DEFINE CHANNEL) noch für die Überwachung (z. B. DISPLAY CHSTATUS) Ihrer Kanäle verwenden.

**Initialisierungsdatei**
Sie müssen eine eigene Initialisierungsdatei angeben, wenn Sie einen benötigen.

**Anwendungsdatenkonvertierung**
Möglicherweise möchten Sie die Datenkonvertierung für Nachrichten, die Sie an ein anderes System senden, ermöglichen. Ist dies der Fall, verwenden Sie die Option MQGMO_CONVERT im MQGET-Aufruf, wenn Sie Nachrichten von überall dort abrufen, wo Ihre Anwendung sie einreiht, z. B. die Übertragungswarteschlange.

**Benutzerexits**
Überlegen Sie, ob Sie Benutzerexits benötigen. Wenn dies der Fall ist, können Sie dieselben Schnittstellendefinitionen verwenden, die IBM MQ verwendet.

**Auslösefunktion**
Wenn Ihre Anwendung Nachrichten in eine Übertragungswarteschlange einreiht, können Sie die Attribute der Übertragungswarteschlange so konfigurieren, dass Ihr sendende Nachrichtenkanalgruppe ausgelöst wird, wenn Nachrichten in die Warteschlange eintreffen.

**Kanalinitiator**
Möglicherweise müssen Sie einen eigenen Kanalinitiator bereitstellen.

## Andere Aspekte, die für die verteilte Warteschlangenverwaltung zu berücksichtigen sind

Weitere Themen, die bei der Vorbereitung von IBM MQ für die verteilte Warteschlangenverwaltung zu berücksichtigen sind. In diesem Abschnitt finden Sie Informationen zu Nicht zugestellten Nachrichtenwarteschlangen, Warteschlangen in Verwendung, Systemerweiterungen und Benutzerexitprogrammen sowie zur Ausführung von Kanälen und Empfangsprogrammen als vertrauenswürdige Anwendungen.

### Nicht zugegebene Nachrichtenwarteschlange

Um sicherzustellen, dass Nachrichten, die in der Warteschlange für nicht zustellbare Nachrichten (auch die Warteschlange für den dead-letter oder DLQ genannt) ankommen, verarbeitet werden, müssen Sie ein Programm erstellen, das in regelmäßigen Abständen ausgelöst oder ausgeführt werden kann, um diese Nachrichten zu verarbeiten.

Linux AIX Ein DLQ-Handler wird mit IBM MQ auf AIX and Linux-Systemen bereitgestellt. Weitere Informationen hierzu finden Sie im Abschnitt The sample DLQ handler, amqsdlq.

IBM i Weitere Informationen zu IBM MQ for IBM i finden Sie im Abschnitt The IBM MQ for IBM i dead-letter queue handler.

## Warteschlangen im Gebrauch

MCAs für Empfängerkanäle können die Zielwarteschlangen auch dann offen halten, wenn Nachrichten nicht übertragen werden. Dies führt dazu, dass die Warteschlangen " im Gebrauch " angezeigt werden.

## Maximale Anzahl Kanäle

IBM i Unter IBM MQ for IBM i können Sie die maximale Anzahl der Kanäle angeben, die in Ihrem System zulässig sind, und die maximale Anzahl, die gleichzeitig aktiv sein kann. Sie geben diese Zahlen in der Datei `qm.ini` im Verzeichnis QIBM/UserData/mqm/qmgrs/*Warteschlangenmanagername*an. Informationen hierzu finden Sie im Abschnitt Zeilengruppen für Konfigurationsdateien für verteilte Steuerung von Warteschlangen .

## Systemerweiterungen und Benutzerexitprogramme

In der Kanaldefinition wird eine Funktion zur Verfügung gestellt, damit zusätzliche Programme während der Verarbeitung von Nachrichten zu definierten Zeitpunkten ausgeführt werden können. Diese Programme sind nicht im Lieferumfang von IBM MQ enthalten, können aber von jeder Installation entsprechend den lokalen Anforderungen bereitgestellt werden.

Damit diese Benutzerexitprogramme ausgeführt werden können, müssen sie über vordefinierte Namen verfügen und im Aufruf an die Kanalprogramme verfügbar sein. Die Namen der Benutzerexitprogramme sind in den Nachrichtenkanaldefinitionen enthalten.

Es gibt eine definierte Steuerblockschnittstelle für die Übergabe der Steuerung an diese Programme und für die Steuerung der Rückgabe der Steuerung von diesen Programmen.

Die genauen Stellen, an denen diese Programme aufgerufen werden, sowie Details zu Steuerblöcken und -namen befinden sich in Channel-Exit-Programme für Messaging-Kanäle.

## Kanäle und Empfangsprogramme als vertrauenswürdige Anwendungen ausführen

Wenn die Leistung eine wichtige Überlegung in Ihrer Umgebung ist und Ihre Umgebung stabil ist, können Sie Ihre Kanäle und Empfangsprogramme unter Verwendung der FASTPATH-Bindung als vertrauenswürdig ausführen. Es gibt zwei Faktoren, die beeinflussen, ob Kanäle und Empfangsprogramme als vertrauenswürdig ausgeführt werden:

- Die Umgebungsvariable MQ_CONNECT_TYPE=FASTPATH oder MQ_CONNECT_TYPE = STANDARD. Hierbei wird die Groß-/Kleinschreibung beachtet. Wenn Sie einen Wert angeben, der nicht gültig ist, wird er ignoriert.
- MQIBindType in der Zeilengruppe 'Channels' der `qm.ini`-oder Registry-Datei. Sie können diese Einstellung auf FASTPATH oder STANDARD setzen und die Groß-/Kleinschreibung muss nicht beachtet werden. Der Standardwert ist STANDARD.

Sie können MQIBindType in Verbindung mit der Umgebungsvariablen verwenden, um den erforderlichen Effekt wie folgt zu erzielen:

| **MQIBindType** | **Umgebungsvariable** | **Ergebnis** |
|---|---|---|
| STANDARD | NICHT DEFINIERT | STANDARD |
| FASTPATH | NICHT DEFINIERT | FASTPATH |
| STANDARD | STANDARD | STANDARD |
| FASTPATH | STANDARD | STANDARD |
| STANDARD | FASTPATH | STANDARD |

| MQIBindType | Umgebungsvariable | Ergebnis |
|---|---|---|
| FASTPATH | FASTPATH | FASTPATH |
| STANDARD | CLIENT | CLIENT |
| FASTPATH | CLIENT | STANDARD |
| STANDARD | LOKAL | STANDARD |
| FASTPATH | LOKAL | STANDARD |

In der Zusammenfassung gibt es nur zwei Möglichkeiten, Kanäle und Empfangsprogramme als vertrauenswürdig zu nutzen:

1. Durch Angabe von 'MQIBindType=FASTPATH' in `qm.ini` oder in der Registry und nicht unter Angabe der Umgebungsvariablen.
2. Durch Angabe von MQIBindType=FASTPATH in `qm.ini` oder in der Registry und Setzen der Umgebungsvariable auf FASTPATH.

Ziehen Sie die Ausführung von Empfangsprogrammen als vertrauenswürdig in Betracht, da Empfangsprogramme stabile Prozesse sind. Ziehen Sie die Ausführung von Kanälen als vertrauenswürdig in Betracht, es sei denn, Sie verwenden instabile Kanalexits oder den Befehl STOP CHANNEL MODE (TERMINATE).

## ALW Kanäle in AIX, Linux, and Windows überwachen und steuern

Für DQM müssen Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Sie können Kanäle mit Befehlen, Programmen, IBM MQ Explorer, Dateien für die Kanaldefinitionen und einem Speicherbereich für Synchronisationsinformationen steuern.

### Informationen zu diesem Vorgang

Sie können die folgenden Befehlstypen für die Steuerung von Kanälen verwenden:

**Die IBM MQ-Befehle (MQSC)**
Sie können den MQSC als Einzelbefehle in einer MQSC-Sitzung in AIX, Linux, and Windows-Systemen verwenden. Wenn Sie komplexere oder mehrere Befehle ausgeben möchten, kann der MQSC in eine Datei integriert werden, die Sie dann über die Befehlszeile ausführen. Weitere Informationen finden Sie in MQSC-Befehle . Dieser Abschnitt enthält einige einfache Beispiele für die Verwendung von MQSC für die verteilte Steuerung von Warteschlangen.

Bei den Kanalbefehlen handelt es sich um eine Untergruppe der IBM MQ-Befehle (MQSC). Sie verwenden MQSC und die Steuerbefehle wie folgt:

- Kanaldefinitionen erstellen, kopieren, anzeigen, ändern und löschen
- Kanäle starten und stoppen, Pingsignal absetzen, Kanalfolgenummern zurücksetzen und unbestätigte Nachrichten auflösen, wenn Links nicht erneut aufgebaut werden können
- Statusinformationen zu Kanälen anzeigen

**Steuerbefehle**
Sie können auch *Steuerbefehle* in der Befehlszeile für einige dieser Funktionen ausgeben. Weitere Informationen finden Sie unter IBM MQ for Multiplatforms mit Steuerbefehlen verwalten.

**Programmierbare Befehlsformatbefehle**
Weitere Informationen finden Sie in PCF-Befehle .

Windows Linux **IBM MQ Explorer**
Auf Linux- und Windows-Systemen können Sie die IBM MQ Explorer verwenden. Dies stellt eine grafische Verwaltungsschnittstelle zur Verfügung, mit der Verwaltungstasks als Alternative zur Verwendung von Steuerbefehlen oder MQSC-Befehlen ausgeführt werden können. Kanaldefinitionen werden als WS-Manager-Objekte gehalten.

Jeder WS-Manager verfügt über eine DQM-Komponente zur Steuerung von Verbindungen zu kompatiblen fernen Warteschlangenmanagern. Ein Speicherbereich enthält Folgenummern und *IDs der logischen Arbeitseinheit (Logical Unit of Work, LUW)* . Diese werden für Kanalsynchronisationszwecke verwendet.

Eine Liste der Funktionen, die Ihnen bei der Einrichtung und Steuerung von Nachrichtenkanälen mit den verschiedenen Befehlstypen zur Verfügung stehen, finden Sie in Tabelle 21 auf Seite 266.

## Prozedur

- „Erforderliche Funktionen für die Einrichtung und Steuerung von Kanälen” auf Seite 265
- „Erste Schritte mit Objekten” auf Seite 267
- „Kommunikation unter Windows einrichten” auf Seite 275
- „Kommunikation unter AIX and Linux einrichten” auf Seite 283

**Zugehörige Tasks**

„Kanäle in IBM i überwachen und steuern” auf Seite 289
Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Jeder WS-Manager verfügt über ein DQM-Programm zur Steuerung von Verbindungen zu kompatiblen fernen Warteschlangenmanagern.

**Zugehörige Verweise**

ALW Kanalprogramme unter AIX, Linux, and Windows

ALW Beispiel für Nachrichtenkanalplanung für AIX, Linux, and Windows

Beispielkonfigurationsdaten

Kanalattribute

# ALW Erforderliche Funktionen für die Einrichtung und Steuerung von Kanälen

Es kann eine Reihe von IBM MQ-Funktionen erforderlich sein, um Kanäle einzurichten und zu steuern. Die Kanalfunktionen werden in diesem Thema erläutert.

Sie können eine Kanaldefinition mit den von IBM MQ bereitgestellten Standardwerten erstellen. Geben Sie dabei den Namen des Kanals, den Typ des zu erstellenden Kanals, die zu verwendende Übertragungsmethode, den Namen der Übertragungswarteschlange und den Verbindungsnamen an.

Der Kanalname muss an beiden Enden des Kanals identisch sein und innerhalb des Netzes eindeutig sein. Die verwendbaren Zeichen müssen jedoch auf diejenigen eingeschränkt werden, die für IBM MQ-Objektnamen gültig sind.

Informationen zu anderen kanalbezogenen Funktionen finden Sie in den folgenden Abschnitten:

- „Erste Schritte mit Objekten” auf Seite 267
- „Erstellen von zugeordneten Objekten” auf Seite 268
- „Standardobjekte erstellen” auf Seite 268
- „Kanal erstellen” auf Seite 269
- „Anzeigen eines Kanals” auf Seite 269
- „Kanalstatus anzeigen” auf Seite 270
- „Links mit Ping überprüfen” auf Seite 270
- „Kanal starten” auf Seite 271
- „-Kanal stoppen” auf Seite 272
- „Kanal umbenennen” auf Seite 273
- „Kanal zurücksetzen” auf Seite 273
- „Unbestätigte Nachrichten in einem Kanal auflösen” auf Seite 274

In Tabelle 21 auf Seite 266 ist die komplette Liste der IBM MQ-Funktionen angezeigt, die Sie eventuell benötigen.

*Tabelle 21. In AIX, Linux, and Windows-Systemen erforderliche Funktionen*

| **Funktion** | **Steuerbefehle** | **MQSC** | **IBM MQ Explorer-Entsprechung?** |
|---|---|---|---|
| WS-Managerfunktionen | | | |
| Warteschlangenmanager ändern | | ALTER QMGR | Ja |
| Warteschlangenmanager erstellen | crtmqm | | Ja |
| WS-Manager löschen | dltmqm | | Ja |
| WS-Manager anzeigen | | ANZEIGEN QMGR | Ja |
| WS-Manager beenden | endmqm | | Ja |
| Ping-WS-Manager | | PING QMGR | Nein |
| WS-Manager starten | strmqm | | Ja |
| Befehlsserverfunktionen | | | |
| Befehlsserver anzeigen | dspmqcsv | | Nein |
| Befehlsserver beenden | endmqcsv | | Nein |
| Befehlsserver starten | strmqcsv | | Nein |
| Warteschlangenfunktionen | | | |
| Warteschlange ändern | | ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE<br>Siehe, ALTER-Warteschlangen . | Ja |
| Warteschlange löschen | | CLEAR QLOCAL | Ja |
| Warteschlange erstellen | | DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE<br>Siehe, DEFINE Queues . | Ja |
| Warteschlange löschen | | DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE<br>Siehe DELETE-Warteschlangen . | Ja |
| Warteschlange anzeigen | | ANZEIGEN QUEUE | Ja |
| Prozessfunktionen | | | |
| Änderungsprozess | | ALTER PROCESS | Ja |
| Prozess erstellen | | DEFINE PROCESS | Ja |

| *Tabelle 21. In AIX, Linux, and Windows-Systemen erforderliche Funktionen (Forts.)* | | | |
|---|---|---|---|
| **Funktion** | **Steuerbefehle** | **MQSC** | **IBM MQ Explorer-Entsprechung?** |
| Prozess löschen | | DELETE PROCESS | Ja |
| Anzeigeprozess | | ANZEIGEN PROZESS | Ja |
| Kanalfunktionen | | | |
| Kanal ändern | | ALTER CHANNEL | Ja |
| Kanal erstellen | | DEFINE CHANNEL | Ja |
| Kanal löschen | | DELETE CHANNEL | Ja |
| Kanal anzeigen | | ANZEIGEN CHANNEL | Ja |
| Kanalstatus anzeigen | | ANZEIGEN CHSTATUS | Ja |
| Kanal beenden | | STOP CHANNEL | Ja |
| Pingkanal | | Pingkanal | Ja |
| Kanal zurücksetzen | | Kanal zurücksetzen | Ja |
| Kanal auflösen | | Auflösungskanal | Ja |
| Kanal ausführen | runmqchl | START CHANNEL | Ja |
| Kanalinitiator ausführen | runmqchi | START CHINIT | Nein |
| Listener ausführen [1] | runmqlsr | START LISTENER | Nein |
| Listener beenden | endmqlsr , nur auf den folgenden Plattformen:<br>• AIX AIX<br>• Windows Systeme mit Windows | | Nein |

**Anmerkung:**

1. Ein Listener kann beim Start des Warteschlangenmanagers automatisch gestartet werden.

## ALW Erste Schritte mit Objekten

Kanäle müssen definiert sein, und die zugehörigen Objekte müssen vorhanden und verfügbar sein, bevor ein Kanal gestartet werden kann. In diesem Abschnitt wird gezeigt, wie.

Verwenden Sie die IBM MQ-Befehle (MQSC) oder IBM MQ Explorer für folgende Zwecke:

1. Nachrichtenkanäle und zugehörige Objekte definieren
2. Nachrichtenkanäle überwachen und steuern

Zu den zugeordneten Objekten, die Sie möglicherweise definieren müssen, gehören:

- Übertragungswarteschlangen
- Definitionen ferner Warteschlangen
- WS-Manager-Aliasdefinitionen

- Aliasnamendefinitionen für Antwortwarteschlange
- Antwort-in lokale Warteschlangen
- Prozesse für Triggerung (MCAs)
- Nachrichtenkanaldefinitionen

Die jeweilige Kommunikationsverbindung für jeden Kanal muss definiert und verfügbar sein, bevor ein Kanal ausgeführt werden kann. Eine Beschreibung der Definition von LU 6.2-, TCP/IP-, NetBIOS-, SPX-und DECnet-Links finden Sie in dem jeweiligen Kommunikationshandbuch für Ihre Installation. Siehe auch Beispielkonfigurationsdaten .

Weitere Informationen zum Erstellen und Arbeiten mit Objekten finden Sie in den folgenden Unterabschnitten:

### ALW *Erstellen von zugeordneten Objekten*

MQSC wird zum Erstellen von zugeordneten Objekten verwendet.

Verwenden Sie MQSC zum Erstellen der Warteschlangen-und Aliasobjekte: Übertragungswarteschlangen, Definitionen für ferne Warteschlangen, Definitionen von WS-Manager-Aliasnamen, Antwortwarteschlangen-Aliasdefinitionen und Antwort-in-lokale Warteschlangen.

Erstellen Sie außerdem die Definitionen von Prozessen für die Auslösung (MCAs) auf ähnliche Weise.

Ein Beispiel für die Erstellung aller erforderlichen Objekte finden Sie im Abschnitt Beispiel für Nachrichtenkanalplanung für AIX, Linux, and Windows.

### ALW *Standardobjekte erstellen*

Standardobjekte werden automatisch erstellt, wenn ein Warteschlangenmanager erstellt wird. Bei diesen Objekten handelt es sich um Warteschlangen, Kanäle, eine Prozessdefinition und Verwaltungswarteschlangen. Nachdem die Standardobjekte erstellt wurden, können Sie sie jederzeit ersetzen, indem Sie den Befehl strmqm mit der Option -c ausführen.

Wenn Sie den Befehl crtmqm zum Erstellen eines Warteschlangenmanagers verwenden, leitet der Befehl auch ein Programm ein, um eine Gruppe von Standardobjekten zu erstellen.

1. Jedes Standardobjekt wird wiederum erstellt. Das Programm protokolliert, wie viele Objekte erfolgreich definiert wurden, wie viele vorhanden waren und ersetzt wurden, und wie viele erfolglose Versuche es gab.
2. Das Programm zeigt die Ergebnisse an, und wenn Fehler aufgetreten sind, leitet Sie das entsprechende Fehlerprotokoll auf die Details zu.

Wenn die Ausführung des Programms beendet ist, können Sie den Warteschlangenmanager mit dem Befehl strmqm starten.

Weitere Informationen zu den Befehlen 'crtmqm' und 'strmqm ' finden Sie unter IBM MQ for Multiplatforms mit Steuerbefehlen verwalten .

## Ändern der Standardobjekte

Wenn Sie die Option -c angeben, wird der Warteschlangenmanager vorübergehend gestartet, während die Objekte erstellt werden, und wird anschließend wieder heruntergefahren. Wenn Sie strmqm mit der Option -c absetzen, werden vorhandene Systemobjekte mit den Standardwerten aktualisiert (z. B. wird das Attribut MCAUSER einer Kanaldefinition auf Leerzeichen gesetzt). Sie müssen den Befehl strmqm erneut verwenden, ohne die Option -c zu verwenden, wenn Sie den Warteschlangenmanager starten wollen.

Wenn Sie die Standardobjekte ändern möchten, können Sie eine eigene Version der alten Datei 'amqscoma.tst' erstellen und diese bearbeiten.

### ALW *Kanal erstellen*

Erstellen Sie zwei Kanaldefinitionen, eine an jedem Ende der Verbindung. Sie erstellen die erste Kanaldefinition auf dem ersten Warteschlangenmanager. Anschließend erstellen Sie die zweite Kanaldefinition am zweiten WS-Manager am anderen Ende der Verbindung.

Beide Enden müssen mit demselben Kanalnamen definiert werden. Die beiden Enden müssen kompatible Kanaltypen haben, z. B. Sender und Empfänger.

Verwenden Sie den MQSC-Befehl DEFINE CHANNEL, um eine Kanaldefinition für ein Ende der Verbindung zu erstellen. Geben Sie den Namen des Kanals, den Kanaltyp für dieses Ende der Verbindung, einen Verbindungsnamen, eine Beschreibung (falls erforderlich), den Namen der Übertragungswarteschlange (falls erforderlich) und das Übertragungsprotokoll an. Geben Sie außerdem alle anderen Attribute an, die von den Systemstandardwerten für den erforderlichen Kanaltyp verschieden sein sollen, und verwenden Sie dabei die zuvor erfassten Informationen.

Sie erhalten Hilfe bei der Entscheidung über die Werte der Kanalattribute in Kanalattribute .

**Anmerkung:** Es wird empfohlen, alle Kanäle in Ihrem Netzwerk eindeutig zu benennen. Dies ist eine gute Möglichkeit, die Namen der Quellen-und Zielwarteschlangenmanager in den Kanalnamen zu berücksichtigen.

## Kanalbeispiel erstellen

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +
DESCR('Sender channel to QM2') +
CONNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

In allen MQSC-Beispielen wird der Befehl in einer Datei mit Befehlen angezeigt und wie in AIX, Linux, and Windows angegeben. Die beiden Methoden sehen identisch aus, es sei denn, dass Sie einen Befehl interaktiv absetzen müssen. Sie müssen zunächst eine MQSC-Sitzung starten. Geben Sie `runmqsc`für den Standardwarteschlangenmanager oder `runmqsc` *qmname* ein, wobei *qmname* der Name des erforderlichen Warteschlangenmanagers ist. Geben Sie dann eine beliebige Anzahl Befehle ein, wie in den Beispielen gezeigt.

Beschränken Sie die Zeilenlänge Ihrer Befehle auf 72 Zeichen, um die Portierbarkeit zu begrenzen. Verwenden Sie das Verkettungszeichen ( + ), wie in der folgenden Abbildung dargestellt, um mehr als eine Zeile zu verwenden:

- Windows Verwenden Sie unter Windows die Tastenkombination Strg + Z, um den Eintrag in der Befehlszeile zu beenden.
- Linux AIX Verwenden Sie unter AIX and Linux die Tastenkombination Strg + d.
- Alternativ dazu können Sie unter AIX, Linux, and Windows den Befehl **end** verwenden.

### ALW *Anzeigen eines Kanals*

Verwenden Sie den WebSphere MQ-Scriptbefehl DISPLAY CHANNEL, um die Attribute eines Kanals anzuzeigen.

Der Parameter ALL des Befehls DISPLAY CHANNEL wird standardmäßig angenommen, wenn keine bestimmten Attribute angefordert werden, und der angegebene Kanalname nicht generisch ist.

Die Attribute werden in Kanalattribute beschrieben.

## Kanalbeispiele anzeigen

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE,CONVERT

DISPLAY CHANNEL(QM1.TO.*) TRPTYPE,CONVERT

DISPLAY CHANNEL(*) TRPTYPE,CONVERT
```

```
DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

### ALW *Kanalstatus anzeigen*

Verwenden Sie den WebSphere MQ-Scriptbefehl DISPLAY CHSTATUS, geben Sie den Kanalnamen an und geben Sie an, ob der aktuelle Status von Kanälen oder der Status von gespeicherten Informationen angezeigt werden soll.

DISPLAY CHSTATUS gilt für alle Nachrichtenkanäle. Sie gilt nicht für MQI-Kanäle, die keine Serververbindungskanäle sind.

Zu den angezeigten Informationen gehören:

- Kanalname
- Kommunikationsverbindungsname
- Unbestätigte Kanalstatus (falls erforderlich)
- Letzte Folgenummer
- Name der Übertragungswarteschlange (falls erforderlich)
- Die unbestätigte ID (falls erforderlich)
- Die zuletzt festgeschriebene Folgenummer
- ID der logischen Arbeitseinheit
- Prozess-ID
- Windows Thread-ID (nur Windows)

## Kanalstatusbeispiele anzeigen

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

Der gespeicherte Status gilt erst dann, wenn mindestens ein Nachrichtenstachsatz auf dem Kanal übertragen wurde. Der Status wird auch gespeichert, wenn ein Kanal gestoppt wird (mit dem Befehl STOP CHL) und wenn der Warteschlangenmanager beendet wird.

### ALW *Links mit Ping überprüfen*

Verwenden Sie den MQSC-Befehl **`PING CHANNEL`** , um eine feste Datennachricht mit dem fernen Ende auszutauschen.

Ping gibt dem Systembetreuer ein wenig Vertrauen, dass der Link verfügbar und funktionsfähig ist.

Ping bedeutet nicht die Verwendung von Übertragungswarteschlangen und Zielwarteschlangen. Sie verwendet Kanaldefinitionen, die zugehörige Kommunikationsverbindung und die Netzkonfiguration. Es kann nur verwendet werden, wenn der Kanal momentan nicht aktiv ist.

Sie ist nur über Sender-, Server-und Clustersenderkanäle verfügbar. Der entsprechende Kanal wird an der fernen Seite des Links gestartet und führt die Startparametervereinbarung aus. Fehler werden normal benachrichtigt.

Das Ergebnis des Nachrichtenaustauschs wird als `Ping complete` oder als Fehlernachricht angezeigt.

## Ping mit LU 6.2

Wenn Ping aufgerufen wird, wird standardmäßig keine Benutzer-ID oder kein Kennwort an das empfangende Ende fließen. Wenn die Benutzer-ID und das Kennwort erforderlich sind, können sie am einleitenden Ende in der Kanaldefinition erstellt werden. Wenn ein Kennwort in die Kanaldefinition eingegeben

wird, wird es von IBM MQ verschlüsselt, bevor es gespeichert wird. Anschließend wird sie entschlüsselt, bevor sie über den Datenaustausch fließt.

**Zugehörige Tasks**

Ping zum Testen der Kommunikation verwenden

Ping-Abfrage für einen Kanal, um eine Verbindung zu prüfen

**Zugehörige Verweise**

PING CHANNEL (Antwort des Testkanals)

## ALW *Kanal starten*

Verwenden Sie den MQSC-Befehl START CHANNEL für Sender-, Server-und Requesterkanäle. Damit Anwendungen Nachrichten austauschen können, müssen Sie ein Empfangsprogramm für eingehende Verbindungen starten.

START CHANNEL ist nicht erforderlich, wenn ein Kanal mit Warteschlangenmanagerauslösung konfiguriert wurde.

Nach dem Start liest der sendende MCA die Kanaldefinitionen und öffnet die Übertragungswarteschlange. Es wird eine Kanalstartsequenz ausgegeben, die den entsprechenden Nachrichtenkanalserver (MCA) des Empfängers oder Serverkanals über Remotezugriff startet. Wenn sie gestartet wurden, warten die Absender-und Serverprozesse auf Nachrichten, die in die Übertragungswarteschlange eintreffen und sie bei ihrer Ankunft übertragen.

Wenn Sie als Threads Trigger-oder Ausführungskanäle verwenden, stellen Sie sicher, dass der Kanalinitiator für die Überwachung der Initialisierungswarteschlange verfügbar ist. Der Kanalinitiator wird standardmäßig als Teil des Warteschlangenmanagers gestartet.

TCP und LU 6.2 stellen jedoch andere Funktionen bereit:

- Linux AIX Für TCP unter AIX and Linux kann inetd so konfiguriert werden, dass ein Kanal gestartet wird. inetd wird als separater Prozess gestartet.
- Linux AIX Für LU 6.2 in AIX and Linux konfigurieren Sie Ihr SNA-Produkt so, dass der LU 6.2-Responder-Prozess gestartet wird.
- Windows Für LU 6.2 in Windows können Sie mit SNA Server TpStart (ein Dienstprogramm, das mit dem SNA-Server bereitgestellt wird) verwenden, um einen Kanal zu starten. TpStart wird als separater Prozess gestartet.

Die Verwendung der Option Start bewirkt, dass der Kanal bei Bedarf immer resynchronisiert wird.

Damit der Start erfolgreich ist:

- Kanaldefinitionen, lokale und ferne, müssen vorhanden sein. Wenn für einen Empfänger-oder Serververbindungskanal keine entsprechende Kanaldefinition vorhanden ist, wird automatisch ein Standardkanal erstellt, wenn der Kanal automatisch definiert wird. Siehe Exitprogramm für die automatische Kanaldefinition (Channel Auto-Definition)
- Die Übertragungswarteschlange muss vorhanden sein, und sie dürfen keine anderen Kanäle verwenden.
- MCAs, lokale und ferne, müssen vorhanden sein.
- Die Kommunikationsverbindung muss verfügbar sein.
- WS-Manager müssen aktiv, lokal und fern ausgeführt werden.
- Der Nachrichtenkanal darf nicht bereits aktiv sein.

Es wird eine Nachricht an die Anzeige zurückgegeben, in der bestätigt wird, dass die Anforderung zum Starten eines Kanals akzeptiert wurde. Überprüfen Sie zur Bestätigung, dass der Startbefehl erfolgreich war, das Fehlerprotokoll, oder verwenden Sie DISPLAY CHSTATUS. Die Fehlerprotokolle sind:

**Windows**

*MQ_DATA_PATH*\qmgrs\qmname\errors\AMQERR01.LOG (für jeden Warteschlangenmanager mit dem Namen qmname)

*MQ_DATA_PATH*\qmgrs\@SYSTEM\errors\AMQERR01.LOG (für allgemeine Fehler)

*MQ_DATA_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

**Anmerkung:** Unter Windows erhalten Sie weiterhin auch eine Nachricht im Anwendungsereignisprotokoll von Windows-Systemen.

**AIX and Linux**

/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG (für jeden Warteschlangenmanager mit dem Namen qmname)

/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG (für allgemeine Fehler)

Verwenden Sie unter AIX, Linux, and Windows den Befehl **runmqlsr**, um den IBM MQ-Listenerprozess zu starten. Standardmäßig bewirkt jeder eingehende Anforderungen für den Kanalanschluss, dass der Listenerprozess MCAs als Threads des Prozesses 'amqrmppa' startet.

```
runmqlsr -t tcp -m QM2
```

Für abgehende Verbindungen müssen Sie den Kanal auf eine der folgenden drei Arten starten:

1. Verwenden Sie den MQSC-Befehl START CHANNEL, und geben Sie dabei den Kanalnamen an, um den Kanal als Prozess oder als Thread zu starten, abhängig vom Parameter MCATYPE. (Wenn Kanäle als Threads gestartet werden, handelt es sich um Threads eines Kanalinitiators.)

   ```
   START CHANNEL(QM1.TO.QM2)
   ```

2. Verwenden Sie den Steuerbefehl runmqchl, um den Kanal als Prozess zu starten.

   ```
   runmqchl -c QM1.TO.QM2 -m QM1
   ```

3. Verwenden Sie den Kanalinitiator, um den Kanal auszulösen.

### *-Kanal stoppen*

Verwenden Sie den MQSC-Befehl STOP CHANNEL, um den Kanal anzufordern, die Aktivität zu stoppen. Der Kanal startet keinen neuen Stapel von Nachrichten, bis der Bediener den Kanal erneut startet.

Informationen zum erneuten Starten von gestoppten Kanälen finden Sie in „Gestoppte Kanäle erneut starten” auf Seite 251.

Dieser Befehl kann an einen Kanal jedes Typs mit Ausnahme von MQCHT_CLNTCONN ausgegeben werden.

Sie können den Typ des erforderlichen Stoppes auswählen:

## Quiesce-Beispiel stoppen

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

Mit diesem Befehl wird der Kanal in geordneter Weise geschlossen, um sie ordnungsgemäß zu schließen. Der aktuelle Nachrichtenstationsstapel ist abgeschlossen, und die Synchronisationspunktprozedur wird mit dem anderen Ende des Kanals ausgeführt. Wenn der Kanal inaktiv ist, beendet dieser Befehl keinen empfangenden Kanal.

### Beispiel für Stoppkraft

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

Mit dieser Option wird der Kanal sofort gestoppt, aber der Thread oder der Prozess des Kanals wird nicht beendet. Der Kanal beendet die Verarbeitung des aktuellen Nachrichtenstroms nicht vollständig und kann daher den Kanal im Zweifel lassen. Im Allgemeinen sollten Sie die Option für Stilllegung in den Wartemodus (Quiesce stop

### Stoppbeispiel stoppen

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

Mit dieser Option wird der Kanal sofort gestoppt und der Thread oder der Prozess des Kanals beendet.

### Stoppbeispiel stoppen (Quiesce)

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

Für diesen Befehl ist kein MODE angegeben, daher wird standardmäßig MODE (QUIESCE) angenommen. Er fordert den Kanal auf, den Kanal zu stoppen, damit er nicht automatisch erneut gestartet werden kann, sondern manuell gestartet werden muss.

### Inaktives Beispiel stoppen (Quiesce)

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

Für diesen Befehl ist kein MODE angegeben, daher wird standardmäßig MODE (QUIESCE) angenommen. Er fordert die Inaktivierung des Kanals an, so dass er bei Bedarf automatisch erneut gestartet wird.

## ALW *Kanal umbenennen*

Verwenden Sie MQSC, um einen Nachrichtenkanal umzubenennen.

Verwenden Sie MQSC, um die folgenden Schritte auszuführen:

1. Verwenden Sie STOP CHANNEL, um den Kanal zu stoppen.
2. Verwenden Sie DEFINE CHANNEL, um eine doppelte Kanaldefinition mit dem neuen Namen zu erstellen.
3. Verwenden Sie DISPLAY CHANNEL, um zu überprüfen, ob er korrekt erstellt wurde.
4. Verwenden Sie DELETE CHANNEL, um die ursprüngliche Kanaldefinition zu löschen.

Wenn Sie einen Nachrichtenkanal umbenennen möchten, müssen Sie daran denken, dass ein Kanal über zwei Kanaldefinitionen verfügt, eine an jedem Ende. Stellen Sie sicher, dass Sie den Kanal an beiden Enden gleichzeitig umbenennen.

## ALW *Kanal zurücksetzen*

Verwenden Sie den MQSC-Befehl RESET CHANNEL, um die Nachrichtenfolgenummer zu ändern.

Der Befehl RESET CHANNEL ist für jeden Nachrichtenkanal verfügbar, aber nicht für MQI-Kanäle (Clientverbindung oder Serververbindung). Die erste Nachricht startet die neue Sequenz, wenn der Kanal das nächste Mal gestartet wird.

Wenn der Befehl auf einem Sender-oder Serverkanal abgesetzt wird, informiert er die andere Seite der Änderung, wenn der Kanal erneut gestartet wird.

**Zugehörige Konzepte**

„Erste Schritte mit Objekten” auf Seite 267
Kanäle müssen definiert sein, und die zugehörigen Objekte müssen vorhanden und verfügbar sein, bevor ein Kanal gestartet werden kann. In diesem Abschnitt wird gezeigt, wie.

„Kanalsteuerfunktion” auf Seite 239
Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

**Zugehörige Verweise**

RESET CHANNEL

## ALW *Unbestätigte Nachrichten in einem Kanal auflösen*

Verwenden Sie den MQSC-Befehl RESOLVE CHANNEL, wenn Nachrichten von einem Sender oder Server im Zweifel gehalten werden. Zum Beispiel, weil ein Ende der Verbindung beendet wurde und es keine Aussicht auf eine Wiederherstellung gibt.

Der Befehl RESOLVE CHANNEL akzeptiert einen der beiden folgenden Parameter: BACKOUT oder COMMIT. Mit Backout werden Nachrichten in die Übertragungswarteschlange zurückgespeichert, während Commit sie löscht.

Das Kanalprogramm versucht nicht, eine Sitzung mit einem Partner aufzubauen. Stattdessen bestimmt sie die ID der logischen Arbeitseinheit (LUWID), die die unbestätige_Nachrichten darstellt. Anschließend gibt es, wie angefordert, folgende Probleme aus:

- BACKOUT, um die Nachrichten in die Übertragungswarteschlange zurückzuspeichen; oder
- COMMIT, um die Nachrichten aus der Übertragungswarteschlange zu löschen.

Damit die Auflösung erfolgreich ist:

- Der Kanal muss inaktiv sein.
- Der Kanal muss im Zweifel sein.
- Der Kanaltyp muss "sender", "server" oder "cluster-sender" sein.
- Es muss eine lokale Kanaldefinition vorhanden sein.
- Der lokale WS-Manager muss aktiv sein.

**Zugehörige Konzepte**

„Erste Schritte mit Objekten” auf Seite 267
Kanäle müssen definiert sein, und die zugehörigen Objekte müssen vorhanden und verfügbar sein, bevor ein Kanal gestartet werden kann. In diesem Abschnitt wird gezeigt, wie.

„Kanalsteuerfunktion” auf Seite 239
Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

**Zugehörige Verweise**

GELÖST-CHANNEL

# Windows Kommunikation unter Windows einrichten

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dies gelingt, muss die Verbindung definiert und verfügbar sein. In diesem Abschnitt wird erläutert, wie Sie dies tun, indem Sie die Kommunikationsformen verwenden, die für IBM MQ for Windows-Systeme verfügbar sind.

## Vorbereitende Schritte

Es kann hilfreich sein, auch im Abschnitt Beispielkonfiguration - IBM MQ for Windows nachzulesen.

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

## Informationen zu diesem Vorgang

Wenn Sie die Kommunikation für IBM MQ unter Windows einrichten, können Sie zwischen den folgenden Arten von Kommunikation wählen:

- TCP/IP
- LU 6.2
- NetBIOS

## Prozedur

- Informationen zum Konfigurieren der Kommunikation für Ihr Windows-System finden Sie im Unterabschnitt für den ausgewählten Kommunikationstyp:
  - „TCP-Verbindung unter Windows definieren” auf Seite 276
  - „LU 6.2-Verbindung unter Windows definieren” auf Seite 278
  - „NetBIOS-Verbindung in Windows definieren” auf Seite 280

  Nicht alle Funktionen und Einrichtungen von IBM MQ for Windows sind in Umgebungen verfügbar, die ein anderes Kommunikationsprotokoll als TCP/IP verwenden. Das nicht verfügbare Element ist IBM MQ Explorer.

### Zugehörige Tasks

„Kanäle in AIX, Linux, and Windows überwachen und steuern” auf Seite 264
Für DQM müssen Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Sie können Kanäle mit Befehlen, Programmen, IBM MQ Explorer, Dateien für die Kanaldefinitionen und einem Speicherbereich für Synchronisationsinformationen steuern.

„Verbindungen zwischen Client und Server konfigurieren” auf Seite 15
Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

„Kommunikation unter AIX and Linux einrichten” auf Seite 283
Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dies gelingt, muss die Verbindung definiert und verfügbar sein. In diesem Abschnitt wird erläutert, wie Sie dies tun, indem Sie die Kommunikationsformen verwenden, die für IBM MQ for UNIX or Linux-Systeme verfügbar sind.

### Zugehörige Verweise

„Zu verwendende Übertragungsart” auf Seite 16

Unterschiedliche Plattformen unterstützen unterschiedliche Kommunikationsprotokolle. Ihre Auswahl des Übertragungsprotokolls hängt von Ihrer Kombination von IBM MQ MQI client- und Serverplattformen ab.

## Windows *TCP-Verbindung unter Windows definieren*

Definieren Sie eine TCP-Verbindung, indem Sie einen Kanal auf der sendenden Seite konfigurieren, um die Adresse des Ziels anzugeben, und indem Sie ein Empfangsprogramm auf der Empfangsseite ausführen.

### Vorbereitungen

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

### Sendende Beendigung

Geben Sie den Hostnamen oder die TCP-Adresse der Zielmaschine in das Feld Verbindungsname der Kanaldefinition an.

Der Port, zu dem die Verbindung hergestellt werden soll, standardmäßig 1414. Die Portnummer 1414 ist IBM MQ von der Internet Assigned Numbers Authority (IANA) zugewiesen.

Wenn Sie eine andere Portnummer als die Standardportnummer verwenden möchten, geben Sie sie im Feld für den Verbindungsnamen der Kanalobjektdefinition an:

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +
          TRPTYPE(TCP) +
          CONNAME('OS2ROG3(1822)') +
          XMITQ('XMitQ name') +
          REPLACE
```

Dabei steht 0S2R0G3 für den DNS-Namen des fernen Warteschlangenmanagers und 1822 für den erforderlichen Port. (Dies muss der Port sein, an dem der Listener empfangsbereit ist.)

Ein Laufkanal muss gestoppt und erneut gestartet werden, um eine Änderung an der Kanalobjektdefinition abzuholen.

Sie können die Standardportnummer ändern, indem Sie sie in der `.ini`-Datei für IBM MQ for Windows angeben:

```
TCP:
Port=1822
```

**Anmerkung:** Um die TCP/IP-Portnummer auszuwählen, die verwendet werden soll, verwendet IBM MQ die erste Portnummer, die in der folgenden Reihenfolge gefunden wird:

1. Die Portnummer, die explizit in der Kanaldefinition oder Befehlszeile angegeben wurde. Mit dieser Zahl kann die Standardportnummer für einen Kanal außer Kraft gesetzt werden.
2. Das Portattribut, das in der TCP-Zeilengruppe der `.ini`-Datei angegeben ist. Mit dieser Zahl kann die Standardportnummer für einen WS-Manager überschrieben werden.
3. Der Standardwert ist 1414. Dies ist die Nummer, die IBM MQ von der Internet Assigned Numbers Authority für eingehende und abgehende Verbindungen zugeordnet ist.

Weitere Informationen zu den Werten, die Sie mit qm.ini festlegen, finden Sie unter Zeilengruppen für Konfigurationsdateien für verteilte Steuerung von Warteschlangen .

## Empfang auf TCP

Um ein empfangendes Kanalprogramm zu starten, muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Sie können den IBM MQ-Listener verwenden.

Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet.

Um ein empfangendes Kanalprogramm zu starten, muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Sie können den IBM MQ-Listener verwenden.

Wenn Sie den mit IBM MQ bereitgestellten Listener ausführen möchten, der neue Kanäle als Threads startet, verwenden Sie den Befehl runmqlsr.

Ein grundlegendes Beispiel für die Verwendung des Befehls **runmqlsr**:

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

Die eckigen Klammern geben optionale Parameter an; QMNAME ist für den Standard-WS-Manager nicht erforderlich, und die Portnummer ist nicht erforderlich, wenn Sie die Standardeinstellung (1414) verwenden. Die Port-Nummer darf 65535 nicht überschreiten.

**Anmerkung:** Um die TCP/IP-Portnummer auszuwählen, die verwendet werden soll, verwendet IBM MQ die erste Portnummer, die in der folgenden Reihenfolge gefunden wird:

1. Die Portnummer, die explizit in der Kanaldefinition oder Befehlszeile angegeben wurde. Mit dieser Zahl kann die Standardportnummer für einen Kanal außer Kraft gesetzt werden.
2. Das Portattribut, das in der TCP-Zeilengruppe der `.ini`-Datei angegeben ist. Mit dieser Zahl kann die Standardportnummer für einen WS-Manager überschrieben werden.
3. Der Standardwert ist 1414. Dies ist die Nummer, die IBM MQ von der Internet Assigned Numbers Authority für eingehende und abgehende Verbindungen zugeordnet ist.

Führen Sie für bestmögliche Leistung den IBM MQ-Listener wie im Abschnitt „Kanäle und Empfangsprogramme als vertrauenswürdige Anwendungen ausführen” auf Seite 263 beschrieben als vertrauenswürdige Anwendung aus. Weitere Informationen zu vertrauenswürdigen Anwendungen finden Sie im Abschnitt Einschränkungen für vertrauenswürdige Anwendungen.

## Verwendung der Option "TCP/IP SO_KEEPALIVE"

Wenn Sie die Option Windows SO_KEEPALIVE verwenden möchten, müssen Sie den folgenden Eintrag in Ihrer Registry hinzufügen:

```
TCP:
KeepAlive=yes
```

Weitere Informationen zur Option SO_KEEPALIVE finden Sie in „Überprüfen, ob das andere Ende des Kanals noch verfügbar ist” auf Seite 247.

Unter Windowssteuert der Registrierungswert `HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` für die Option Windows **KeepAliveTime** das Intervall, das vergeht, bevor die Verbindung überprüft wird. Der Standardwert ist zwei Stunden.

## Verwendung der Option TCP-Listener-Backlog

In TCP werden die Verbindungen nur unvollständig behandelt, wenn zwischen dem Server und dem Client ein Dreiwege-Handshake nicht stattfindet. Diese Verbindungen werden als ausstehende Verbindungsanforderungen bezeichnet. Für diese ausstehenden Verbindungsanforderungen wird ein Maximalwert fest-

gelegt und kann als Rückstand von Anforderungen betrachtet werden, die auf den TCP-Port warten, damit der Listener die Anforderung akzeptiert.

Weitere Informationen und den jeweiligen Wert für Windowsfinden Sie unter „Verwenden der Backlog-Option für den TCP-Listener unter IBM MQ for Multiplatforms” auf Seite 286 .

### Windows *LU 6.2-Verbindung unter Windows definieren*

SNA muss so konfiguriert werden, dass ein LU 6.2-Dialog zwischen den beiden Maschinen aufgebaut werden kann.

Sobald die SNA konfiguriert ist, fahren Sie wie folgt fort.

Informationen hierzu finden Sie in der folgenden Tabelle.

*Tabelle 22. Einstellungen auf dem lokalen Windows-System für eine ferne Warteschlangenmanagerplattform*

| Ferne Plattform | TPNAME | TPPATH |
|---|---|---|
| z/OS oder MVS/ESA ohne CICS | Entsprechendes gilt für die entsprechenden Nebeninformationen zum fernen Warteschlangenmanager. | - |
| z/OS oder MVS/ESA mit CICS | CKRC (Sender) CKSV (Requester) CKRC (Server) | - |
| IBM i | Entsprechendes gilt für den Vergleichswert im Routing-Eintrag auf dem IBM i-System. | - |
| Systeme mit AIX and Linux | Entsprechendes gilt für die entsprechenden Nebeninformationen zum fernen Warteschlangenmanager. | *MQ_INSTALLATION_PATH*/bin/amqcrs6a |
| Windows | Wie im Windows-Befehl 'Listener ausführen' angegeben, oder dem aufrufbaren Transaktionsprogramm, das mit TpSetup unter Windows definiert wurde. | *MQ_INSTALLATION_PATH*\bin\amqcrs6a |

*MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Wenn mehrere WS-Manager auf derselben Maschine vorhanden sind, stellen Sie sicher, dass die TPNames in den Kanaldefinitionen eindeutig sind.

Die neuesten Informationen zum Konfigurieren von AnyNet SNA über TCP/IP finden Sie in der folgenden Onlinedokumentation zu IBM: AnyNet SNA over TCP/IP und SNA-Knotenoperationen.

**Zugehörige Konzepte**

„Sendeseite für LU 6.2 unter Windows” auf Seite 278
Erstellen Sie ein CPI-C-Nebenobjekt (symbolisches Ziel) aus der Verwaltungsanwendung des LU 6.2-Produkts, das Sie verwenden. Geben Sie diesen Namen in das Feld Verbindungsname in der Kanaldefinition ein. Erstellen Sie außerdem einen LU 6.2-Link zu dem Partner.

„Empfangen auf LU 6.2 unter Windows” auf Seite 279
Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet.

### Windows *Sendeseite für LU 6.2 unter Windows*

Erstellen Sie ein CPI-C-Nebenobjekt (symbolisches Ziel) aus der Verwaltungsanwendung des LU 6.2-Produkts, das Sie verwenden. Geben Sie diesen Namen in das Feld Verbindungsname in der Kanaldefinition ein. Erstellen Sie außerdem einen LU 6.2-Link zu dem Partner.

Geben Sie im CPI-C-Nebenobjekt den Partner-LU-Namen auf der empfangenden Maschine, den TP-Namen und den Modusnamen ein. For example:

```
Partner LU Name              OS2ROG2
Partner TP Name              recv
Mode Name                    #INTER
```

Windows *Empfangen auf LU 6.2 unter Windows*

Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet.

Um ein empfangendes Kanalprogramm zu starten, muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Sie starten dieses Listenerprogramm mit dem Befehl RUNMQLSR und geben den TpName an, auf dem empfangsbereit ist. Alternativ können Sie unter SNA Server für Windows auch 'TpStart' verwenden.

## Befehl RUNMQLSR verwenden

Beispiel für den Befehl zum Starten des Listeners:

```
RUNMQLSR -t LU62 -n RECV -m QMNAME
```

Dabei ist RECV der TpName, der auf der anderen Seite (Senden) als "TpName to start on the remote side" angegeben wird. Der im letzten Teil dieses Befehls verwendete Parameter **-m** ist optional und für den Standardwarteschlangenmanager nicht erforderlich.

Es ist möglich, mehr als einen Warteschlangenmanager auf einer Maschine auszuführen. Sie müssen jedem WS-Manager einen anderen TpName zuordnen und anschließend ein Empfangsprogramm für die einzelnen WS-Manager starten. For example:

```
RUNMQLSR -t LU62 -m QM1 -n TpName1
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

Führen Sie für bestmögliche Leistung den IBM MQ-Listener als vertrauenswürdige Anwendung aus, wie im Abschnitt Kanäle und Empfangsprogramme als vertrauenswürdige Anwendungen ausführen beschrieben. Informationen zu vertrauenswürdigen Anwendungen finden Sie im Abschnitt Einschränkungen für vertrauenswürdige Anwendungen .

Sie können alle IBM MQ-Listener, die auf einem inaktiven WS-Manager ausgeführt werden, mit folgendem Befehl stoppen:

```
ENDMQLSR -m QMNAME
```

## Microsoft-SNA-Server unter Windows verwenden

Sie können TpSetup (über das SNA-Server-SDK) verwenden, um ein aufrufbares TP zu definieren, das dann amqcrs6a.exe steuert, oder Sie können verschiedene Registrierungswerte manuell festlegen. Die Parameter, die an amqcrs6a.exe übergeben werden sollen, sind:

```
-m QM -n TpName
```

Hierbei steht *QM* für den Namen des WS-Managers und *TpName* für den TP-Namen. Weitere Informationen hierzu finden Sie im Handbuch *Microsoft SNA Server APPC Programmers Guide* oder im Handbuch *Microsoft SNA Server CPI-C Programmers Guide*.

Wenn Sie keinen Warteschlangenmanagernamen angeben, wird der Standardwarteschlangenmanager angenommen.

## Windows *NetBIOS-Verbindung in Windows definieren*

Eine NetBIOS-Verbindung gilt nur für einen Client und Server, auf dem Windows ausgeführt wird. IBM MQ verwendet drei Typen von NetBIOS-Ressourcen, wenn eine NetBIOS-Verbindung zu einem anderen IBM MQ-Produkt hergestellt wird: Sitzungen, Befehle und Namen. Jede dieser Ressourcen hat einen Grenzwert, der entweder standardmäßig oder nach Auswahl während der Installation von NetBIOS festgelegt wird.

Jeder aktive Kanal, unabhängig vom Typ, verwendet eine NetBIOS-Sitzung und einen NetBIOS-Befehl. Die NetBIOS-Implementierung von IBM ermöglicht es mehreren Prozessen, denselben lokalen NetBIOS-Namen zu verwenden. Daher muss nur ein NetBIOS-Name für die Verwendung durch IBM MQ verfügbar sein. Andere Anbieterimplementierungen, z. B. Novell's NetBIOS-Emulation, erfordern einen anderen lokalen Namen pro Prozess. Überprüfen Sie Ihre Anforderungen in der Dokumentation für das NetBIOS-Produkt, das Sie verwenden.

Stellen Sie in allen Fällen sicher, dass bereits genügend Ressourcen für die einzelnen Typen verfügbar sind, oder erhöhen Sie die in der Konfiguration angegebenen Maximalwerte. Alle Änderungen an den Werten erfordern einen Neustart des Systems.

Während des Systemstarts zeigt der NetBIOS-Einheitentreiber die Anzahl der Sitzungen, Befehle und Namen an, die für Anwendungen zur Verfügung stehen. Diese Ressourcen stehen für alle NetBIOS-basierten Anwendungen zur Verfügung, die auf demselben System ausgeführt werden. Daher ist es möglich, dass andere Anwendungen diese Ressourcen in Anspruch nehmen, bevor IBM MQ sie erwerben muss. Ihr LAN-Netzadministrator sollte dies für Sie klären können.

**Zugehörige Konzepte**

„Lokalen NetBIOS-Namen für IBM MQ definieren” auf Seite 280
Der lokale NetBIOS-Name, der von IBM MQ-Kanalprozessen verwendet wird, kann auf drei Arten angegeben werden.

„WS-Manager-NetBIOS-Sitzung, -Befehl und -Namensbegrenzungen einrichten” auf Seite 281
Die Grenzwerte des Warteschlangenmanagers für NetBIOS-Sitzungen, -Befehle und -Namen können auf zwei Arten angegeben werden.

„LAN-Adapternummer erstellen” auf Seite 281
Damit Kanäle erfolgreich über NetBIOS ausgeführt werden können, muss die Adapterunterstützung an jedem Ende kompatibel sein. IBM MQ ermöglicht, zu steuern, welche LAN-Adapternummer (LANA) verwendet wird, hierzu wird der Wert "AdapterNum" in der NETBIOS-Zeilengruppe der qm.ini-Datei und der Parameter **-a** im Befehl "runmqlsr" angegeben.

„NetBIOS-Verbindung initialisieren” auf Seite 282
Definieren der Schritte, die zum Einleiten einer Verbindung erforderlich sind.

„Ziellistener für die NetBIOS-Verbindung definieren” auf Seite 282
Definieren der Schritte, die am empfangenden Ende der NetBIOS-Verbindung ausgeführt werden sollen.

## Windows *Lokalen NetBIOS-Namen für IBM MQ definieren*

Der lokale NetBIOS-Name, der von IBM MQ-Kanalprozessen verwendet wird, kann auf drei Arten angegeben werden.

In der Reihenfolge der Vorrangstellung sind die drei folgenden Möglichkeiten:

1. Der Wert, der im Parameter **-l** des Befehls **runmqlsr** angegeben ist. Beispiel:

   ```
   runmqlsr -t netbios -l my_station
   ```

2. Die Umgebungsvariable **MQNAME** mit einem Wert, der vom Befehl festgelegt wird:

   ```
   SET MQNAME= my_station
   ```

   For example:

```
SET MQNAME=CLIENT1
```

Sie können den Wert **MQNAME** für jeden Prozess festlegen. Alternativ können Sie ihn auf einer Systemebene in der Windows-Registry festlegen.

Wenn Sie eine NetBIOS -Implementierung verwenden, die eindeutige Namen erfordert, müssen Sie in jedem Fenster, in dem ein IBM MQ -Prozess gestartet wird, einen **SET MQNAME** -Befehl absetzen. Der Wert **MQNAME** ist beliebig, muss aber für jeden Prozess eindeutig sein.

3. Die NETBIOS-Zeilengruppe in der Konfigurationsdatei des Warteschlangenmanagers `qm.ini`. For example:

```
NETBIOS:

LocalName= my_station
```

**Anmerkung:**

1. Aufgrund der Unterschiede bei der Implementierung der unterstützten NetBIOS-Produkte wird empfohlen, die einzelnen NetBIOS-Namen im Netz eindeutig zu machen. Wenn dies nicht der Fall ist, kann es zu unvorhersehbaren Ergebnissen kommen. Wenn Sie Probleme beim Aufbau eines NetBIOS-Kanals haben und im Fehlerprotokoll des Warteschlangenmanagers Fehlernachrichten enthalten sind, die einen NetBIOS-Rückkehrcode von X'15 ' aufweisen, überprüfen Sie die Verwendung von NetBIOS-Namen.
2. Unter Windows können Sie Ihren Maschinennamen nicht als NetBIOS-Namen verwenden, da Windows ihn bereits verwendet.
3. Für die Initialisierung des Senderkanals ist es erforderlich, dass ein NetBIOS-Name entweder über die Umgebungsvariable MQNAME oder über den lokalen Namen in der Datei qm.ini angegeben wird.

Windows *WS-Manager-NetBIOS-Sitzung, -Befehl und -Namensbegrenzungen einrichten*

Die Grenzwerte des Warteschlangenmanagers für NetBIOS-Sitzungen, -Befehle und -Namen können auf zwei Arten angegeben werden.

In der Reihenfolge der Vorrangstellung sind folgende Möglichkeiten:

1. Die im Befehl RUNMQLSR angegebenen Werte:

```
-s Sessions
-e Names
-o Commands
```

Wenn der Operand -m nicht im Befehl angegeben ist, gelten die Werte nur für den Standardwarteschlangenmanager.

2. Die NETBIOS-Zeilengruppe in der WS-Managerkonfigurationsdatei qm.ini. For example:

```
NETBIOS:

NumSess= Qmgr_max_sess
NumCmds= Qmgr_max_cmds
NumNames= Qmgr_max_names
```

Windows *LAN-Adapternummer erstellen*

Damit Kanäle erfolgreich über NetBIOS ausgeführt werden können, muss die Adapterunterstützung an jedem Ende kompatibel sein. IBM MQ ermöglicht, zu steuern, welche LAN-Adapternummer (LANA) verwendet wird, hierzu wird der Wert "AdapterNum" in der NETBIOS-Zeilengruppe der qm.ini-Datei und der Parameter **-a** im Befehl "runmqlsr" angegeben.

Die Standard-LAN-Adapternummer, die von IBM MQ für NetBIOS-Verbindungen verwendet wird, ist 0. Stellen Sie sicher, dass die Nummer auf Ihrem System wie folgt verwendet wird:

Unter Windows ist es nicht möglich, die LAN-Adapternummer direkt über das Betriebssystem abzurufen. Stattdessen verwenden Sie das Befehlszeilendienstprogramm LANACFG.EXE, das über Microsoft verfügbar ist. Die Ausgabe des Tools zeigt die Nummern des virtuellen LAN-Adapters und die zugehörigen effektiven Bindungen an. Weitere Informationen zu LAN-Adapternummern finden Sie im Microsoft Knowledge Base-Artikel 138037 *HOWTO: Use LANA Numbers in a 32-bit Environment*.

Geben Sie den korrekten Wert in der Zeilengruppe NETBIOS der WS-Manager-Konfigurationsdatei qm.ini an:

```
NETBIOS:
AdapterNum= n
```

Dabei steht n für die korrekte LAN-Adapternummer für dieses System.

Windows *NetBIOS-Verbindung initialisieren*

Definieren der Schritte, die zum Einleiten einer Verbindung erforderlich sind.

Führen Sie die folgenden Schritte aus, um die Verbindung zu starten:

1. Definieren Sie den NetBIOS-Stationsnamen mit dem Wert MQNAME oder LocalName.
2. Überprüfen Sie, ob die LAN-Adapternummer auf Ihrem System verwendet wird, und geben Sie die korrekte Datei mit dem AdapterNum an.
3. Geben Sie im Feld ConnectionName der Kanaldefinition den NetBIOS-Namen an, der vom Ziel-Listener-Programm verwendet wird. Unter Windows müssen NetBIOS-Kanäle als Threads ausgeführt werden. Geben Sie dazu MCATYPE (THREAD) in der Kanaldefinition an.

   ```
   DEFINE CHANNEL (chname) CHLTYPE(SDR) +
   TRPTYPE(NETBIOS) +
   CONNAME(your_station) +
   XMITQ(xmitq) +
   MCATYPE(THREAD) +
   REPLACE
   ```

Windows *Ziellistener für die NetBIOS-Verbindung definieren*

Definieren der Schritte, die am empfangenden Ende der NetBIOS-Verbindung ausgeführt werden sollen.

Führen Sie auf der Empfangsseite die folgenden Schritte aus:

1. Definieren Sie den NetBIOS-Stationsnamen mit dem Wert MQNAME oder LocalName.
2. Überprüfen Sie, ob die LAN-Adapternummer auf Ihrem System verwendet wird, und geben Sie die korrekte Datei mit dem AdapterNum an.
3. Definieren Sie den Empfängerkanal:

   ```
   DEFINE CHANNEL (chname) CHLTYPE(RCVR) +
   TRPTYPE(NETBIOS) +
   REPLACE
   ```

4. Starten Sie das IBM MQ-Listenerprogramm, um die Station einzurichten und eine Verbindung zu dieser Station herzustellen. For example:

   ```
   RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
   ```

   Mit diesem Befehl wird `your_station` als NetBIOS-Station eingerichtet, die darauf wartet, kontaktiert zu werden. Der NetBIOS-Stationsname muss im gesamten NetBIOS-Netz eindeutig sein.

Führen Sie für bestmögliche Leistung den IBM MQ-Listener wie im Abschnitt „Kanäle und Empfangsprogramme als vertrauenswürdige Anwendungen ausführen” auf Seite 263 beschrieben als vertrauenswürdige Anwendung aus. Informationen zu vertrauenswürdigen Anwendungen finden Sie im Abschnitt Einschränkungen für vertrauenswürdige Anwendungen .

Sie können alle IBM MQ-Listener, die auf einem inaktiven WS-Manager ausgeführt werden, mit folgendem Befehl stoppen:

```
ENDMQLSR [-m QMNAME]
```

Wenn Sie keinen Warteschlangenmanagernamen angeben, wird der Standardwarteschlangenmanager angenommen.

## Linux AIX Kommunikation unter AIX and Linux einrichten

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dies gelingt, muss die Verbindung definiert und verfügbar sein. In diesem Abschnitt wird erläutert, wie Sie dies tun, indem Sie die Kommunikationsformen verwenden, die für IBM MQ for UNIX or Linux-Systeme verfügbar sind.

### Vorbereitende Schritte

Es kann hilfreich sein, auf die folgenden Abschnitte zu verweisen:

- AIX Beispielkonfiguration - IBM MQ for AIX
- Linux Beispielkonfiguration - IBM MQ for Linux

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

### Informationen zu diesem Vorgang

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Um erfolgreich zu sein, ist es erforderlich, dass die Verbindung definiert und verfügbar ist. In diesem Abschnitt wird beschrieben, wie dies ausgeführt wird.

Wenn Sie die Kommunikation für IBM MQ unter AIX and Linux einrichten, können Sie zwischen den folgenden Arten von Kommunikation wählen:

- TCP/IP
- LU 6.2

Jede Kanaldefinition muss nur ein Attribut des Übertragungsprotokolls (Transporttyp) angeben. Ein oder mehrere Protokolle können von einem WS-Manager verwendet werden.

Für IBM MQ MQI clients kann es sinnvoll sein, alternative Kanäle mit unterschiedlichen Übertragungsprotokollen zu verwenden. Siehe hierzu IBM MQ MQI clients.

### Vorgehensweise

Informationen zum Einrichten der Kommunikation für Ihr AIX- oder Linux-System finden Sie im Unterabschnitt für den ausgewählten Kommunikationstyp:

- „TCP-Verbindung unter AIX and Linux definieren” auf Seite 284
- „LU 6.2-Verbindung unter AIX and Linux definieren” auf Seite 288
- MQ Adv. MQ Adv. VUE „Aspera gateway -Verbindung auf Linux -oder Windows -Plattformen definieren” auf Seite 913

**Zugehörige Tasks**

„Kanäle in AIX, Linux, and Windows überwachen und steuern” auf Seite 264
Für DQM müssen Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Sie können Kanäle mit Befehlen, Programmen, IBM MQ Explorer, Dateien für die Kanaldefinitionen und einem Speicherbereich für Synchronisationsinformationen steuern.

„Verbindungen zwischen Client und Server konfigurieren” auf Seite 15
Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

„Kommunikation unter Windows einrichten” auf Seite 275
Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dies gelingt, muss die Verbindung definiert und verfügbar sein. In diesem Abschnitt wird erläutert, wie Sie dies tun, indem Sie die Kommunikationsformen verwenden, die für IBM MQ for Windows-Systeme verfügbar sind.

**Zugehörige Verweise**

„Zu verwendende Übertragungsart” auf Seite 16
Unterschiedliche Plattformen unterstützen unterschiedliche Kommunikationsprotokolle. Ihre Auswahl des Übertragungsprotokolls hängt von Ihrer Kombination von IBM MQ MQI client- und Serverplattformen ab.

## Linux AIX *TCP-Verbindung unter AIX and Linux definieren*

Die Kanaldefinition auf der sendenden Seite gibt die Adresse des Ziels an. Der Listener-oder inet-Dämon ist für die Verbindung an der empfangenden Seite konfiguriert.

### Vorbereitungen

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

### Sendende Beendigung

Geben Sie den Hostnamen oder die TCP-Adresse der Zielmaschine in das Feld Verbindungsname der Kanaldefinition an. Der Port, zu dem die Verbindung hergestellt werden soll, standardmäßig 1414. Die Portnummer 1414 ist IBM MQ von der Internet Assigned Numbers Authority (IANA) zugewiesen.

Wenn Sie eine andere Portnummer als die Standardportnummer verwenden möchten, ändern Sie das Feld für den Verbindungsnamen wie folgt:

```
Connection Name REMHOST(1822)
```

Hierbei steht REMHOST für den Hostnamen der fernen Maschine und 1822 für die erforderliche Portnummer. (Dies muss der Port sein, an dem der Listener empfangsbereit ist.)

Alternativ können Sie die Portnummer ändern, indem Sie sie in der WS-Managerkonfigurationsdatei (qm.ini) angeben:

```
TCP:
Port=1822
```

Weitere Informationen zu den Werten, die Sie mit qm.ini festlegen, finden Sie unter Zeilengruppen für Konfigurationsdateien für verteilte Steuerung von Warteschlangen .

## Empfang auf TCP

Sie können entweder den TCP/IP-Listener verwenden, bei dem es sich um den inet-Dämon (inetd) handelt, oder den IBM MQ-Listener.

Einige Linux-Verteilungen verwenden jetzt anstelle des inet-Dämons den erweiterten inet-Dämon (xinetd). Weitere Informationen zur Verwendung des erweiterten inet-Dämons auf einem Linux -System finden Sie in Schritt 2 unter Beispiel: Plattformübergreifende Kommunikation für IBM MQ unter Linuxeinrichten.

**Zugehörige Konzepte**

„TCP/IP-Listener unter AIX and Linux verwenden” auf Seite 285
Um Kanäle in AIX and Linux zu starten, müssen die `/etc/services`-Datei und die `inetd.conf`-Datei bearbeitet werden.

„Verwenden der Backlog-Option für den TCP-Listener unter IBM MQ for Multiplatforms” auf Seite 286
In TCP werden die Verbindungen nur unvollständig behandelt, wenn zwischen dem Server und dem Client ein Dreiwege-Handshake nicht stattfindet. Diese Verbindungen werden als ausstehende Verbindungsanforderungen bezeichnet. Für diese ausstehenden Verbindungsanforderungen wird ein Maximalwert festgelegt und kann als Rückstand von Anforderungen betrachtet werden, die auf den TCP-Port warten, damit der Listener die Anforderung akzeptiert.

„IBM MQ-Listener verwenden” auf Seite 287
Verwenden Sie den Befehl `runmqlsr` , um das mit IBM MQbereitgestellte Empfangsprogramm auszuführen, das neue Kanäle als Threads startet.

„Verwendung der Option "TCP/IP SO_KEEPALIVE"” auf Seite 288
Auf einigen AIX and Linux-Systemen können Sie festlegen, wie lange TCP warten soll, bevor überprüft wird, ob die Verbindung noch verfügbar ist, und wie oft eine Verbindungswiederholung erfolgen soll, wenn die erste Überprüfung nicht erfolgreich ist. Dies ist entweder ein optimierbarer Kernelparameter oder kann in der Befehlszeile eingegeben werden.

Linux AIX *TCP/IP-Listener unter AIX and Linux verwenden*

Um Kanäle in AIX and Linux zu starten, müssen die `/etc/services`-Datei und die `inetd.conf`-Datei bearbeitet werden.

Befolgen Sie diese Anweisungen:

1. Bearbeiten Sie die `/etc/services`-Datei:

   **Anmerkung:** Um die `/etc/services`-Datei zu bearbeiten, müssen Sie als Superuser oder Root angemeldet sein. Sie können diese Änderung ändern, aber sie muss mit der Portnummer übereinstimmen, die auf der Senderseite angegeben wurde.

   Fügen Sie die folgende Zeile zur Datei hinzu:

   ```
   MQSeries 1414/tcp
   ```

   Dabei ist 1414 die Portnummer, die von IBM MQ benötigt wird. Die Port-Nummer darf 65535 nicht überschreiten.
2. Fügen Sie eine Zeile in der Datei `inetd.conf` hinzu, um das Programm amqcrsta aufzurufen, wobei *MQ_INSTALLATION_PATH* für das übergeordnete Verzeichnis steht, in dem IBM MQ installiert ist:

   ```
   MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
   [-m Queue_Man_Name]
   ```

Die Aktualisierungen sind aktiv, nachdem 'inetd' die Konfigurationsdateien erneut gelesen hat. Geben Sie dazu die folgenden Befehle von der Rootbenutzer-ID aus:

- AIX Unter AIX:

```
refresh -s inetd
```

- Linux Auf Linux-Systemen:

```
kill -1 process_number
```

Wenn das von inetd gestartete Empfangsprogrammprogramm die Ländereinstellung von inetd übernimmt, ist es möglich, dass der MQMDE nicht berücksichtigt wird (zusammengeführt) und als Nachrichtendaten in die Warteschlange gestellt wird. Um sicherzustellen, dass der MQMDE berücksichtigt wird, müssen Sie die Ländereinstellung korrekt festlegen. Die von inetd festgelegte Ländereinstellung stimmt möglicherweise nicht mit der Ländereinstellung überein, die für andere Ländereinstellungen ausgewählt wurde, die von IBM MQ-Prozessen verwendet werden. So legen Sie die Ländereinstellung fest:

1. Erstellen Sie ein Shell-Script, das die Umgebungsvariablen LANG, LC_COLLATE, LC_CTYPE, LC_MONETARY, LC_NUMERIC, LC_TIME und LC_MESSAGES auf die Ländereinstellung setzt, die für andere IBM MQ-Prozesse verwendet wird.
2. Rufen Sie in demselben Shell-Script das Empfangsprogrammprogramm auf.
3. Ändern Sie die `inetd.conf`-Datei so, dass Sie Ihr Shell-Script anstelle des Listenerprogramms aufrufen.

Es ist möglich, mehr als einen Warteschlangenmanager auf dem Server zu verwenden. Sie müssen jeder der beiden Dateien eine Zeile für jeden der WS-Manager hinzufügen. For example:

```
MQSeries1   1414/tcp
MQSeries2   1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

Dabei steht *MQ_INSTALLATION_PATH* für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

So lässt sich vermeiden, dass Fehlermeldungen ausgegeben werden, wenn eine Beschränkung für die Anzahl der ausstehenden Verbindungsanforderungen gilt, die in der Warteschlange eines einzelnen TCP-Ports stehen können. Informationen zur Anzahl der ausstehenden Verbindungsanforderungen finden Sie in „Verwenden der Backlog-Option für den TCP-Listener unter IBM MQ for Multiplatforms” auf Seite 286.

Multi *Verwenden der Backlog-Option für den TCP-Listener unter IBM MQ for Multiplatforms*

In TCP werden die Verbindungen nur unvollständig behandelt, wenn zwischen dem Server und dem Client ein Dreiwege-Handshake nicht stattfindet. Diese Verbindungen werden als ausstehende Verbindungsanforderungen bezeichnet. Für diese ausstehenden Verbindungsanforderungen wird ein Maximalwert festgelegt und kann als Rückstand von Anforderungen betrachtet werden, die auf den TCP-Port warten, damit der Listener die Anforderung akzeptiert.

Die Standardwerte für das Listener-Rückstandsprotokoll werden in Tabelle 23 auf Seite 286 angezeigt.

*Tabelle 23. Maximale Anzahl ausstehender Verbindungsanforderungen, die an einem TCP/IP-Port in die Warteschlange*

| **Serverplattform** | **Maximale Verbindungsanforderungen** |
|---|---|
| AIX AIX | 100 |
| Linux Linux | 100 |
| IBM i IBM i | 255 |

| *Tabelle 23. Maximale Anzahl ausstehender Verbindungsanforderungen, die an einem TCP/IP-Port in die Warteschlange (Forts.)* | |
|---|---|
| **Serverplattform** | **Maximale Verbindungsanforderungen** |
| Windows Windows -Server | 100 |

Wenn der Rückstand die in Tabelle 23 auf Seite 286 angegebenen Werte erreicht, wird die TCP/IP-Verbindung abgelehnt und der Kanal kann nicht gestartet werden.

Bei MCA-Kanälen führt dies dazu, dass der Kanal in einen RETRY-Status eingeht und die Verbindung zu einem späteren Zeitpunkt erneut versucht.

Um diesen Fehler zu vermeiden, können Sie jedoch einen Eintrag in der `qm.ini`-Datei hinzufügen:

```
TCP:
ListenerBacklog = n
```

Dies überschreibt die standardmäßige maximale Anzahl ausstehender Anforderungen (siehe Tabelle 23 auf Seite 286) für den TCP/IP-Listener.

**Anmerkung:** Einige Betriebssysteme unterstützen einen größeren Wert als der Standardwert. Falls erforderlich, kann dieser Wert verwendet werden, um das Erreichen des Verbindungsgrenzwerts zu vermeiden.

So führen Sie den Listener mit aktivierter Option `backlog` aus:

- Verwenden Sie den Befehl `runmqlsr -b` oder
- Verwenden Sie den MQSC-Befehl **`DEFINE LISTENER`** mit dem Attribut BACKLOG, das auf den erforderlichen Wert gesetzt ist.

Weitere Informationen zum Befehl **`runmqlsr`** finden Sie im Abschnitt runmqls. Weitere Informationen zum Befehl DEFINE LISTENER finden Sie in DEFINE LISTENER .

**Zugehörige Konzepte**

„Verwenden der Backlog-Option für den TCP-Listener unter z/OS” auf Seite 1077
Beim Empfang über TCP/IP wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Diese ausstehenden Anforderungen können als *Rückstand* von Anforderungen betrachtet werden, die auf den TCP/IP-Port warten, bis der Listener die Anforderung akzeptiert hat.

Linux AIX *IBM MQ-Listener verwenden*

Verwenden Sie den Befehl `runmqlsr` , um das mit IBM MQbereitgestellte Empfangsprogramm auszuführen, das neue Kanäle als Threads startet.

For example:

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

Die eckigen Klammern geben optionale Parameter an; QMNAME ist für den Standard-WS-Manager nicht erforderlich, und die Portnummer ist nicht erforderlich, wenn Sie die Standardeinstellung (1414) verwenden. Die Port-Nummer darf 65535 nicht überschreiten.

Führen Sie für bestmögliche Leistung den IBM MQ-Listener wie im Abschnitt „Kanäle und Empfangsprogramme als vertrauenswürdige Anwendungen ausführen” auf Seite 263 beschrieben als vertrauenswürdige Anwendung aus. Informationen zu vertrauenswürdigen Anwendungen finden Sie im Abschnitt Einschränkungen für vertrauenswürdige Anwendungen .

Sie können alle IBM MQ-Listener, die auf einem inaktiven WS-Manager ausgeführt werden, mit folgendem Befehl stoppen:

```
endmqlsr [-m QMNAME]
```

Wenn Sie keinen Warteschlangenmanagernamen angeben, wird der Standardwarteschlangenmanager angenommen.

Linux AIX *Verwendung der Option "TCP/IP SO_KEEPALIVE"*

Auf einigen AIX and Linux-Systemen können Sie festlegen, wie lange TCP warten soll, bevor überprüft wird, ob die Verbindung noch verfügbar ist, und wie oft eine Verbindungswiederholung erfolgen soll, wenn die erste Überprüfung nicht erfolgreich ist. Dies ist entweder ein optimierbarer Kernelparameter oder kann in der Befehlszeile eingegeben werden.

Wenn Sie die Option SO_KEEPALIVE verwenden möchten (weitere Informationen hierzu finden Sie im Abschnitt „Überprüfen, ob das andere Ende des Kanals noch verfügbar ist” auf Seite 247), müssen Sie den folgenden Eintrag in Ihrer Warteschlangenmanagerkonfigurationsdatei (qm.ini) hinzufügen:

```
TCP:
KeepAlive=yes
```

Weitere Informationen finden Sie in der Dokumentation zu Ihrem AIX- oder Linux-System.

## Linux AIX *LU 6.2-Verbindung unter AIX and Linux definieren*

SNA muss so konfiguriert werden, dass ein LU 6.2-Dialog zwischen den beiden Maschinen aufgebaut werden kann.

Die neuesten Informationen zum Konfigurieren von SNA über TCP/IP finden Sie in der folgenden Onlinedokumentation zu IBM: Communications Server.

SNA muss so konfiguriert werden, dass ein LU 6.2-Dialog zwischen den beiden Systemen aufgebaut werden kann.

Informationen hierzu finden Sie im Handbuch *Multiplatform APPC Configuration Guide* und in der folgenden Tabelle.

*Tabelle 24. Einstellungen auf dem lokalen AIX- oder Linux-System für einen fernen Plattformmanager*

| **Ferne Plattform** | **TPNAME** | **TPPATH** |
|---|---|---|
| z/OS ohne CICS | Dasselbe gilt für den entsprechenden TPName-Wert in den Nebeninformationen zum fernen Warteschlangenmanager. | - |
| z/OS mit CICS | CKRC (Sender) CKSV (Requester) CKRC (Server) | - |
| IBM i | Entsprechendes gilt für den Vergleichswert im Routing-Eintrag auf dem IBM i-System. | - |
| Systeme mit AIX and Linux | Dasselbe gilt für den entsprechenden TPName-Wert in den Nebeninformationen zum fernen Warteschlangenmanager. | *MQ_INSTALLATION_PATH*/bin/amqcrs6a |
| Windows | Wie im Windows-Befehl 'Listener ausführen' angegeben, oder dem aufrufbaren Transaktionsprogramm, das mit TpSetup unter Windows definiert wurde. | *MQ_INSTALLATION_PATH*\bin\amqcrs6a |

*MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Wenn mehrere WS-Manager auf derselben Maschine vorhanden sind, stellen Sie sicher, dass die TPNames in den Kanaldefinitionen eindeutig sind.

**Zugehörige Konzepte**

„Sendeseite für LU 6.2 unter AIX and Linux” auf Seite 289
Erstellen Sie auf AIX and Linux-Systemen ein CPI-C-Nebenobjekt (symbolisches Ziel), und geben Sie diesen Namen in das Feld Verbindungsname in der Kanaldefinition ein. Erstellen Sie außerdem einen LU 6.2-Link zu dem Partner.

„Empfangen auf LU 6.2 unter AIX and Linux” auf Seite 289
Erstellen Sie auf AIX and Linux-Systemen einen empfangsbereiten Anschluss auf der Empfangsseite, ein logisches LU 6.2-Verbindungsprofil und ein TPN-Profil.

Linux AIX *Sendeseite für LU 6.2 unter AIX and Linux*

Erstellen Sie auf AIX and Linux-Systemen ein CPI-C-Nebenobjekt (symbolisches Ziel), und geben Sie diesen Namen in das Feld Verbindungsname in der Kanaldefinition ein. Erstellen Sie außerdem einen LU 6.2-Link zu dem Partner.

Geben Sie im CPI-C-Nebenobjekt den Namen der Partner-LU auf der empfangenden Maschine, den Namen des Transaktionsprogramms und den Modusnamen ein. For example:

```
Partner LU Name              REMHOST
Remote TP Name               recv
Service Transaction Program  no
Mode Name                    #INTER
```

SECURITY PROGRAM wird verwendet, sofern Unterstützung durch CPI-C besteht, wenn IBM MQ versucht, eine SNA-Sitzung aufzubauen.

Linux AIX *Empfangen auf LU 6.2 unter AIX and Linux*

Erstellen Sie auf AIX and Linux-Systemen einen empfangsbereiten Anschluss auf der Empfangsseite, ein logisches LU 6.2-Verbindungsprofil und ein TPN-Profil.

Geben Sie im TPN-Profil den vollständigen Pfad zu der ausführbaren Datei und den Namen des Transaktionsprogramms ein:

```
Full path to TPN executable    MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name      recv
User ID                       0
```

*MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Geben Sie auf Systemen, auf denen Sie die Benutzer-ID festlegen können, einen Benutzer an, der Mitglied der Gruppe "mqm" ist.

AIX Geben Sie unter AIX die Umgebungsvariablen APPCTPN (Transaktionsname) und APPCLLU (lokaler LU-Name) an (hierzu können Sie die Konfigurationsanzeigen für das aufgerufene Transaktionsprogramm verwenden).

Möglicherweise müssen Sie einen anderen WS-Manager als den Standardwarteschlangenmanager verwenden. Ist dies der Fall, definieren Sie eine Befehlsdatei, die Folgendes aufruft:

```
amqcrs6a -m Queue_Man_Name
```

Rufen Sie dann die Befehlsdatei auf.

## IBM i Kanäle in IBM i überwachen und steuern

Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Jeder WS-Manager verfügt über ein DQM-Programm zur Steuerung von Verbindungen zu kompatiblen fernen Warteschlangenmanagern.

## Informationen zu diesem Vorgang

Die folgende Liste enthält eine kurze Beschreibung der Komponenten der Kanalsteuerfunktion:

- Kanaldefinitionen werden als WS-Manager-Objekte gehalten.
- Bei den Kanalbefehlen handelt es sich um eine Untergruppe der IBM MQ for IBM i-Befehlsgruppe.

  Verwenden Sie den Befehl "GO CMDMQM", um die vollständige Gruppe der IBM MQ for IBM i-Befehle anzuzeigen.
- Sie verwenden Kanaldefinitionsanzeigen oder Befehle für:
  - Kanaldefinitionen erstellen, kopieren, anzeigen, ändern und löschen
  - Kanäle starten und stoppen, Pingsignal absetzen, Kanalfolgenummern zurücksetzen und unbestätigte Nachrichten auflösen, wenn Links nicht erneut aufgebaut werden können
  - Statusinformationen zu Kanälen anzeigen
- Kanäle können auch mit Hilfe von MQSC verwaltet werden.
- Kanäle können auch mit IBM MQ Explorer verwaltet werden.
- Die Folgenummern und die Kennungen der *logischen Arbeitseinheit (LUW)* werden in der Synchronisationsdatei gespeichert und werden für die Kanalsynchronisation verwendet.

Sie können die Befehle und Anzeigen verwenden, um Nachrichtenkanäle und zugehörige Objekte zu definieren sowie Nachrichtenkanäle zu überwachen und zu steuern. Wenn Sie die Taste F4 (Bedienerführung) verwenden, können Sie den entsprechenden Warteschlangenmanager angeben. Wenn Sie die Eingabeaufforderung nicht verwenden, wird der Standardwarteschlangenmanager angenommen. Mit F4 = Bedienerführung wird eine zusätzliche Anzeige aufgerufen, in der der Name des relevanten Warteschlangenmanagers und manchmal auch andere Daten eingegeben werden können.

Zu den Objekten, die Sie mit den Anzeigen definieren müssen, gehören:

- Übertragungswarteschlangen
- Definitionen ferner Warteschlangen
- WS-Manager-Aliasdefinitionen
- Aliasnamendefinitionen für Antwortwarteschlange
- Antwort-in lokale Warteschlangen
- Nachrichtenkanaldefinitionen

Weitere Informationen zu den Konzepten, die an der Verwendung dieser Objekte beteiligt sind, finden Sie in „Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206.

Kanäle müssen vollständig definiert sein, und ihre zugeordneten Objekte müssen vorhanden und verfügbar sein, bevor ein Kanal gestartet werden kann.

Darüber hinaus muss die jeweilige Kommunikationsverbindung für jeden Kanal definiert und verfügbar sein, bevor ein Kanal ausgeführt werden kann. Eine Beschreibung der Definition von LU 6.2-und TCP/IP-Verbindungen finden Sie in der jeweiligen Kommunikationsleitfaden für Ihre Installation.

## Prozedur

- Weitere Informationen zum Erstellen und Arbeiten mit Objekten finden Sie unter:
  - „Objekte unter IBM i erstellen” auf Seite 291
  - „Kanal unter IBM i erstellen” auf Seite 291
  - „Kanal unter IBM i starten” auf Seite 293
  - „Kanal unter IBM i auswählen” auf Seite 294
  - „Durchsuchen eines Kanals unter IBM i” auf Seite 294
  - „Kanal unter IBM i umbenennen” auf Seite 296
  - „Mit Kanalstatus unter IBM i arbeiten” auf Seite 296

– „Auswahlmöglichkeiten für Arbeit mit Kanal unter IBM i” auf Seite 297

**Zugehörige Konzepte**

„Kommunikation für IBM i konfigurieren” auf Seite 303
Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit sie erfolgreich ist, ist es erforderlich, dass die Verbindung definiert und verfügbar ist.

**Zugehörige Tasks**

„Verbindungen zwischen Client und Server konfigurieren” auf Seite 15
Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

**Zugehörige Verweise**

Beispielkonfiguration - IBM MQ for IBM i
Beispiel für Nachrichtenkanalplanung für IBM MQ for IBM i
CL-Befehle von IBM MQ for IBM i

## IBM i Objekte unter IBM i erstellen

Sie können den Befehl CRTMQMQ verwenden, um die Warteschlangen-und Aliasobjekte zu erstellen.

Sie können die Warteschlangen-und Aliasobjekte wie z. B. Übertragungswarteschlangen, ferne Warteschlangendefinitionen, WS-Manager-Aliasnamendefinitionen, Antwortwarteschlangenaliasdefinitionen und Antworten-in lokale Warteschlangen erstellen.

Eine Liste der Standardobjekte finden Sie im Abschnitt System- und Standardobjekte.

## IBM i Kanal unter IBM i erstellen

Sie können einen Kanal über die Anzeige "Kanal erstellen" oder über den Befehl CRTMQMCHL in der Befehlszeile erstellen.

So erstellen Sie einen Kanal:

1. Verwenden Sie F6 in der Anzeige "Mit MQM-Kanälen arbeiten" (WRKMQMCHL).

   Alternativ können Sie den Befehl CRTMQMCHL über die Befehlszeile verwenden.

   In jedem Fall wird die Anzeige 'Kanal erstellen' angezeigt. Typ:

   - Der Name des Kanals in dem angegebenen Feld.
   - Der Kanaltyp für dieses Ende des Links
2. Drücken Sie die Eingabetaste.

**Anmerkung:** Sie müssen alle Kanäle in Ihrem Netzwerk eindeutig benennen. Wie in Netzdiagramm mit allen Kanälen gezeigt, ist dies eine gute Möglichkeit, die Namen der Quellen-und Zielwarteschlangenmanager in den Kanalnamen zu verwenden.

Ihre Eingaben werden validiert, und die Fehler werden sofort gemeldet. Beheben Sie alle Fehler und fahren Sie fort.

Sie werden mit der entsprechenden Anzeige für die Kanaleinstellungen für die Art des ausgewählten Kanals angezeigt. Füllen Sie die Felder mit den Informationen aus, die Sie zuvor zusammengestellt haben. Drücken Sie die Eingabetaste, um den Kanal zu erstellen.

Sie erhalten Hilfe bei der Entscheidung über den Inhalt der verschiedenen Felder in den Beschreibungen der Kanaldefinitionsanzeigen in den Hilfetexten und in Kanalattribute .

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . > CHANNAME_____________
Channel type . . . . . > *SDR__    *RCVR, *SDR, *SVR, *RQSTR...
Message Queue Manager name     *DFT_____________________________
_____
Replace . . . . . .  *NO_     *NO, *YES
Transport type . . . . . .  *TCP____   *LU62, *TCP, *SYSDFTCHL
Text 'description' . . . . . > 'Example Channel Definition'_____________
_____________________________
Connection name . . . . . .  *SYSDFTCHL_____________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
_____________________________________________
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 25. Kanal erstellen (1)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Transmission queue . . . . . .  'TRANSMISSION_QUEUE_NAME'_______________
_________________
Message channel agent . . . . .  *NONE______  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
Message channel agent user ID .  *SYSDFTCHL__ Character value...
Coded Character Set Identifier   *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Batch size . . . . . .  50_________  1-9999, *SYSDFTCHL
Disconnect interval . . . . . .  6000_______  1-999999, *SYSDFTCHL
Short retry interval . . . . . .  60_________  0-999999999, *SYSDFTCHL
Short retry count . . . . . .  10_________  0-999999999, *SYSDFTCHL
Long retry interval . . . . . .  1200_______  0-999999999, *SYSDFTCHL
Long retry count . . . . . .  999999999__  0-999999999, *SYSDFTCHL
Security exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
Security exit user data . . . .  *SYSDFTCHL_____________________

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 26. Kanal erstellen (2)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Send exit . . . . . .  *NONE______  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________ Name
+ for more values   __________
Send exit user data . . . . . .  _________________________________
+ for more values  _____________________________
Receive exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
+ for more values  __________
__________
Receive exit user data . . . . .  _________________________________
+ for more values  _____________________________
Message exit . . . . . .  *NONE_____  Name, *SYSDFTCHL, *NONE
Library . . . . . .  __________  Name
+ for more values  __________
__________
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 27. Kanal erstellen (3)*

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Message exit user data . . . . .  _________________________________
+ for more values  _____________________________
Convert message . . . . . .  *SYSDFTCHL_  *YES, *NO, *SYSDFTCHL
Sequence number wrap . . . . . .  999999999__  100-999999999, *SYSDFTCHL
Maximum message length . . . . .  4194304____  0-4194304, *SYSDFTCHL
Heartbeat interval . . . . . .  300________  0-999999999, *SYSDFTCHL
Non Persistent Message Speed . .  *FAST_____  *FAST, *NORMAL, *SYSDFTCHL
Password . . . . . .  *SYSDFTCHL_  Character value, *BLANK...
Task User Profile . . . . . .  *SYSDFTCHL_  Character value, *BLANK...
Transaction Program Name . . . .  *SYSDFTCHL




Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 28. Kanal erstellen (4)*

## IBM i Kanal unter IBM i starten

Sie können einen Kanal über die Anzeige "Mit Kanälen arbeiten" oder über den Befehl STRMQMCHL in der Befehlszeile starten.

Empfangsprogramme sind nur für TCP gültig. Für SNA-Empfangsprogramme müssen Sie das DFV-Subsystem konfigurieren.

Damit Anwendungen Nachrichten austauschen können, müssen Sie ein Empfangsprogramm für eingehende Verbindungen mit dem Befehl STRMQMLSR starten.

Für abgehende Verbindungen müssen Sie den Kanal auf eine der folgenden Arten starten:

1. Den CL-Befehl STRMQMCHL unter Angabe des Kanalnamens verwenden, um den Kanal in Abhängigkeit vom Parameter MCATYPE als Prozess oder als Thread zu starten. (Wenn Kanäle als Threads gestartet werden, handelt es sich um Threads eines Kanalinitiators.)

```
STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
```

2. Verwenden Sie einen Kanalinitiator, um den Kanal auszulösen. Ein Kanalinitiator wird automatisch gestartet, wenn der WS-Manager gestartet wird. Dieser automatische Start kann durch Ändern der Zeilengruppe 'chinit' in der Datei 'qm.ini' für diesen WS-Manager entfernt werden.
3. Verwenden Sie den Befehl WRKMQMCHL, um mit der Anzeige "Mit Kanälen arbeiten" zu beginnen, und wählen Sie die Option 14 aus, um einen Kanal zu starten.

## IBM i Kanal unter IBM i auswählen

Sie können einen Kanal in der Anzeige "Mit Kanälen arbeiten" auswählen.

Um einen Kanal auszuwählen, verwenden Sie den Befehl WRKMQMCHL, um mit der Anzeige Work with Channels zu beginnen:

1. Setzen Sie den Cursor in das Optionsfeld, das dem erforderlichen Kanalnamen zugeordnet ist.
2. Geben Sie eine Optionsnummer ein.
3. Drücken Sie die Eingabetaste, um die Auswahl zu aktivieren.

Wenn Sie mehr als einen Kanal auswählen, werden die Optionen in der Reihenfolge aktiviert.

```
Work with MQM Channels

Queue Manager Name . . :  CNX

Type options, press Enter.
2=Change  3=Copy  4=Delete  5=Display  8=Work with Status  13=Ping
14=Start  15=End  16=Reset  17=Resolve

Opt   Name           Type    Transport   Status
CHLNIC          *RCVR    *TCP     INACTIVE
CORSAIR.TO.MUSTANG    *SDR    *LU62    INACTIVE
FV.CHANNEL.MC.DJE1    *RCVR    *TCP     INACTIVE
FV.CHANNEL.MC.DJE2    *SDR    *TCP     INACTIVE
FV.CHANNEL.MC.DJE3    *RQSTR   *TCP     INACTIVE
FV.CHANNEL.MC.DJE4    *SVR    *TCP     INACTIVE
FV.CHANNEL.PETER     *RCVR    *TCP     INACTIVE
FV.CHANNEL.PETER.LU   *RCVR    *LU62    INACTIVE
FV.CHANNEL.PETER.LU1   *RCVR    *LU62    INACTIVE
More...
Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F12=Cancel
F21=Print
```

*Abbildung 29. Mit Kanälen arbeiten*

## IBM i Durchsuchen eines Kanals unter IBM i

Sie können einen Kanal über die Anzeige "Kanal anzeigen" oder über den Befehl DSPMQMCHL in der Befehlszeile durchsuchen.

Um die Einstellungen eines Kanals zu durchsuchen, verwenden Sie den Befehl WRKMQMCHL, um in der Anzeige "Kanal anzeigen" zu beginnen:

1. Geben Sie die Option 5 (Display) gegen den erforderlichen Kanalnamen ein.
2. Drücken Sie die Eingabetaste, um die Auswahl zu aktivieren.

Wenn Sie mehr als einen Kanal auswählen, werden diese in der Reihenfolge angezeigt.

Alternativ können Sie den Befehl DSPMQMCHL über die Befehlszeile verwenden.

Dies führt dazu, dass die entsprechende Anzeige "Display Channel" mit Details zu den aktuellen Einstellungen für den Kanal angezeigt wird. Die Felder werden in Kanalattribute beschrieben.

```
Display MQM Channel

Channel name . . . . . :  ST.JST.2TO1
Queue Manager Name . . . . . :  QMREL
Channel type . . . . . :  *SDR
Transport type . . . . . :  *TCP
Text 'description' . . . . . :  John's sender to WINSDOA1

Connection name . . . . . :  MUSTANG


Transmission queue . . . . . :  WINSDOA1

Message channel agent . . . . :
Library . . . . . :
Message channel agent user ID :  *NONE
Batch interval . . . . . :  0
Batch size . . . . . :  50
Disconnect interval . . . . . :  6000



F3=Exit  F12=Cancel  F21=Print
```

*Abbildung 30. TCP/IP-Kanal anzeigen (1)*

```
Display MQM Channel

Short retry interval . . . . . :  60
Short retry count . . . . . :  10
Long retry interval . . . . . :  6000
Long retry count . . . . . :  10
Security exit . . . . . :
Library . . . . . :
Security exit user data . . . :
Send exit . . . . . :
Library . . . . . :
Send exit user data . . . . . :
Receive exit . . . . . :
Library . . . . . :
Receive exit user data . . . . :
Message exit . . . . . :
Library . . . . . :
Message exit user data . . . . :
More...



F3=Exit  F12=Cancel  F21=Print
```

*Abbildung 31. TCP/IP-Kanal anzeigen (2)*

```
Display MQM Channel

Sequence number wrap . . . . . :  999999999
Maximum message length . . . . :  10000
Convert message . . . . . . . . :  *NO
Heartbeat interval . . . . . . .  300
Nonpersistent message speed . .  *FAST









Bottom



F3=Exit  F12=Cancel  F21=Print
```

*Abbildung 32. TCP/IP-Kanal anzeigen (3)*

## IBM i Kanal unter IBM i umbenennen

Sie können einen Kanal in der Anzeige "Mit Kanälen arbeiten" umbenennen.

Wenn Sie einen Nachrichtenkanal umbenennen möchten, beginnen Sie in der Anzeige "Mit Kanälen arbeiten":

1. Beenden Sie den Kanal.
2. Verwenden Sie Option 3 (Kopieren), um ein Duplikat mit dem neuen Namen zu erstellen.
3. Verwenden Sie Option 5 (Anzeigen), um zu überprüfen, ob sie korrekt erstellt wurde.
4. Verwenden Sie Option 4 (Löschen), um den ursprünglichen Kanal zu löschen.

Wenn Sie einen Nachrichtenkanal umbenennen möchten, müssen Sie sicherstellen, dass beide Kanalenden gleichzeitig umbenannt werden.

## IBM i Mit Kanalstatus unter IBM i arbeiten

Sie können mit dem Kanalstatus in der Anzeige "Mit Kanalstatus arbeiten" arbeiten.

Verwenden Sie den Befehl WRKMQMCHST, um die erste einer Gruppe von Anzeigen anzuzeigen, die den Status Ihrer Kanäle anzeigt. Sie können die Statusanzeigen in der Reihenfolge anzeigen, wenn Sie die Option Ansicht ändern (F11) auswählen.

Alternativ wird in der Anzeige "Mit MQM-Kanälen arbeiten" auch die Option 8 (Mit Status arbeiten) in der Anzeige "Mit MQM-Kanälen arbeiten" aufgerufen.

```
MQSeries Work with Channel Status

Type options, press Enter.
5=Display  13=Ping  14=Start  15=End  16=Reset  17=Resolve


Opt Name          Connection      Indoubt   Last Seq
CARTS_CORSAIR_CHAN  GBIBMIYA.WINSDOA1    NO        1
CHLNIC        9.20.2.213       NO        3
FV.CHANNEL.PETER2   9.20.2.213       NO      6225
JST.1.2        9.20.2.201       NO       28
MP_MUST_TO_CORS    9.20.2.213       NO       100
MUSTANG.TO.CORSAIR  GBIBMIYA.WINSDOA1    NO       10
MP_CORS_TO_MUST    9.20.2.213       NO       101
JST.2.3        9.5.7.126        NO       32
PF_WINSDOA1_LU62   GBIBMIYA.IYA80020    NO       54
PF_WINSDOA1_LU62   GBIBMIYA.WINSDOA1    NO       500
ST.JCW.EXIT.2TO1.CHL 9.20.2.213       NO       216

Bottom
Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F11=Change view
F12=Cancel  F21=Print
```

*Abbildung 33. Erster Teil der Gruppe von Kanalstatusanzeigen*

Die folgenden Optionen sind in der Anzeige "Mit Kanalstatus arbeiten" verfügbar:

| Menüoption | Beschreibung |
|---|---|
| 5=Display | Zeigt die Kanaleinstellungen an. |
| 13=Ping | Initialisiert eine Ping-Aktion, falls erforderlich. |
| 14=Start | Startet den Kanal. |
| 15=End | Stoppt den Kanal. |
| 16=Reset | Setzt die Kanalfolgenummer zurück. |
| 17=Resolve | Löst eine unbestätigte Kanalsituation manuell aus. |

## IBM i Auswahlmöglichkeiten für Arbeit mit Kanal unter IBM i

Die Anzeige "Mit Kanälen arbeiten" wird mit dem Befehl WRKMQMCHL erreicht, und es ermöglicht Ihnen, den Status aller aufgelisteten Kanäle zu überwachen und Befehle für ausgewählte Kanäle auszugeben.

Die folgenden Optionen sind in der Anzeige "Mit Kanal arbeiten" verfügbar:

| Menüoption | Beschreibung |
|---|---|
| „2=Change" auf Seite 298 | Ändert die Attribute eines Kanals. |
| „3=Copy" auf Seite 298 | Kopiert die Attribute eines Kanals in einen neuen Kanal. |
| „4=Delete" auf Seite 298 | Löscht einen Kanal. |
| „5=Display" auf Seite 298 | Zeigt die aktuellen Einstellungen für den Kanal an. |
| „6=Create" auf Seite 299 | Zeigt die Anzeige 'Kanal erstellen' an |
| „8 = Mit Status arbeiten" auf Seite 299 | Zeigt die Kanalstatusanzeigen an. |

| Menüoption | Beschreibung |
|---|---|
| „13=Ping” auf Seite 300 | Führt die Ping-Funktion aus, um die Verbindung zum benachbarten System zu testen, indem eine feste Datennachricht mit dem fernen Ende ausgetauscht wird. |
| „14=Start” auf Seite 300 | Startet den ausgewählten Kanal oder setzt einen inaktivierten Empfängerkanal zurück. |
| „15=End” auf Seite 301 | Fordert den Kanal zum Schließen an. |
| „16=Reset” auf Seite 302 | Fordert den Kanal an, die Folgenummern an diesem Ende des Links zurückzusetzen. Die Zahlen müssen an beiden Enden gleich sein, damit der Kanal gestartet werden kann. |
| „17=Resolve” auf Seite 302 | Fordert den Kanal an, um unbestätigte Nachrichten aufzulösen, ohne dass eine Verbindung zum anderen Ende hergestellt wird. |
| „18 = Berechtigung anzeigen” auf Seite 303 | Zeigt die IBM MQ-Objektberechtigung an |
| „19 = Berechtigung erteilen” auf Seite 303 | Erteilt die IBM MQ-Objektberechtigung |
| „20 = Berechtigung entziehen” auf Seite 303 | Widerruft die IBM MQ-Objektberechtigung |
| „21 = Objekt wiederherstellen” auf Seite 303 | Stellt das IBM MQ-Objekt wieder her |
| „22 = Satzabbild” auf Seite 303 | Zeichnet das IBM MQ-Objektimage auf |

### IBM i 2=Change

Verwenden Sie die Option 'Ändern', um eine vorhandene Kanaldefinition zu ändern.

Mit der Option Ändern oder mit dem Befehl CHGMQMCHL wird eine vorhandene Kanaldefinition mit Ausnahme des Kanalnamens geändert. Geben Sie die zu ändernden Felder in der Anzeige für die Kanaldefinition ein, und speichern Sie die aktualisierte Definition, indem Sie die Eingabetaste drücken.

### IBM i 3=Copy

Verwenden Sie die Option Kopieren, um einen vorhandenen Kanal zu kopieren.

Die Option "Kopieren" verwendet den Befehl CPYMQMCHL, um einen vorhandenen Kanal zu kopieren. In der Anzeige "Kopieren" können Sie den neuen Kanalnamen definieren. Die verwendbaren Zeichen müssen jedoch auf diejenigen eingeschränkt werden, die für IBM i-Objektnamen gültig sind. Weitere Informationen finden Sie im Abschnitt IBM MQ for IBM i verwalten.

Drücken Sie die Eingabetaste in der Anzeige "Kopieren", um die Details der aktuellen Einstellungen anzuzeigen. Sie können jede der neuen Kanaleinstellungen ändern. Speichern Sie die neue Kanaldefinition durch Drücken der Eingabetaste.

### IBM i 4=Delete

Verwenden Sie die Option Löschen, um den ausgewählten Kanal zu löschen.

Es wird ein Fenster angezeigt, in dem Sie Ihre Anforderung bestätigen oder abbrechen können.

### IBM i 5=Display

Verwenden Sie die Option Anzeigen, um die aktuellen Definitionen für den Kanal anzuzeigen.

Mit dieser Auswahl wird die Anzeige mit den Feldern angezeigt, die die aktuellen Werte der Parameter anzeigen und die gegen Benutzereingabe geschützt sind.

### IBM i 6=Create

Verwenden Sie die Option Erstellen, um die Anzeige 'Kanal erstellen' anzuzeigen.

Verwenden Sie die Option Erstellen, oder geben Sie den Befehl CRTMQMCHL in der Befehlszeile ein, um die Anzeige "Kanal erstellen" zu erhalten. Ab Abbildung 25 auf Seite 292 gibt es Beispiele für 'Create Channel'-Anzeigen.

In dieser Anzeige erstellen Sie eine Kanaldefinition aus einer Anzeige von Feldern, die mit Standardwerten befüllt sind, die von IBM MQ for IBM ibereitgestellt werden. Geben Sie den Namen des Kanals ein, wählen Sie den Typ des zu erstellenden Kanals und die zu verwendende Übertragungsmethode aus.

Wenn Sie die Eingabetaste drücken, wird die Anzeige aufgerufen. Geben Sie Informationen in alle erforderlichen Felder in dieser Anzeige und die übrigen Anzeigen ein, und speichern Sie die Definition, indem Sie die Eingabetaste drücken.

Der Kanalname muss an beiden Enden des Kanals identisch sein und innerhalb des Netzes eindeutig sein. Die verwendbaren Zeichen müssen jedoch auf diejenigen Zeichen eingeschränkt werden, die für IBM MQ for IBM i-Objektnamen gültig sind.

Für alle Anzeigen sind Standardwerte vorhanden, die von IBM MQ for IBM i für einige Felder bereitgestellt werden. Sie können diese Werte anpassen, oder Sie können sie ändern, wenn Sie Kanäle erstellen oder kopieren. Informationen zum Anpassen der Werte finden Sie in der *IBM MQ for IBM i-Systemverwaltung*.

Sie können eigene Kanalstandardwerte erstellen, indem Sie Dummy-Kanäle mit den erforderlichen Standardwerten für jeden Kanaltyp einrichten und sie jedes Mal kopieren, wenn Sie neue Kanaldefinitionen erstellen möchten.

**Zugehörige Verweise**

Kanalattribute

### IBM i 8 = Mit Status arbeiten

Verwenden Sie "Mit Status arbeiten", um detaillierte Kanalstatusinformationen anzuzeigen.

In der Statusspalte wird angezeigt, ob der Kanal aktiv oder inaktiv ist, und wird in der Anzeige "Mit MQM-Kanälen arbeiten" kontinuierlich angezeigt. Verwenden Sie Option 8 (Mit Status arbeiten), um weitere Statusinformationen anzuzeigen. Alternativ können diese Informationen über die Befehlszeile mit dem Befehl WRKMQMCHST angezeigt werden. Weitere Informationen finden Sie unter „Mit Kanalstatus unter IBM i arbeiten” auf Seite 296.

- Kanalname
- Kanaltyp
- Kanalstatus
- Kanalinstanz
- Ferner Warteschlangenmanager
- Name der Übertragungswarteschlange
- Kommunikationsverbindungsname
- Status des Kanals im Status 'Unbestätigt'
- Letzte Folgenummer
- Anzahl der unbestäselten Nachrichten
- Folgenummer in unbestälicher Reihenfolge
- Anzahl der Nachrichten in der Übertragungswarteschlange
- ID der logischen Arbeitseinheit
- Unbestätigte logische Arbeitseinheit mit Kennung
- Teilstatus von Kanal
- Kanalüberwachung
- Header-Komprimierung

- Nachrichtenkomprimierung
- Anzeiger für Komprimierungszeit
- Komprimierungsratenanzeiger
- Zeitindikator für Übertragungswarteschlange
- Netzzeitanzeiger
- Anzeiger für Exitzeit
- Anzeiger für Stapelgröße
- Aktuelle gemeinsame Dialoge
- Maximale Anzahl gemeinsamer Dialoge

### IBM i *13=Ping*

Verwenden Sie die Ping-Option, um eine feste Datennachricht mit dem fernen Ende auszutauschen.

Ein erfolgreiches IBM MQ-Ping gibt dem Systembetreuer eine gewisse Sicherheit, dass der Kanal verfügbar und funktionsfähig ist.

Ping bedeutet nicht die Verwendung von Übertragungswarteschlangen und Zielwarteschlangen. Sie verwendet Kanaldefinitionen, die zugehörige Kommunikationsverbindung und die Netzkonfiguration.

Es ist nur über Sender-und Serverkanäle verfügbar. Der entsprechende Kanal wird an der fernen Seite des Links gestartet und führt die Startparametervereinbarung durch. Fehler werden normal benachrichtigt.

Das Ergebnis des Nachrichtenaustauschs wird in der Ping-Anzeige für Sie angezeigt und ist der zurückgegebene Nachrichtentext, zusammen mit dem Zeitpunkt, zu dem die Nachricht gesendet wurde, und dem Zeitpunkt, zu dem die Antwort empfangen wurde.

## Ping mit LU 6.2

Wenn Ping in IBM MQ for IBM i aufgerufen wird, wird es mit der Benutzer-ID des Benutzers ausgeführt, der die Funktion anfordert, während ein Kanalprogramm normalerweise für die QMQM-Benutzer-ID, die für Kanalprogramme verwendet werden soll, ausgeführt wird. Die Benutzer-ID fließt zur Empfängerseite, und sie muss auf der empfangenden Seite gültig sein, damit der LU 6.2-Dialog zugeordnet werden kann.

### IBM i *14=Start*

Verwenden Sie die Option 'Start', um einen Kanal manuell zu starten.

Die Option Start steht für Sender-, Server-und Requesterkanäle zur Verfügung. Es ist nicht erforderlich, dass ein Kanal mit dem Warteschlangenmanager des Warteschlangenmanagers eingerichtet wurde.

Die Option Start wird auch für Empfänger-, Serververbindung-, Clustersenderkanäle und Clusterempfängerkanäle verwendet. Wird ein Empfängerkanal gestartet, der sich im Status STOPPED befindet, kann er über den fernen Kanal gestartet werden.

Nach dem Start liest der sendende MCA die Kanaldefinitionsdatei und öffnet die Übertragungswarteschlange. Es wird eine Kanalstartsequenz ausgegeben, die den entsprechenden Nachrichtenkanalserver (MCA) des Empfängers oder Serverkanals über Remotezugriff startet. Wenn sie gestartet wurden, warten die Absender-und Serverprozesse auf Nachrichten, die in die Übertragungswarteschlange eintreffen und sie bei ihrer Ankunft übertragen.

Wenn Sie die Triggerung verwenden, müssen Sie den kontinuierlich aktiven Auslöserprozess starten, um die Initialisierungswarteschlange zu überwachen. Der Befehl STRMQMCHLI kann zum Starten des Prozesses verwendet werden.

Am äußersten Ende eines Kanals kann der Empfangsprozess als Antwort auf einen Kanalstart von der sendenden Seite aus gestartet werden. Die Vorgehensweise ist bei LU 6.2-und TCP/IP-Kanälen unterschiedlich:

- Für LU 6.2-Kanäle, die miteinander verbunden sind, ist keine explizite Aktion am empfangenden Ende eines Kanals erforderlich.
- Für angeschlossene TCP-Kanäle ist es erforderlich, dass ein Empfangsprogrammprozess kontinuierlich ausgeführt wird. Dieser Prozess wartet auf Kanalstartanforderungen vom fernen Ende des Links und startet den Prozess, der in den Kanaldefinitionen für diese Verbindung definiert ist.

  Wenn es sich bei dem fernen System um IBM i handelt, können Sie den Befehl STRMQMLSR verwenden.

Die Verwendung der Option Start bewirkt, dass der Kanal bei Bedarf immer resynchronisiert wird.

Damit der Start erfolgreich ist:

- Kanaldefinitionen, lokale und ferne müssen vorhanden sein. Wenn für einen Empfänger-oder Serververbindungskanal keine entsprechende Kanaldefinition vorhanden ist, wird automatisch ein Standardkanal erstellt, wenn der Kanal automatisch definiert wird. Siehe Exitprogramm für die automatische Kanaldefinition (Channel Auto-Definition)
- Die Übertragungswarteschlange muss vorhanden sein, für GETs aktiviert sein und keine anderen Kanäle verwenden.
- MCAs, lokale und ferne, müssen vorhanden sein.
- Die Kommunikationsverbindung muss verfügbar sein.
- Die WS-Manager müssen aktiv, lokal und fern sein.
- Der Nachrichtenkanal muss inaktiv sein.

Damit Nachrichten übertragen werden können, müssen ferne Warteschlangen und Definitionen ferner Warteschlangen vorhanden sein.

Es wird eine Nachricht an die Anzeige zurückgegeben, in der bestätigt wird, dass die Anforderung zum Starten eines Kanals akzeptiert wurde. Überprüfen Sie zur Bestätigung, dass der Startprozess erfolgreich war, das Systemprotokoll, oder drücken Sie F5 (Anzeige aktualisieren).

### IBM i *15=End*

Verwenden Sie das Ende zum Stoppen der Kanalaktivität.

Verwenden Sie die Option "Ende", um den Kanal anzufordern, die Aktivität zu stoppen. Der Kanal sendet keine weiteren Nachrichten.

Wählen Sie F4 vor dem Drücken der Eingabetaste aus, um auszuwählen, ob der Kanal STOPPED oder INACTIVE wird, und ob der Kanal mit Hilfe einer CONTROLLED oder einem Stopp von IMMEDIATE gestoppt werden soll. Ein gestoppte Kanal muss vom Bediener erneut gestartet werden, um wieder aktiv zu werden. Es kann ein inaktiver Kanal ausgelöst werden.

## Sofortiges Stoppen

Verwenden Sie "Stop immediate", um einen Kanal zu stoppen, ohne eine Arbeitseinheit zu beenden.

Mit dieser Option wird der Kanalprozess beendet. Dies führt dazu, dass der Kanal die Verarbeitung des aktuellen Nachrichtenstroms nicht abgeschlossen hat, und kann daher den Kanal im Zweifel nicht verlassen. Im Allgemeinen ist es für die Operatoren besser, die Option zum kontrollierten Stoppen zu verwenden.

## Stopp gesteuert

Verwenden Sie Stop kontrolliert, um einen Kanal am Ende der aktuellen Arbeitseinheit zu stoppen.

Diese Auswahl fordert den Kanal auf, ordnungsgemäß zu schließen; die aktuelle Nachrichtenstelle ist abgeschlossen, und die Synchronisationspunktprozedur wird mit dem anderen Ende des Kanals ausgeführt.

## Gestoppte Kanäle erneut starten

Wenn ein Kanal in den Status STOPPED wechselt, müssen Sie den Kanal manuell erneut starten. Sie können den Kanal folgendermaßen erneut starten:

- Mit dem Befehl **START CHANNEL**.
- Mit dem PCF-Befehl **Start Channel**.
- Mit dem IBM MQ Explorer.
- z/OS Unter z/OS mithilfe der Anzeige 'Kanal starten'.
- IBM i Unter IBM i mit dem Befehl **STRMQMCHL CL** oder der Option **START** in der Anzeige WRKMQMCHL.

Für Sender-oder Serverkanäle wurde die zugeordnete Übertragungswarteschlange auf GET (DISABLED) gesetzt und die Auslösung wurde inaktiviert, wenn der Kanal in den Status STOPPED (STOPPED) eingetreten ist. Wenn die Startanforderung empfangen wird, werden diese Attribute automatisch zurückgesetzt.

z/OS Wenn der Kanalinitiator gestoppt wird, während sich ein Kanal im Status RETRYING oder STOPPED befindet, wird der Kanalstatus gespeichert, wenn der Kanalinitiator erneut gestartet wird. Der Kanalstatus für den Kanaltyp SVRCONN wird jedoch zurückgesetzt, wenn der Kanalinitiator gestoppt wird, während sich der Kanal im Status STOPPED befindet.

Multi Wenn der Warteschlangenmanager gestoppt wird, während sich ein Kanal im Status RETRYING oder STOPPED befindet, wird der Kanalstatus gespeichert, wenn der Warteschlangenmanager erneut gestartet wird. Ab IBM MQ 8.0 gilt dies auch für SVRCONN-Kanäle. Zuvor wurde der Kanalstatus für den Kanaltyp SVRCONN zurückgesetzt, wenn der Kanalinitiator gestoppt wurde, während sich der Kanal im Status STOPPED befand.

### IBM i *16=Reset*

Verwenden Sie die Option Zurücksetzen, um eine neue Nachrichtenfolge zu erzwingen.

Mit der Option Zurücksetzen wird die Nachrichtenfolgenummer geändert. Verwenden Sie diese Option mit Vorsicht, und erst nachdem Sie die Option Resolve verwendet haben, um alle unbestätgen Situationen zu beheben. Diese Option ist nur auf dem Sender-oder Serverkanal verfügbar. Die erste Nachricht startet die neue Sequenz, wenn der Kanal das nächste Mal gestartet wird.

### IBM i *17=Resolve*

Verwenden Sie die Option "Resolve", um eine lokale Festschreibung oder Zurücksetzung von unbestägen Nachrichten in einer Übertragungswarteschlange zu erzwingen.

Verwenden Sie die Option Resolve, wenn Nachrichten von einem Sender oder Server im Zweifel gehalten werden, z. B. weil ein Ende der Verbindung beendet wurde und keine Aussicht auf Wiederherstellung besteht. Die Option Resolve akzeptiert einen der beiden Parameter: BACKOUT oder COMMIT. Mit Backout werden Nachrichten in die Übertragungswarteschlange zurückgespeichert, während Commit sie löscht.

Das Kanalprogramm versucht nicht, eine Sitzung mit einem Partner aufzubauen. Stattdessen bestimmt sie die ID der logischen Arbeitseinheit (LUWID), die die unbestätige_Nachrichten darstellt. Anschließend gibt es, wie angefordert, folgende Probleme aus:

- BACKOUT, um die Nachrichten in die Übertragungswarteschlange zurückzuspeichen; oder
- COMMIT, um die Nachrichten aus der Übertragungswarteschlange zu löschen.

Damit die Auflösung erfolgreich ist:

- Der Kanal muss inaktiv sein.
- Der Kanal muss im Zweifel sein.
- Der Kanaltyp muss "sender" oder "server" sein
- Die Kanaldefinition (lokal) muss vorhanden sein.

- Der Warteschlangenmanager muss aktiv, lokal ausgeführt werden.

### IBM i *18 = Berechtigung anzeigen*

Verwenden Sie die Option "Berechtigung anzeigen", um anzuzeigen, welche Aktionen ein Benutzer für ein bestimmtes IBM MQ-Objekt ausführen darf.

Für ein ausgewähltes Objekt und einen ausgewählten Benutzer zeigt der Befehl DSPMQAUT die Berechtigungen an, über die der Benutzer verfügt, um Aktionen für ein IBM MQ-Objekt auszuführen. Wenn der Benutzer Mitglied mehrerer Gruppen ist, zeigt der Befehl die kombinierte Berechtigung aller Gruppen für das Objekt an.

### IBM i *19 = Berechtigung erteilen*

Verwenden Sie die Option "Berechtigung erteilen", um einem anderen Benutzer oder einer anderen Benutzergruppe die Berechtigung zum Ausführen von Aktionen für IBM MQ-Objekte erteilen.

Der Befehl GRTMQMAUT ist nur für Benutzer in der Gruppe QMQMADM verfügbar. Ein Benutzer in QMQMADM erteilt anderen Benutzern die Berechtigung zum Ausführen von Aktionen für die IBM MQ-Objekte, die im Befehl angegeben sind, entweder durch namentliche Angabe der Benutzer oder durch Erteilen der Berechtigung an alle Benutzer in *PUBLIC.

### IBM i *20 = Berechtigung entziehen*

Verwenden Sie die Widerrufberechtigung, um die Berechtigung zum Ausführen von Aktionen für Objekte von Benutzern zu entfernen.

Der Befehl RVKMQMAUT ist nur für Benutzer in der Gruppe QMQMADM verfügbar. Ein Benutzer in der Gruppe QMQMADM entzieht anderen Benutzern die Berechtigung, Aktionen für die im Befehl angegebenen IBM MQ-Objekte auszuführen, entweder durch Angabe der Benutzer nach Namen oder durch Widerrufen der Berechtigung von allen Benutzern in *PUBLIC.

### IBM i *21 = Objekt wiederherstellen*

Mit 'Objekt wiederherstellen' können Sie beschädigte Objekte aus Informationen zurückspeichern, die in IBM MQ-Journalen gespeichert sind.

Das Objekt 'Recover' verwendet den Befehl 'Re-create MQ Object' (RCRMQMOBJ), um alle beschädigten Objekte wiederherzustellen, die im Befehl angegeben sind. Wenn ein Objekt nicht beschädigt ist, wird für dieses Objekt keine Aktion ausgeführt.

### IBM i *22 = Satzabbild*

Verwenden Sie das Datensatzimage, um die Anzahl der Journalempfänger zu reduzieren, die für die Wiederherstellung einer Gruppe von Objekten erforderlich sind, und um die Wiederherstellungszeit zu minimieren.

Der Befehl RCDMQMIMG verwendet einen Prüfpunkt für alle Objekte, die im Befehl ausgewählt wurden. Es synchronisiert die aktuellen Werte der Objekte im Integrated File System (IFS) mit späteren Informationen zu den Objekten, wie z. B. MQPUTs und MQGETs, die in Journalempfängern aufgezeichnet wurden.

Wenn der Befehl die Objekte im IFS abschließt, sind die Objekte auf dem neuesten Stand, und diese Journalempfänger müssen nicht mehr vorhanden sein, um die Objekte wiederherzustellen. Alle nicht verbundenen Journalempfänger können abgehängt werden (solange sie nicht vorhanden sind, um andere Objekte wiederherzustellen).

## IBM i Kommunikation für IBM i konfigurieren

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit sie erfolgreich ist, ist es erforderlich, dass die Verbindung definiert und verfügbar ist.

DQM ist eine ferne Warteschlangenfunktion für IBM MQ for IBM i. Es stellt Kanalsteuerprogramme für den IBM MQ for IBM i-Warteschlangenmanager bereit, die die Schnittstelle zu Kommunikationsverbindungen bilden, die vom Systembediener gesteuert werden können.

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit sie erfolgreich ist, ist es erforderlich, dass die Verbindung definiert und verfügbar ist. In diesem Abschnitt wird erläutert, wie Sie sicherstellen können, dass die Verbindung definiert und verfügbar ist.

Bevor ein Kanal gestartet werden kann, muss die Übertragungswarteschlange wie in diesem Abschnitt beschrieben definiert sein und muss in die Nachrichtenkanaldefinition eingeschlossen werden.

Sie können zwischen den folgenden beiden Formen der Kommunikation zwischen IBM MQ for IBM i-Systemen wählen:

- „TCP-Verbindung unter IBM i definieren” auf Seite 304

  Für TCP kann eine Hostadresse verwendet werden, und diese Verbindungen werden wie in der *IBM i Communication Configuration Reference* beschrieben konfiguriert.

  In der TCP-Umgebung wird jedem verteilten Service eine eindeutige TCP-Adresse zugeordnet, die von fernen Maschinen für den Zugriff auf den Service verwendet werden kann. Die TCP-Adresse setzt sich aus einem Hostnamen/einer Portnummer und einer Portnummer zusammen. Alle WS-Manager verwenden eine solche Zahl, um über TCP miteinander zu kommunizieren.
- „Empfang auf TCP” auf Seite 305

  Diese Art der Kommunikation erfordert die Definition einer logischen Einheit des Typs 6.2 (LU 6.2) der IBM i-Systemnetzwerkarchitektur, in der die physische Verbindung zwischen dem IBM i-System, das den lokalen WS-Manager bedient, und dem System, das den fernen WS-Manager bedient, bereitgestellt wird. Details zur Konfiguration der Kommunikation in IBM ifinden Sie im Handbuch *IBM i Communication Configuration Reference* .

Bei Bedarf muss die Auslöseranordnung außerdem mit der Definition der erforderlichen Prozesse und Warteschlangen vorbereitet werden.

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

**Zugehörige Tasks**

„Kanäle in IBM i überwachen und steuern” auf Seite 289
Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern. Jeder WS-Manager verfügt über ein DQM-Programm zur Steuerung von Verbindungen zu kompatiblen fernen Warteschlangenmanagern.

**Zugehörige Verweise**

Beispielkonfiguration - IBM MQ for IBM i

Beispiel für Nachrichtenkanalplanung für IBM MQ for IBM i

Jobs für übergreifende Kommunikation unter IBM i

Kanalzustände unter IBM i

## IBM i *TCP-Verbindung unter IBM i definieren*

Sie können eine TCP-Verbindung in der Kanaldefinition mit Hilfe des Felds Verbindungsname definieren.

Die Kanaldefinition enthält ein Feld (CONNECTION NAME), das entweder die TCP-Netzadresse des Ziels oder den Hostnamen (z. B. ABCHOST) enthält. Die TCP-Netzadresse kann in Dezimalschreibweise mit Trennzeichen gemäß IPv4 (z. B. 127.0.0.1) oder im Hexadezimalformat nach IPv6 (z. B. 2001:DB8:0:0:0:0:0:0) angegeben werden. Wenn im Feld für den Verbindungsnamen (CONNECTION

NAME) ein Hostname oder ein Namensserver angegeben ist, wird der Hostname mithilfe der IBM i-Hosttabelle in eine TCP-Hostadresse konvertiert.

Für eine vollständige TCP-Adresse ist eine Portnummer erforderlich. Wenn diese Nummer nicht angegeben wird, wird die Standardportnummer 1414 verwendet. Am einleitenden Ende einer Verbindung (Sender-, Requester-und Serverkanaltypen) ist es möglich, eine optionale Portnummer für die Verbindung bereitzustellen, z. B.:

```
Connection name 127.0.0.1 (1555)
```

In diesem Fall versucht das einleitende Ende, eine Verbindung zu einem empfangenden Programm an Port 1555 herzustellen.

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

## Verwendung der Option TCP-Listener-Backlog

In TCP werden die Verbindungen nur unvollständig behandelt, wenn zwischen dem Server und dem Client ein Dreiwege-Handshake nicht stattfindet. Diese Verbindungen werden als ausstehende Verbindungsanforderungen bezeichnet. Für diese ausstehenden Verbindungsanforderungen wird ein Maximalwert festgelegt und kann als Rückstand von Anforderungen betrachtet werden, die auf den TCP-Port warten, damit der Listener die Anforderung akzeptiert.

Weitere Informationen und den jeweiligen Wert für IBM ifinden Sie unter „Verwenden der Backlog-Option für den TCP-Listener unter IBM MQ for Multiplatforms” auf Seite 286 .

**Zugehörige Konzepte**

„Empfang auf TCP” auf Seite 305
Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet. Um auf die Startanforderung zu reagieren, muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Dieses Listenerprogramm wird mit dem Befehl STRMQMLSR gestartet.

IBM i *Empfang auf TCP*
Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet. Um auf die Startanforderung zu reagieren, muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Dieses Listenerprogramm wird mit dem Befehl STRMQMLSR gestartet.

Sie können für jeden WS-Manager mehr als einen Listener starten. Der Befehl STRMQMLSR verwendet standardmäßig Port 1414, aber Sie können diesen Wert überschreiben. Wenn Sie die Standardeinstellung überschreiben möchten, fügen Sie die folgenden Anweisungen zur Datei qm.ini des ausgewählten Warteschlangenmanagers hinzu. In diesem Beispiel ist der Listener für die Verwendung von Port 2500 erforderlich:

```
TCP:
Port=2500
```

Die Datei qm.ini befindet sich in diesem IFS-Verzeichnis: /QIBM/UserData/mqm/qmgrs/ *Warteschlangenmanagername* .

Dieser neue Wert ist schreibgeschützt, wenn der TCP-Listener gestartet wird. Wenn ein Listener bereits aktiv ist, wird diese Änderung von diesem Programm nicht angezeigt. Wenn Sie den neuen Wert verwenden möchten, stoppen Sie den Listener, und setzen Sie den Befehl STRMQMLSR erneut ab. Wenn Sie nun den Befehl STRMQMLSR verwenden, nimmt der Listener standardmäßig den neuen Port an.

Alternativ können Sie im Befehl STRMQMLSR eine andere Portnummer angeben. For example:

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

Durch diese Änderung wird die Listener-Funktion standardmäßig für den neuen Port für die Dauer des Listenerjobs verwendet.

## Verwendung der Option TCP SO_KEEPALIVE

Wenn Sie die Option SO_KEEPALIVE verwenden möchten (weitere Informationen finden Sie in „Überprüfen, ob das andere Ende des Kanals noch verfügbar ist" auf Seite 247), müssen Sie den folgenden Eintrag zur Konfigurationsdatei des Warteschlangenmanagers hinzufügen ('qm.ini' im IFS-Verzeichnis, /QIBM/UserData/mqm/qmgrs/ *Warteschlangenmanagername* ):

```
TCP:
KeepAlive=yes
```

Sie müssen dann den folgenden Befehl ausgeben:

```
CFGTCP
```

Wählen Sie Option 3 (TCP-Attribute ändern) aus. Sie können jetzt ein Zeitintervall in Minuten angeben. Sie können einen Wert im Bereich von 1 bis 40320 Minuten angeben; der Standardwert ist 120.

## Verwendung der Option TCP-Listener-Backlog

Beim Empfang auf TCP wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Diese Zahl kann als *Rückstand* von Anforderungen betrachtet werden, die auf den TCP-Port für den Listener warten, um die Anforderung zu akzeptieren.

Der Standardwert für das Rückstandsprotokoll des Listeners unter IBM i ist 255. Wenn der Backlog diesen Wert erreicht, wird die TCP-Verbindung zurückgewiesen und der Kanal kann nicht gestartet werden.

Bei MCA-Kanälen führt dies dazu, dass der Kanal in einen RETRY-Status eingeht und die Verbindung zu einem späteren Zeitpunkt erneut versucht.

Für Clientverbindungen empfängt der Client einen Ursachencode MQRC_Q_MGR_NOT_AVAILABLE von MQCONN und kann die Verbindung zu einem späteren Zeitpunkt wiederholen.

Um diesen Fehler zu vermeiden, können Sie jedoch einen Eintrag in der Datei qm.ini hinzufügen:

```
ListenerBacklog = n
```

Dies überschreibt die Standardanzahl ausstehender Anforderungen (255) für den TCP-Listener.

**Anmerkung:** Einige Betriebssysteme unterstützen einen größeren Wert als der Standardwert. Falls erforderlich, kann dieser Wert verwendet werden, um das Erreichen des Verbindungsgrenzwerts zu vermeiden.

## IBM i *LU 6.2-Verbindung unter IBM i definieren*

Definieren Sie die LU 6.2-Kommunikationsdetails unter Verwendung eines Modusnamens, eines TP-Namens und des Verbindungsnamens einer vollständig qualifizierten LU 6.2-Verbindung.

Das eingeleitete Ende des Links muss über eine Leitwegeintragsdefinition verfügen, um dieses CSI-Objekt zu ergänzen. Weitere Informationen zum Verwalten von Arbeitsanforderungen von fernen LU 6.2-Systemen finden Sie im Handbuch *IBM iProgramming: Work Management Guide*.

Informationen hierzu finden Sie im Handbuch *Multiplatform APPC Configuration Guide* und in der folgenden Tabelle.

| *Tabelle 25. Einstellungen auf dem lokalen IBM i-System für eine ferne Warteschlangenmanagerplattform* | |
|---|---|
| **Ferne Plattform** | **TPNAME** |
| z/OS oder MVS | Entsprechendes gilt für die entsprechenden Nebeninformationen zum fernen Warteschlangenmanager. |
| IBM i | Entsprechendes gilt für den Vergleichswert im Routing-Eintrag auf dem IBM i-System. |
| Systeme mit AIX and Linux | Das aufrufbare Transaktionsprogramm, das in der fernen LU 6.2-Konfiguration definiert ist. |
| Windows | Wie im Windows-Befehl 'Listener ausführen' angegeben, oder dem aufrufbaren Transaktionsprogramm, das mit TpSetup unter Windows definiert wurde. |

Wenn mehrere WS-Manager auf demselben Computer vorhanden sind, müssen Sie sicherstellen, dass die TPNames in den Kanaldefinitionen eindeutig sind.

**Zugehörige Konzepte**

„Initialisierungsende (Absender)" auf Seite 307
Verwenden Sie den Befehl CRTMQMCHL, um einen Kanal des Transporttyps *LU62 zu definieren.

„Initiiertes Ende (Empfänger)" auf Seite 310
Mit dem Befehl CRTMQMCHL können Sie das empfangende Ende der Nachrichtenkanalverbindung mit dem Transporttyp *LU62 definieren.

IBM i *Initialisierungsende (Absender)*

Verwenden Sie den Befehl CRTMQMCHL, um einen Kanal des Transporttyps *LU62 zu definieren.

Die Verwendung des CSI-Objekts ist optional in IBM MQ for IBM i V5.3 oder höher.

Die einleitende Endanzeige wird in Abbildung LU 6.2-DFV-Konfigurationsanzeige-einleitendes Ende angezeigt. Drücken Sie die Taste F10 in der ersten Anzeige, um die gesamte Anzeige wie gezeigt zu erhalten.

```
Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . > WINSDOA1   Name
Library . . . . . >  QSYS    Name, *CURLIB
Remote location . . . . . > WINSDOA1   Name
Transaction program . . . . . > MQSERIES

Text 'description' . . . . . .  *BLANK


Additional Parameters

Device . . . . . .  *LOC     Name, *LOC
Local location . . . . . .  *LOC     Name, *LOC, *NETATR
Mode . . . . . .  JSTMOD92   Name, *NETATR
Remote network identifier . . .  *LOC     Name, *LOC, *NETATR, *NONE
Authority . . . . . .  *LIBCRTAUT  Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 34. LU 6.2-DFV-Konfigurationsanzeige-Einleitendes Ende*

Füllen Sie die einleitenden Endfelder wie folgt aus:

**Nebeninformationen**
Geben Sie dieser Definition einen Namen zu, mit dem das Nebeninformationsobjekt gespeichert wird, das erstellt werden soll, z. B. WINSDOA1.

**Anmerkung:** Für LU 6.2 ist die Verbindung zwischen der Nachrichtenkanaldefinition und der Kommunikationsverbindung das Feld **Verbindungsname** der Nachrichtenkanaldefinition an der sendenden Seite. Dieses Feld enthält den Namen des CSI-Objekts.

**Bibliothek**
Der Name der Bibliothek, in der diese Definition gespeichert ist.

Das CSI-Objekt muss in einer Bibliothek verfügbar sein, die für das Programm zugänglich ist, das den Nachrichtenkanal bedient, z. B. QSYS, QMQM und QGPL.

Wenn der Name falsch, fehlt oder nicht gefunden werden kann, tritt beim Kanalstart ein Fehler auf.

**Ferne Position**
Gibt den Namen des fernen Standorts an, mit dem das Programm kommuniziert.

Kurz, dieser erforderliche Parameter enthält den Namen der logischen Einheit des Partners auf dem fernen System, wie in der Einheitenbeschreibung definiert, die für die Datenübertragungsverbindung zwischen den beiden Systemen verwendet wird.

Der Name des **fernen Standorts** kann durch Absetzen des Befehls DSPNETA auf dem fernen System und durch das Ansehen des lokalen Standardstandortnamens gefunden werden.

**Transaktionsprogramm**
Gibt den Namen (bis zu 64 Zeichen) des Transaktionsprogramms auf dem fernen System an, das gestartet werden soll. Dabei kann es sich um einen Transaktionsprozessnamen, einen Programmnamen, den Kanalnamen oder eine Zeichenfolge handeln, die mit dem **Vergleichswert** im Leitwegeintrag übereinstimmt.

Dieser Parameter ist erforderlich.

**Anmerkung:** Geben Sie die hexadezimale Darstellung des Namens des Servicetransaktionsprogramms ein, um SNA-Servicetransaktionsprogrammnamen anzugeben Wenn Sie beispielsweise den Namen eines Servicetransaktionsprogramms mit einer hexadezimalen Darstellung von 21F0F0F1 angeben möchten, geben Sie X'21F0F0F1 ' ein.

Weitere Informationen zu SNA-Servicetransaktionsprogrammnamen finden Sie im Handbuch *SNA Transaction Programmer's Reference* für LU-Typ 6.2.

Wenn es sich bei der Empfangsseite um ein weiteres IBM i-System handelt, wird mit dem **Transaktionsprogrammnamen** das CSI-Objekt auf der Sendeseite mit dem Routing-Eintrag auf der Empfängerseite abgeglichen. Dieser Name darf für jeden Warteschlangenmanager im IBM i-Zielsystem jeweils nur einmal vorhanden sein. Siehe den Parameter **Program to call** unter Initiated end (Receiver) . Siehe auch den Parameter **Vergleichsdaten: Vergleichswert** in der Anzeige 'Routing-Eintrag hinzufügen'.

**Textbeschreibung**
Eine Beschreibung (bis zu 50 Zeichen), die Sie an die beabsichtigte Verwendung dieser Verbindung erinnert.

**Einheit**
Gibt den Namen der Einheitenbeschreibung an, die für das ferne System verwendet wird. Folgende Werte sind möglich:

***LOC**
Die Einheit wird durch das System bestimmt.

**Einheitenname**
Geben Sie den Namen der Einheit an, die dem fernen Standort zugeordnet ist.

**Lokaler Standort**
Gibt den Namen des lokalen Standorts an. Folgende Werte sind möglich:

***LOC**
Der Name des lokalen Standorts wird durch das System festgelegt.

***NETATR**
Der in den Systemnetzattributen angegebene Wert für LCLLOCNAME wird verwendet.

**Name des lokalen Standorts**
Geben Sie den Namen des Standorts an. Geben Sie die lokale Position an, wenn Sie einen bestimmten Standortnamen für den fernen Standort angeben möchten. Der Positionsname kann mit dem Befehl DSPNETA (Befehl DSPNETA) gefunden werden.

**Modus**
Gibt den Modus an, der zur Steuerung der Sitzung verwendet wird. Dieser Name ist mit der Common Programming Interface (CPI)-Communications Mode_Name identisch. Folgende Werte sind möglich:

***NETATR**
Der Modus in den Netzwerkattributen wird verwendet.

**BLANK**
Es werden acht Leerzeichen verwendet.

**Modusname**
Geben Sie einen Modusnamen für den fernen Standort an.

**Anmerkung:** Da der Modus die Übertragungspriorität der DFV-Sitzung bestimmt, kann es sinnvoll sein, abhängig von der Priorität der gesendeten Nachrichten unterschiedliche Modi zu definieren; z. B. MQMODE_HI, MQMODE_MED und MQMODE_LOW. (Sie können mehrere CSI-Systeme auf dieselbe Position verweisen.)

**Ferne Netzwerk-ID**
Gibt die ferne Netzwerk-ID an, die mit dem fernen Standort verwendet wird. Folgende Werte sind möglich:

***LOC**
Die ferne Netzwerk-ID für den fernen Standort wird verwendet.

***NETATR**
Die in den Netzwerkattributen angegebene ferne Netzwerk-ID wird verwendet.

***NONE**
Das ferne Netzwerk hat keinen Namen.

**Fernes Netz-ID**
Geben Sie eine ferne Netzwerk-ID an. Verwenden Sie den Befehl DSPNETA an der fernen Position, um den Namen dieser Netzwerk-ID zu ermitteln. Es handelt sich um die lokale Netzwerk-ID am fernen Standort.

**Berechtigung**
Gibt die Berechtigung an, die Benutzern erteilt werden soll, die keine bestimmte Berechtigung für das Objekt haben, die sich nicht in einer Berechtigungsliste befinden, und mit einem Gruppenprofil, das keine bestimmte Berechtigung für das Objekt hat. Folgende Werte sind möglich:

***LIBCRTAUT**
Die öffentliche Berechtigung für das Objekt wird aus dem Parameter CRTAUT der angegebenen Bibliothek übernommen. Dieser Wert wird zur Erstellungszeit bestimmt. Wenn der Wert des Befehls CRTAUT für die Bibliothek nach dem Erstellen des Objekts geändert wird, hat der neue Wert keine Auswirkungen auf vorhandene Objekte.

***ÄNDERN**
Die Änderungsberechtigung ermöglicht es dem Benutzer, Basisfunktionen für das Objekt auszuführen, der Benutzer kann das Objekt jedoch nicht ändern. Die Änderungsberechtigung stellt die Objektverwender-und die gesamte Datenberechtigung zur Verfügung.

***ALL**
Der Benutzer kann alle Operationen ausführen, mit Ausnahme der Operationen, die auf den Eigner beschränkt sind oder von der Berechtigungslistenverwaltungsberechtigung gesteuert werden. Der Benutzer kann die Existenz des Objekts steuern und die Sicherheit für das Objekt angeben, das Objekt ändern und Basisfunktionen für das Objekt ausführen. Der Benutzer kann das Eigentumsrecht für das Objekt ändern.

***NUTZ**
Die Benutzungsberechtigung stellt die Objektverwender-und Leseberechtigung zur Verfügung.

***EXCLUDE**
Die Ausschlussberechtigung verhindert, dass der Benutzer auf das Objekt zugreift.

**Berechtigungsliste**
Geben Sie den Namen der Berechtigungsliste mit der Berechtigung an, die für die Nebeninformationen verwendet wird.

IBM i *Initiiertes Ende (Empfänger)*

Mit dem Befehl CRTMQMCHL können Sie das empfangende Ende der Nachrichtenkanalverbindung mit dem Transporttyp *LU62 definieren.

Lassen Sie das Feld CONNECTION NAME leer, und stellen Sie sicher, dass die entsprechenden Details mit dem sendenden Ende des Kanals übereinstimmen. Weitere Informationen hierzu finden Sie im Abschnitt Kanal erstellen .

Um das einleitende Ende zu aktivieren, um den empfangenden Kanal zu starten, fügen Sie einen Leitwegeintrag zu einem Subsystem am initiierten Ende hinzu. Das Subsystem muss das Subsystem sein, das die APPC-Einheit zuordnet, die in den LU 6.2-Sitzungen verwendet wird. Daher muss sie über einen gültigen DFV-Eintrag für diese Einheit verfügen. Der Leitwegeintrag ruft das Programm auf, das das Empfangsende des Nachrichtenkanals startet.

Verwenden Sie die IBM i-Befehle (z. B. ADDRTGE), um das Ende der Verbindung zu definieren, die von einer Kommunikationssitzung eingeleitet wird.

Die eingeleitete Endanzeige wird in der Anzeige LU 6.2 communication setup panel-add routing entry (Leitwegeintrag hinzufügen) angezeigt.

```
Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description . . . . .  QCMN     Name
Library . . . . . .   *LIBL    Name, *LIBL, *CURLIB
Routing entry sequence number .  1       1-9999
Comparison data:
Compare value . . . . . .  MQSERIES

Starting position . . . . . .  37      1-80
Program to call . . . . . .  AMQCRC6B   Name, *RTGDTA
Library . . . . . .  QMAS400    Name, *LIBL, *CURLIB
Class . . . . . .  *SBSD     Name, *SBSD
Library . . . . . .  *LIBL     Name, *LIBL, *CURLIB
Maximum active routing steps . .  *NOMAX    0-1000, *NOMAX
Storage pool identifier . . . .  1       1-10



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

*Abbildung 35. LU 6.2-DFV-Setup-Anzeige-eingeleitet*

**Subsystembeschreibung**
Der Name des Subsystems, in dem sich diese Definition befindet. Verwenden Sie den IBM i-Befehl WRKSBSD, um die entsprechende Subsystembeschreibung für den Routing-Eintrag anzuzeigen und zu aktualisieren.

**Folgenummer des Leitwegeintrags**
Eine eindeutige Nummer in Ihrem Subsystem, um diese Kommunikationsdefinition zu identifizieren. Sie können Werte im Bereich von 1 bis 9999 verwenden.

**Vergleichsdaten: Vergleichswert**
Eine Textzeichenfolge, die mit der Zeichenfolge verglichen werden soll, die empfangen wird, wenn die Sitzung mit einem **Transaktionsprogramm**-Parameter gestartet wird, wie in Abbildung 1 dargestellt. Die Zeichenfolge wird aus dem Feld Transaktionsprogramm des Absenders CSI abgeleitet.

**Vergleichsdaten: Startposition**
Die Zeichenposition in der Zeichenfolge, an der der Vergleich beginnen soll.

**Anmerkung:** Das Startpositionsfeld ist die Zeichenposition in der Zeichenfolge für den Vergleich, und diese Position ist immer 37.

**Aufrufprogramm**
Der Name des Programms, das das ankommende Nachrichtenprogramm ausführt, das zum Starten der Sitzung aufgerufen werden soll.

Das Programm AMQCRC6A wird für den Standardwarteschlangenmanager aufgerufen. Dieses Programm wird mit IBM MQ for IBM i bereitgestellt und richtet die Umgebung ein und ruft dann AMQCRS6A auf.

Für zusätzliche WS-Manager:

- Jeder WS-Manager verfügt über ein spezielles LU 6.2-aufrufbares Programm in seiner Bibliothek. Dieses Programm wird als AMQCRC6B bezeichnet und wird automatisch beim Erstellen des Warteschlangenmanagers generiert.
- Jeder WS-Manager benötigt einen bestimmten Leitwegeintrag mit eindeutigen Routing-Daten, die hinzugefügt werden sollen. Diese Routing-Daten müssen mit dem Namen des **Transaktionsprogramms** übereinstimmen, der vom anfordernden System bereitgestellt wird (siehe Initialisierungsende (Sender) ).

Ein Beispiel wird in der Anzeige LU 6.2 communication setup panel-display routing entries angezeigt:

```
Display Routing Entries
System:  MY400
Subsystem description:  QCMN      Status:  ACTIVE

Type options, press Enter.
5=Display details

Start
Opt  Seq Nbr  Program    Library    Compare Value   Pos
10   *RTGDTA           'QZSCSRVR'     37
20   *RTGDTA           'QZRCSRVR'     37
30   *RTGDTA           'QZHQTRG'     37
50   *RTGDTA           'QVPPRINT'     37
60   *RTGDTA           'QNPSERVR'     37
70   *RTGDTA           'QNMAPINGD'    37
80   QNMAREXECD  QSYS     'AREXECD'     37
90   AMQCRC6A   QMQMBW    'MQSERIES'     37
100   *RTGDTA           'QTFDWNLD'     37
150   *RTGDTA           'QMFRCVR'     37


F3=Exit  F9=Display all detailed descriptions  F12=Cancel
```

*Abbildung 36. LU 6.2-DFV-Setup-Anzeige-eingeleitet*

In LU 6.2-DFV-Setup-Anzeige - Routing-Einträge anzeigen stellt die Folgenummer 90 den Standardwarteschlangenmanager dar und bietet Kompatibilität mit Konfigurationen aus früheren Releases (d. h. V3R2, V3R6, V3R7 und V4R2) von IBM MQ for IBM i. Mit diesen Releases kann nur ein Warteschlangenmanager ausgeführt werden. Die Folgenummern 92 und 94 stellen zwei zusätzliche WS-Manager mit dem Namen ALPHA und BETA dar, die mit den Bibliotheken QMALPHA und QMBETA erstellt werden.

**Anmerkung:** Sie können für jeden Warteschlangenmanager mehr als einen Leitwegeintrag verwenden, indem Sie verschiedene Routing-Daten verwenden. Diese Einträge geben abhängig von den verwendeten Klassen die Möglichkeit, verschiedene Jobprioritäten zu setzen.

**Klasse**
Der Name und die Bibliothek der Klasse, die für die Schritte verwendet wurden, die durch diesen Leitwegeintrag gestartet wurden. Die Klasse definiert die Attribute der aktiven Umgebung des Routing-Schritts und gibt die Jobpriorität an. Es muss ein entsprechender Klasseneintrag angegeben werden. Verwenden Sie zum Beispiel den Befehl WRKCLS, um vorhandene Klassen anzuzeigen oder um eine Klasse zu erstellen. Weitere Informationen zum Verwalten von Arbeitsanforderungen von fernen LU 6.2-Systemen finden Sie im Handbuch *IBM iProgramming: Work Management Guide*.

### Hinweis zur Arbeitsverwaltung

Der AMQCRS6A-Job kann die normalen IBM i-Arbeitsmanagementfunktionen nicht nutzen, die unter Arbeitsverwaltung dokumentiert sind, da er nicht auf die gleiche Weise wie andere IBM MQ-Jobs gestartet wird. Wenn Sie die Laufzeiteigenschaften der LU62 -Empfängerjobs ändern möchten, können Sie eine der folgenden Änderungen vornehmen:

- Ändern Sie die Klassenbeschreibung, die im Leitwegeintrag für den Job AMQCRS6A angegeben ist.
- Die Jobbeschreibung im DFV-Eintrag ändern.

Weitere Informationen zum Konfigurieren von DFV-Jobs finden Sie im Handbuch *IBM i Programming: Work Management Guide*.

## WS-Manager-Cluster konfigurieren

Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Sie können Cluster-Komponenten definieren und Cluster erstellen und verwalten.

### Vorbereitende Schritte

Eine Einführung in Clusterkonzepte finden Sie unter Cluster.

Wenn Sie den WS-Manager-Cluster entwerfen, müssen Sie einige Entscheidungen treffen. Siehe Beispielcluster und Konstrukte Cluster .

**Zugehörige Tasks**
„Clusterthemendefinition in einen anderen WS-Manager verschieben” auf Seite 474
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

**Zugehörige Verweise**
DELETE TOPIC

### Komponenten eines Clusters definieren

Cluster bestehen aus WS-Managern, Clusterkanälen und Clusterwarteschlangen. Sie können Clusterwarteschlangen definieren und einige Aspekte von Standardclusterobjekten ändern. Sie können Konfigurations-und Statusinformationen zu automatisch definierten Kanälen sowie zu den Beziehungen zwischen einzelnen Clustersenderkanälen und Übertragungswarteschlangen abrufen.

Informationen zum Definieren der einzelnen Clusterkomponenten finden Sie in den folgenden Unterabschnitten:

**Zugehörige Konzepte**
Komponenten eines Clusters
Clusterkanäle

**Zugehörige Tasks**

Clusterthemen definieren

„Neuen Cluster einrichten” auf Seite 327
Führen Sie die folgenden Anweisungen aus, um den Beispielcluster zu konfigurieren. In separaten Anweisungen wird beschrieben, wie der Cluster auf TCP/IP, LU 6.2 und mit einer einzelnen Übertragungswarteschlange oder mehreren Übertragungswarteschlangen eingerichtet wird. Testen Sie den Cluster, indem Sie eine Nachricht von einem WS-Manager an den anderen Warteschlangenmanager senden.

„WS-Manager zu einem Cluster hinzufügen” auf Seite 339
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung der einzigen Clusterübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE übertragen.

### *Clusterwarteschlangen definieren*

Eine Clusterwarteschlange wird von einem Clusterwarteschlangenmanager anderen Warteschlangenmanagern im Cluster zur Verfügung gestellt.Definieren Sie eine Clusterwarteschlange als lokale Warteschlange auf dem Clusterwarteschlangenmanager, auf dem die Warteschlange gehostet wird. Geben Sie den Namen des Clusters an, zu dem die Warteschlange gehört.

Das folgende Beispiel zeigt einen **runmqsc** -Befehl zum Definieren einer Clusterwarteschlange mit der Option CLUSTER :

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

Eine Clusterwarteschlangendefinition wird den anderen Warteschlangenmanagern im Cluster zugänglich gemacht. Die anderen Warteschlangenmanager im Cluster können ohne entsprechende Definition einer fernen Warteschlange Nachrichten in eine Clusterwarteschlange einreihen. Über eine Clusternamensliste kann eine Clusterwarteschlange in mehreren Clustern zugänglich gemacht werden.

Wenn eine Warteschlange zugänglich gemacht wird, können alle Warteschlangenmanager im Cluster Nachrichten in diese Warteschlange einreihen. Um eine Nachricht einzureihen, muss der Warteschlangenmanager anhand der vollständigen Repositorys ermitteln, wo sich die Warteschlange befindet. Anschließend fügt der Warteschlangenmanager der Nachricht einige Routing-Informationen hinzu und stellt sie dann in eine Clusterübertragungswarteschlange.

z/OS Bei einer Clusterwarteschlange kann es sich um eine Warteschlange handeln, die von Mitgliedern einer Gruppe mit gemeinsamer Warteschlange in IBM MQ for z/OS gemeinsam genutzt wird.

## Wird gebunden

Sie können einen Cluster erstellen, in dem mehr als ein Warteschlangenmanager eine Instanz derselben Clusterwarteschlange hostet. Stellen Sie sicher, dass alle Nachrichten in einer Sequenz an die gleiche Instanz der Warteschlange gesendet werden. Sie können eine Reihe von Nachrichten an eine bestimmte Warteschlange binden, indem Sie die Option MQOO_BIND_ON_OPEN im Aufruf MQOPEN verwenden.

## Clusterübertragungswarteschlangen

Ein Warteschlangenmanager kann Nachrichten für andere Warteschlangenmanager in einem Cluster in mehreren Übertragungswarteschlangen speichern. Es gibt zwei Möglichkeiten, einen Warteschlangenmanager so zu konfigurieren, dass er Nachrichten in mehreren Clusterübertragungswarteschlangen speichern kann. Wenn Sie das Warteschlangenmanagerattribut **DEFCLXQ** auf CHANNELsetzen, wird für jeden Clustersenderkanal automatisch eine andere Clusterübertragungswarteschlange aus SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE erstellt. Wenn Sie die Option CLCHNAME für eine Clusterübertragungswarteschlange so setzen, dass sie mit einem oder auch mehreren Clustersenderkanälen übereinstimmt, kann der Warteschlangenmanager in dieser Übertragungswarteschlange Nachrichten für diese Clustersenderkanäle speichern.

**Achtung:** Wenn Sie eine dedizierte SYSTEM.CLUSTER.TRANSMIT.QUEUES -Instanz mit einem Warteschlangenmanager verwenden, für den ein Upgrade von einer früheren Produktversion als

IBM WebSphere MQ 7.5durchgeführt wurde, müssen Sie sicherstellen, dass für SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE die Option SHARE/NOSHARE auf **SHARE** gesetzt ist.

Eine Nachricht für eine Clusterwarteschlange auf einem anderen Warteschlangenmanager wird vor dem Senden in eine Clusterübertragungswarteschlange gestellt. Ein Clustersenderkanal überträgt die Nachrichten von einer Clusterübertragungswarteschlange in Clusterempfängerkanäle auf anderen WS-Managern. Standardmäßig enthält eine systemdefinierte Clusterübertragungswarteschlange alle Nachrichten, die an andere Cluster-WS-Manager übertragen werden sollen. Die Warteschlange wird als SYSTEM.CLUSTER.TRANSMIT.QUEUE bezeichnet. Ein Warteschlangenmanager, der Teil eines Clusters ist, kann Nachrichten in dieser Clusterübertragungswarteschlange an jeden anderen WS-Manager im selben Cluster senden.

Eine Definition für die einzelne SYSTEM.CLUSTER.TRANSMIT.QUEUE -Warteschlange wird standardmäßig auf jedem Warteschlangenmanager außer unter z/OSerstellt. z/OS Unter z/OS kann die Definition mit dem bereitgestellten Beispiel **CSQ4INSX** erstellt werden.

Sie können einen Warteschlangenmanager für die Übertragung von Nachrichten an andere Cluster-WS-Manager mit mehreren Übertragungswarteschlangen konfigurieren. Sie können zusätzliche Clusterübertragungswarteschlangen manuell definieren oder die Warteschlangen vom Warteschlangenmanager automatisch erstellen lassen.

Wenn die Warteschlangen automatisch vom Warteschlangenmanager erstellt werden sollen, ändern Sie das WS-Managerattribut DEFCLXQ von SCTQ in CHANNEL . Das Ergebnis ist, dass der Warteschlangenmanager für jeden zu erstellenden Clustersenderkanal eine einzelne Clusterübertragungswarteschlange erstellt. Die Übertragungswarteschlangen werden als permanente dynamische Warteschlangen aus der Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE erstellt. Der Name jeder permanenten dynamischen Warteschlange lautet SYSTEM.CLUSTER.TRANSMIT. *ChannelName*. Der Name des Clustersenderkanals, dem jede permanente dynamische Clusterübertragungswarteschlange zugeordnet ist, wird im Attribut CLCHNAME der lokalen Übertragungswarteschlange festgelegt. Nachrichten für ferne Cluster-WS-Manager werden in die permanente dynamische Clusterübertragungswarteschlange für den zugehörigen Clustersenderkanal und nicht in SYSTEM.CLUSTER.TRANSMIT.QUEUE gestellt.

Wenn Sie die Clusterübertragungswarteschlangen manuell erstellen möchten, erstellen Sie eine lokale Warteschlange mit dem Attribut USAGE auf XMITQ und das Attribut CLCHNAME auf einen generischen Kanalnamen, der in einen oder mehrere Clustersenderkanäle aufgelöst wird (siehe ClusterChannelName ). Wenn Sie Clusterübertragungswarteschlangen manuell erstellen, haben Sie die Möglichkeit, die Übertragungswarteschlange einem einzelnen Clustersenderkanal oder mehreren Clustersenderkanälen zuzuordnen. Das Attribut CLCHNAME ist ein generischer Name. Dies bedeutet, dass Sie mehrere Platzhalterzeichen ( " * ") in den Namen einfügen können.

Mit Ausnahme der anfänglichen Clustersenderkanäle, die Sie manuell erstellen, um einen WS-Manager mit einem vollständigen Repository zu verbinden, werden die Clustersenderkanäle automatisch erstellt. Sie werden automatisch erstellt, wenn eine Nachricht zum Übertragen an einen Cluster-WS-Manager vorhanden ist. Sie werden mit demselben Namen wie der Name des Clusterempfängerkanals erstellt, der Clusternachrichten für diesen bestimmten Cluster auf dem Zielwarteschlangenmanager empfängt.

Wenn Sie einer Namenskonvention für Clusterempfängerkanäle folgen, ist es möglich, einen generischen Wert für CLCHNAME zu definieren, der verschiedene Arten von Clusternachrichten in verschiedene Übertragungswarteschlangen filtert. Wenn Sie beispielsweise die Namenskonvention für Clusterempfängerkanäle von *ClusterName*. *QmgrName*befolgen, filtert der generische Name *ClusterName*.* Nachrichten für verschiedene Cluster in verschiedenen Übertragungswarteschlangen. Sie müssen die Übertragungswarteschlangen manuell definieren und CLCHNAME in jeder Übertragungswarteschlange auf *ClusterName*.* setzen.

Änderungen an der Zuordnung von Clusterübertragungswarteschlangen zu Clustersenderkanälen werden nicht sofort wirksam. Die momentan zugeordnete Übertragungswarteschlange, die ein Clustersenderkanal verarbeitet, enthält möglicherweise Nachrichten, die gerade vom Clustersenderkanal übertragen werden. Nur wenn keine Nachrichten in der aktuell zugeordneten Übertragungswarteschlange von einem Clustersenderkanal verarbeitet werden, kann der WS-Manager die Zuordnung des Clustersenderkanals zu einer anderen Übertragungswarteschlange ändern. Dies kann entweder auftreten, wenn keine Nachrichten in der Übertragungswarteschlange verbleiben, die vom Clustersenderkanal verarbeitet werden

sollen, oder wenn die Verarbeitung von Nachrichten ausgesetzt wird und der Clustersenderkanal keine " In-Flight " -Nachrichten enthält. In diesem Fall werden alle nicht verarbeiteten Nachrichten für den Clustersenderkanal in die neu zugeordnete Übertragungswarteschlange übertragen, und die Zuordnung der Clustersenderkanäle wird geändert.

Sie können eine Definition einer fernen Warteschlange erstellen, die in eine Clusterübertragungswarteschlange aufgelöst wird. In der Definition befindet sich der WS-Manager QMX in demselben Cluster wie der lokale Warteschlangenmanager, und es gibt keine Übertragungswarteschlange, QMX.

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

Bei der Auflösung von Warteschlangennamen hat die Clusterübertragungswarteschlange Vorrang vor der Standardübertragungswarteschlange. Eine Nachricht, die an A gesendet wird, wird in der Clusterübertragungswarteschlange gespeichert und dann an die ferne Warteschlange B unter QMX gesendet.

WS-Manager können auch mit anderen WS-Managern kommunizieren, die nicht Teil eines Clusters sind. Sie müssen Kanäle und eine Übertragungswarteschlange in derselben Weise wie in einer Umgebung mit verteilter Steuerung von Warteschlangen für den anderen Warteschlangenmanager definieren.

**Anmerkung:** Anwendungen müssen in Warteschlangen schreiben, die in die Clusterübertragungswarteschlange aufgelöst werden, und sie dürfen nicht direkt in die Clusterübertragungswarteschlange schreiben.

## Automatische Definition ferner Warteschlangen

Ein WS-Manager in einem Cluster benötigt keine Definition einer fernen Warteschlange für ferne Warteschlangen im Cluster. Der Clusterwarteschlangenmanager sucht die Position einer fernen Warteschlange aus dem vollständigen Repository. Sie fügt der Nachricht Routing-Informationen hinzu und versetzt sie in die Clusterübertragungswarteschlange. IBM MQ erstellt automatisch eine Definition, die der Definition einer fernen Warteschlange entspricht, so dass die Nachricht gesendet werden kann.

Eine automatisch erstellte Definition einer fernen Warteschlange kann nicht geändert oder gelöscht werden. Wenn Sie jedoch den Befehl DISPLAY QUEUE **runmqsc** mit dem Attribut CLUSINFO verwenden, können Sie alle lokalen Warteschlangen auf einem Warteschlangenmanager sowie alle Clusterwarteschlangen anzeigen, einschließlich Clusterwarteschlangen auf fernen Warteschlangenmanagern. For example:

```
DISPLAY QUEUE(*) CLUSINFO
```

**Zugehörige Konzepte**
Clusterwarteschlangen
Art der zu verwendenden Clusterübertragungswarteschlange auswählen
**Zugehörige Verweise**
ClusterChannelName (MQCHAR20)

### *Mit automatisch definierten Clustersenderkanälen arbeiten*

Nachdem Sie einen Warteschlangenmanager in einen Cluster eingeführt haben, indem Sie seine anfänglichen CLUSSDR- und CLUSRCVR-Definitionen erstellt haben, erstellt IBM MQ automatisch andere Clustersenderkanaldefinitionen, wenn dies erforderlich ist, um Nachrichten in einen anderen Warteschlangenmanager im Cluster zu versetzen. Sie können Informationen zu automatisch definierten Clustersenderkanälen anzeigen, aber Sie können sie nicht ändern. Wenn Sie ihr Verhalten ändern möchten, können Sie einen Exit für die automatische Kanaldefinition verwenden.

## Vorbereitende Schritte

Eine Einführung in automatisch definierte Kanäle finden Sie unter Automatisch definierte Clustersenderkanäle.

## Informationen zu diesem Vorgang

Automatisch definierte Clustersenderkanäle werden vom Cluster als und bei Bedarf erstellt und bleiben aktiv, bis sie unter Verwendung der normalen Unterbrechungsintervallregeln beendet werden.

Clustersenderkanäle (CLUSSDRs) können automatisch definiert werden, um Anwendungsnachrichten und interne Clusterverwaltungsnachrichten zu verschieben. In einem Publish/Subscribe-Cluster (in dem ein Clusterthema definiert wurde) können beispielsweise Kanäle zwischen Teilrepositorys definiert werden, um den Austausch des Proxy-Subskriptionsstatus zu ermöglichen. Wenn nicht erforderlich (inaktiv) für einen längeren Zeitraum werden automatisch definierte CLUSSDRs aus dem Clusterinformationscache eines Teilrepositorys entfernt und sind auf diesem Warteschlangenmanager nicht mehr sichtbar.

Multi Auf Multiplatforms ist dem OAM (Object Authority Manager) nicht bekannt, dass die automatisch definierten Clustersenderkanäle vorhanden sind. Wenn Sie **start**-, **stop**-, **ping**-, **reset**- oder **resolve**-Befehle auf einem automatisch definierten Clustersenderkanal absetzen, prüft der OAM, ob Sie berechtigt sind, die gleiche Aktion für den entsprechenden Clusterempfängerkanal auszuführen.

z/OS Unter z/OS können Sie einen automatisch definierten Clustersenderkanal in der gleichen Weise wie jeden anderen Kanal sichern.

## Prozedur

- Zeigt Informationen zu den automatisch definierten Kanälen für einen bestimmten Clusterwarteschlangenmanager an.

  Mit dem Befehl DISPLAY CHANNEL **runmqsc** können keine automatisch definierten Kanäle angezeigt werden. Verwenden Sie den folgenden Befehl, um die automatisch definierten Kanäle anzuzeigen:

  ```
  DISPLAY CLUSQMGR(qMgrName)
  ```

- Zeigen Sie den Status des automatisch definierten Kanals für einen bestimmten CLUSRCVR an.

  Verwenden Sie den folgenden Befehl, um den Status des automatisch definierten CLUSSDR-Kanals anzuzeigen, der einer von Ihnen erstellten Kanaldefinition CLUSRCVR entspricht:

  ```
  DISPLAY CHSTATUS(channelname)
  ```

- Verwenden Sie einen Exit zur automatischen Kanaldefinition, um das Verhalten eines automatisch definierten Kanals zu ändern.

  Sie können den Exit für die automatische IBM MQ-Kanaldefinition verwenden, wenn Sie ein Benutzerexitprogramm schreiben wollen, um einen Clustersenderkanal oder einen Clusterempfängerkanal anzupassen. Sie können zum Beispiel den Exit für die automatische Kanaldefinition in einer Clusterumgebung verwenden, um die folgenden Änderungen vorzunehmen:

  - Übertragungsdefinitionen, d. L. SNA-LU6.2-Namen,
  - Fügen Sie weitere Exits hinzu oder entfernen Sie sie, z. B. Sicherheitsexits.
  - Ändern Sie die Namen von Kanalexits.

    Der Name des Kanalexits CLUSSDR wird automatisch aus der Kanaldefinition CLUSRCVR generiert und ist daher möglicherweise für Ihre Anforderungen nicht geeignet – insbesondere, wenn die beiden Kanalenden auf unterschiedlichen Plattformen liegen.

    Das Format der Exitnamen ist auf verschiedenen Plattformen unterschiedlich. For example:

    - z/OS Auf der z/OS-Plattform lautet das Format des Parameters SCYEXIT (*Sicherheitsexitname*) SCYEXIT('SECEXIT').
    - Windows Auf Windows-Plattformen hat der Parameter SCYEXIT (*Sicherheitsexitname*) das Format SCYEXIT(' *drive*:*path**library* (secexit)'),

**Anmerkung:** z/OS Wenn kein Exit für die automatische Kanaldefinition vorhanden ist, leitet der z/OS-Warteschlangenmanager den Kanalexitnamen CLUSSDR aus der Kanaldefinition CLUSRCVR auf dem anderen Ende des Kanals ab. Um den Exitnamen von z/OS von einem Nicht-z/OS-Namen abzuleiten, wird der folgende Algorithmus verwendet:

- Exitnamen in Multiplatforms haben die allgemeine Form *path*/*library* (*function*).
- Wenn *function* vorhanden ist, werden bis zu acht Zeichen für diese Zeichen verwendet.
- Andernfalls werden bis zu acht Zeichen von *library* verwendet.

For example:

- `/var/mqm/exits/myExit.so(MsgExit)` konvertiert in `MSGEXIT`
- `/var/mqm/exits/myExit` konvertiert in `MYEXIT`
- `/var/mqm/exits/myExit.so(ExitLongName)` konvertiert in `EXITLONG`

- Wenn der Cluster **PROPCTL** verwenden muss, um Anwendungsheader wie RFH2 aus Nachrichten zu entfernen, die von einem IBM MQ -Warteschlangenmanager an einen Warteschlangenmanager in einer früheren Version des Produkts gesendet werden, müssen Sie einen Exit für automatische Kanaldefinition schreiben, der **PROPCTL** auf den Wert `NONE`setzt.

- Verwenden Sie das Kanalattribut LOCLADDR, um Aspekte der Adressierung zu steuern.
  - Verwenden Sie das Kanalattribut LOCLADDR, um einen abgehenden (TCP-) Kanal für die Verwendung einer bestimmten IP-Adresse, eines bestimmten Ports oder eines bestimmten Portbereichs zu aktivieren. Dies ist nützlich, wenn Sie mehr als eine Netzkarte haben und einen Kanal für die abgehende Kommunikation verwenden möchten.
  - Wenn Sie eine virtuelle IP-Adresse in CLUSSDR-Kanälen angeben möchten, verwenden Sie die IP-Adresse aus dem LOCLADDR-Kanal in einem manuell definierten CLUSSDR. Wenn Sie den Portbereich angeben möchten, verwenden Sie den Portbereich aus dem CLUSRCVR.
  - Wenn ein Cluster LOCLADDR verwenden muss, um die Kanäle für abgehende Kommunikation abzurufen, um eine Bindung an eine bestimmte IP-Adresse zu erhalten, können Sie einen Exit für die automatische Kanaldefinition schreiben, um den Wert für LOCLADDR in einen beliebigen automatisch definierten CLUSSDR-Kanal zu setzen. Sie müssen sie auch in dem manuell definierten CLUSSDR-Kanal angeben.
  - Geben Sie eine Portnummer oder einen Portbereich in LOCLADDR eines CLUSRCVR-Kanals an, wenn Sie möchten, dass alle Warteschlangenmanager in einem Cluster einen bestimmten Port oder einen bestimmten Portbereich für die gesamte abgehende Kommunikation verwenden.

  **Anmerkung:** Geben Sie im Feld LOCLADDR des Kanals CLUSRCVR keine IP-Adresse an, es sei denn, alle Warteschlangenmanager befinden sich auf demselben Server. Die IP-Adresse LOCLADDR wird an die automatisch definierten CLUSSDR-Kanäle aller Warteschlangenmanager weitergegeben, die über den Kanal CLUSRCVR verbunden sind.

  Multi Unter Multiplatforms können Sie einen lokalen Standardwert für die lokale Adresse festlegen, der für alle Senderkanäle verwendet wird, für die keine lokale Adresse definiert ist. Der Standardwert wird definiert, indem die Umgebungsvariable MQ_LCLADDR vor dem Starten des Warteschlangenmanagers festgelegt wird. Das Format des Werts stimmt mit dem des MQSC-Attributs LOCLADDR überein.

**Zugehörige Verweise**

Lokale Adresse (LOCLADDR)

## *Mit Standardclusterobjekten arbeiten*

Sie können die Standardkanaldefinitionen auf dieselbe Weise wie jede andere Kanaldefinition durch die Ausführung von WebSphere MQ-Scriptbefehlen oder PCF-Befehlen ändern. Ändern Sie die Standardwarteschlangendefinitionen mit Ausnahme von `SYSTEM.CLUSTER.HISTORY.QUEUE` nicht.

Eine vollständige Liste dieser Objekte finden Sie unter Standardclusterobjekte. Die folgende Liste enthält nur die Objekte, die Sie ändern können.

**SYSTEM.CLUSTER.HISTORY.QUEUE**
Jeder Warteschlangenmanager in einem Cluster verfügt über eine lokale Warteschlange namens SYSTEM.CLUSTER.HISTORY.QUEUE. SYSTEM.CLUSTER.HISTORY.QUEUE wird verwendet, um das Protokoll der Clusterstatusinformationen für Servicezwecke zu speichern.

In den Standardobjekteinstellungen ist SYSTEM.CLUSTER.HISTORY.QUEUE auf PUT ( ENABLED) gesetzt. Um die Erfassung von Protokolldaten zu unterdrücken, ändern Sie die Einstellung in PUT ( DISABLED).

**SYSTEM.CLUSTER.TRANSMIT.QUEUE**
Jeder Warteschlangenmanager verfügt über eine Definition für eine lokale Warteschlange namens SYSTEM.CLUSTER.TRANSMIT.QUEUE. SYSTEM.CLUSTER.TRANSMIT.QUEUE ist die Standardübertragungswarteschlange für alle Nachrichten an alle Warteschlangen und Warteschlangenmanager innerhalb von Clustern. Sie können die Standardübertragungswarteschlange für jeden Clustersenderkanal in SYSTEM.CLUSTER.TRANSMIT.*ChannelName* ändern, indem Sie das Warteschlangenmanagerattribut DEFXMITQ z/OS, außer unter z/OS ändern. Sie können SYSTEM.CLUSTER.TRANSMIT.QUEUE nicht löschen. Es wird auch verwendet, um Berechtigungsprüfungen zu definieren, ob die verwendete Standardübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE oder SYSTEM.CLUSTER.TRANSMIT.*ChannelName* ist.

**Zugehörige Konzepte**
Standardclusterobjekte

## *Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen*

Nachrichten zwischen Clustering-WS-Managern werden in Clusterübertragungswarteschlangen gespeichert und von Clustersenderkanälen weitergeleitet. Zu einem beliebigen Zeitpunkt ist ein Clustersenderkanal einer Übertragungswarteschlange zugeordnet. Wenn Sie die Konfiguration des Kanals ändern, kann es beim nächsten Start zu einer anderen Übertragungswarteschlange wechseln. Die Verarbeitung dieses Switches ist automatisiert und transaktionsorientiert.

Führen Sie den folgenden MQSC-Befehl aus, um die Übertragungswarteschlangen anzuzeigen, denen Clustersenderkanäle zugeordnet sind:

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)          CHLTYPE(CLUSSDR)
CONNAME(9.146.163.190(1416))    CURRENT
RQMNAME(QM2)             STATUS(STOPPED)
SUBSTATE( )            XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

Die Übertragungswarteschlange, die im gespeicherten Kanalstatus eines gestoppten Clustersenderkanals angezeigt wird, kann sich ändern, wenn der Kanal erneut gestartet wird. In „Auswahl von Standardübertragungswarteschlangen nach Clustersenderkanälen” auf Seite 319 wird der Prozess der Auswahl einer Standardübertragungswarteschlange beschrieben; „Auswahl von manuell definierten Übertragungswarteschlangen nach Clustersenderkanälen” auf Seite 320 beschreibt den Prozess der Auswahl einer manuell definierten Übertragungswarteschlange.

Wenn ein Clustersenderkanal gestartet wird, überprüft er seine Zuordnung zu Übertragungswarteschlangen. Wenn die Konfiguration von Übertragungswarteschlangen oder die Standardwerte des Warteschlangenmanagers geändert werden, kann er den Kanal möglicherweise mit einer anderen Übertragungswarteschlange verknüpft werden. Wenn der Kanal aufgrund einer Konfigurationsänderung mit einer anderen Übertragungswarteschlange neu gestartet wird, findet ein Prozess der Übertragung von Nachrichten an die neu zugeordnete Übertragungswarteschlange statt. In „Funktionsweise des Prozesses zum Wechseln des Clustersenderkanals in eine andere Übertragungswarteschlange” auf Seite 321 wird der Prozess der Übertragung eines Clustersenderkanals von einer Übertragungswarteschlange in eine andere beschrieben.

Das Verhalten von Clustersenderkanälen unterscheidet sich von Sender-und Serverkanälen. Sie bleiben der gleichen Übertragungswarteschlange zugeordnet, bis das Kanalattribut **XMITQ** geändert wird. Wenn

Sie das Attribut für die Übertragungswarteschlange auf einem Sender-oder Serverkanal ändern und ihn erneut starten, werden die Nachrichten nicht von der alten Übertragungswarteschlange in die neue übertragen.

Ein weiterer Unterschied zwischen Clustersenderkanälen und Sender-oder Serverkanälen besteht darin, dass mehrere Clustersenderkanäle eine Clusterübertragungswarteschlange öffnen können, aber nur ein Sender-oder Serverkanal eine normale Übertragungswarteschlange öffnen kann. Sie haben die Möglichkeit, dass Clustersenderkanäle Übertragungswarteschlangen nicht gemeinsam nutzen. Die Exklusivität wird nicht erzwungen; sie ist ein Ergebnis der Konfiguration. Sie können den Pfad einer Nachricht in einem Cluster so konfigurieren, dass er keine Übertragungswarteschlangen oder Kanäle mit Nachrichten gemeinsam nutzt, die zwischen anderen Anwendungen fließen. Siehe Clustering: Konfiguration von Clusterübertragungswarteschlangen planen und „Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden" auf Seite 376.

## Auswahl von Standardübertragungswarteschlangen nach Clustersenderkanälen

Eine Clusterübertragungswarteschlange ist entweder eine Systemstandardwarteschlange mit einem Namen, der mit SYSTEM.CLUSTER.TRANSMIT beginnt, oder eine manuell definierte Warteschlange. Ein Clustersenderkanal wird einer Clusterübertragungswarteschlange auf eine der beiden folgenden Arten zugeordnet: durch den Standardmechanismus der Clusterübertragungswarteschlange oder durch manuelle Konfiguration.

Die standardmäßige Clusterübertragungswarteschlange wird als WS-Manager-Attribut **DEFCLXQ** festgelegt. Der Wert ist entweder SCTQ oder CHANNEL . Neue und migrierte WS-Manager werden auf SCTQ gesetzt. Sie können den Wert in CHANNEL ändern.

Wenn SCTQ festgelegt ist, ist die Standard-Cluster-Übertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE. Jeder Clustersenderkanal kann diese Warteschlange öffnen. Die Clustersenderkanäle, die die Warteschlange öffnen, sind diejenigen, die nicht mit manuell definierten Clusterübertragungswarteschlangen verknüpft sind.

Wenn CHANNEL festgelegt ist, kann der Warteschlangenmanager für jeden Clustersenderkanal eine separate permanente dynamische Übertragungswarteschlange erstellen. Jede Warteschlange hat den Namen SYSTEM.CLUSTER.TRANSMIT. *ChannelName* und wird aus der Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUEerstellt. Jedem Clustersenderkanal, der keiner manuell definierten Clusterübertragungswarteschlange zugeordnet ist, wird eine permanentdynamische Clusterübertragungswarteschlange zugeordnet. Die Warteschlange wird vom Warteschlangenmanager erstellt, wenn er eine separate Clusterübertragungswarteschlange für das Clusterziel benötigt, das von diesem Clustersenderkanal bedient wird, und es ist keine Warteschlange vorhanden.

Einige Clusterziele können von Clustersenderkanälen, die manuell definierten Übertragungswarteschlangen zugeordnet sind, und anderen durch die Standardwarteschlange oder-warteschlangen bedient werden. In der Zuordnung von Clustersenderkanälen mit Übertragungswarteschlangen haben die manuell definierten Übertragungswarteschlangen immer Vorrang vor den Standardübertragungswarteschlangen.

Die Rangfolge der Clusterübertragungswarteschlangen ist in Abbildung 37 auf Seite 320 dargestellt. Der einzige Clustersenderkanal, der keiner manuell definierten Clusterübertragungswarteschlange zugeordnet ist, ist CS.QM1. Sie ist keiner manuell definierten Übertragungswarteschlange zugeordnet, da keiner der Kanalnamen im Attribut **CLCHNAME** der Übertragungswarteschlangen mit CS.QM1 übereinstimmt.

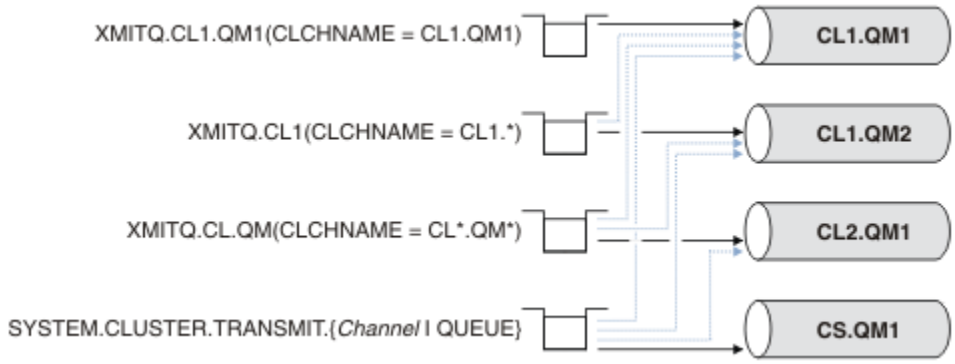



*Abbildung 37. Übertragungswarteschlange/Clustersenderkanalvorrangstellung*

## Auswahl von manuell definierten Übertragungswarteschlangen nach Clustersenderkanälen

Eine manuell definierte Warteschlange hat das Attribut **USAGE** für die Übertragungswarteschlange, das auf XMITQ gesetzt ist, und das Attribut "Clusterkanalname" **CLCHNAME** auf einen bestimmten oder generischen Kanalnamen gesetzt.

Wenn der Name im Warteschlangenattribut von **CLCHNAME** mit einem Clustersenderkanalnamen übereinstimmt, wird der Kanal der Warteschlange zugeordnet. Der Name ist entweder eine exakte Übereinstimmung, wenn der Name keine Platzhalterzeichen enthält, oder es ist die beste Übereinstimmung, wenn der Name Platzhalterzeichen enthält.

Wenn **CLCHNAME** -Definitionen in mehreren Übertragungswarteschlangen mit demselben Clustersenderkanal übereinstimmen, werden die Definitionen als überlappend definiert. Um die Mehrdeutigkeit zu beheben, gibt es eine Rangfolge zwischen Übereinstimmungen. Exakte Übereinstimmungen haben immer Vorrang. Abbildung 37 auf Seite 320 zeigt Zuordnungen zwischen Übertragungswarteschlangen und Clustersenderkanälen. Die schwarzen Pfeile zeigen die tatsächlichen Assoziationen und die grauen Pfeile, mögliche Assoziationen an. Die Prioritätsreihenfolge der Übertragungswarteschlangen in Abbildung 37 auf Seite 320 lautet:

**XMITQ.CL1.QM1**
: Für die Übertragungswarteschlange XMITQ.CL1.QM1 ist das Attribut **CLCHNAME** auf CL1.QM1 festgelegt. Die Definition des Attributs **CLCHNAME** , CL1.QM1, weist keine Platzhalterzeichen auf und hat Vorrang vor allen anderen CLCHNAME -Attributen, die in anderen Übertragungswarteschlangen definiert sind, die mit Platzhalterzeichen übereinstimmen. Der WS-Manager speichert alle Clusternachrichten, die vom CL1.QM1 -Clustersenderkanal in der Übertragungswarteschlange von XMITQ.CL1.QM1 übertragen werden sollen. Die einzige Ausnahme ist, wenn für mehrere Übertragungswarteschlangen das Attribut **CLCHNAME** auf CL1.QM1 gesetzt ist. In diesem Fall speichert der Warteschlangenmanager Nachrichten für den Clustersenderkanal von CL1.QM1 in jeder dieser Warteschlangen. Er wählt eine Warteschlange willkürlich aus, wenn der Kanal gestartet wird. Es kann eine andere Warteschlange auswählen, wenn der Kanal erneut gestartet wird.

**XMITQ.CL1**
: Für die Übertragungswarteschlange XMITQ.CL1 ist das Attribut **CLCHNAME** auf CL1.* festgelegt. Die Definition des Attributs **CLCHNAME** ( CL1.*) weist ein abschließendes Platzhalterzeichen auf, das mit dem Namen eines beliebigen Clustersenderkanals übereinstimmt, der mit CL1. beginnt. Der Warteschlangenmanager speichert jede Clusternachricht, die von einem beliebigen Clustersenderkanal übertragen werden soll, dessen Name mit CL1. in der Übertragungswarteschlange XMITQ.CL1 beginnt, es sei denn, es gibt eine Übertragungswarteschlange mit einer spezifiseren Übereinstimmung, wie z. B. die Warteschlange XMITQ.CL1.QM1. Ein abschließendes Platzhalterzeichen macht die Definition weniger spezifisch als eine Definition ohne Platzhalterzeichen und genauer als eine Definition mit mehreren Platzhaltern oder Platzhalterzeichen, auf die weitere abschließende Zeichen folgen.

**XMITQ.CL.QM**
: XMITQ.CL.QM ist der Name der Übertragungswarteschlange, in der das Attribut **CLCHNAME** auf CL*.QM* gesetzt ist. Die Definition von CL*.QM* weist zwei Platzhalterzeichen auf, die mit dem

Namen eines beliebigen Clustersenderkanals übereinstimmen, der mit `CL.` beginnt, und enthält oder endet mit `QM`. Die Übereinstimmung ist weniger spezifisch als eine Übereinstimmung mit einem Platzhalterzeichen.

**SYSTEM.CLUSTER.TRANSMIT. *channelName*|QUEUE**
Wenn für keine Übertragungswarteschlange ein Attribut **CLCHNAME** vorhanden ist, das mit dem Namen des Clustersenderkanals übereinstimmt, den der Warteschlangenmanager verwenden soll, verwendet der Warteschlangenmanager die Standardwarteschlange für Clusterübertragungen. Die Standardclusterübertragungswarteschlange ist entweder die Einzelsystemclusterübertragungswarteschlange `SYSTEM.CLUSTER.TRANSMIT.QUEUE`oder eine Systemclusterübertragungswarteschlange, die der Warteschlangenmanager für einen bestimmten Clustersenderkanal `SYSTEM.CLUSTER.TRANSMIT.` *channelName*erstellt hat. Welche Warteschlange der Standardwert ist, hängt von der Einstellung des Attributs des Warteschlangenmanagers **DEFXMITQ** ab.

**Tipp:** Sofern Sie keinen eindeutigen Bedarf an überlappenden Definitionen haben, vermeiden Sie sie, da sie zu komplizierten Konfigurationen führen können, die schwer zu verstehen sind.

## Funktionsweise des Prozesses zum Wechseln des Clustersenderkanals in eine andere Übertragungswarteschlange

Wenn Sie die Zuordnung von Clustersenderkanälen zu Clusterübertragungswarteschlangen ändern möchten, ändern Sie den Parameter **CLCHNAME** einer beliebigen Übertragungswarteschlange oder des Warteschlangenmanagerparameters **DEFCLXQ** jederzeit. Nichts passiert sofort. Änderungen treten nur auf, wenn ein Kanal gestartet wird. Wenn er gestartet wird, prüft er, ob Nachrichten aus derselben Übertragungswarteschlange weitergesendet werden. Drei Arten von Änderungen ändern die Zuordnung eines Clustersenderkanals mit einer Übertragungswarteschlange.

1. Wenn der Parameter **CLCHNAME** der Übertragungswarteschlange neu definiert wird, wird der Clustersenderkanal momentan weniger spezifisch oder leer sein, oder die Clusterübertragungswarteschlange wird gelöscht, wenn der Kanal gestoppt wird.

   Eine andere Clusterübertragungswarteschlange könnte jetzt eine bessere Übereinstimmung für den Kanalnamen sein. Oder, wenn keine anderen Übertragungswarteschlangen mit dem Namen des Clustersenderkanals übereinstimmen, muss die Zuordnung auf die Standardübertragungswarteschlange zurückgesetzt werden.
2. Den Parameter **CLCHNAME** einer anderen Clusterübertragungswarteschlange neu definieren oder eine Clusterübertragungswarteschlange hinzufügen.

   Der Parameter **CLCHNAME** einer anderen Übertragungswarteschlange könnte jetzt eine bessere Übereinstimmung für den Clustersenderkanal sein als die Übertragungswarteschlange, der der Clustersenderkanal momentan zugeordnet ist. Wenn der Clustersenderkanal momentan einer Standard-Cluster-Übertragungswarteschlange zugeordnet ist, kann er einer manuell definierten Clusterübertragungswarteschlange zugeordnet werden.
3. Wenn der Clustersenderkanal momentan einer Standard-Cluster-Übertragungswarteschlange zugeordnet ist, ändern Sie den Parameter des **DEFCLXQ** -Warteschlangenmanagers.

Wenn sich die Zuordnung eines Clustersenderkanals ändert, wechselt der Kanal, wenn der Kanal gestartet wird, seine Zuordnung zu der neuen Übertragungswarteschlange. Während des Switchs wird sichergestellt, dass keine Nachrichten verloren gehen. Nachrichten werden in die neue Übertragungswarteschlange in der Reihenfolge übertragen, in der der Kanal die Nachrichten an den fernen WS-Manager übertragen würde.

**Hinweis:** Bei der gemeinsamen Weiterleitung von Nachrichten in einem Cluster müssen Sie Nachrichten in Gruppen einordnen, um sicherzustellen, dass Nachrichten, die in der Reihenfolge zugestellt werden müssen, in der Reihenfolge zugestellt werden. In seltenen Fällen können Nachrichten in einem Cluster nicht mehr in Ordnung sein.

Der Switchprozess durchläuft die folgenden transaktionsorientierten Schritte. Wenn der Switch-Prozess unterbrochen wird, wird der aktuelle transaktionsorientierte Schritt wieder aufgenommen, wenn der Kanal erneut gestartet wird.

**Schritt 1-Prozessnachrichten aus der ursprünglichen Übertragungswarteschlange**

Der Clustersenderkanal ist der neuen Übertragungswarteschlange zugeordnet, die er möglicherweise mit anderen Clustersenderkanälen gemeinsam nutzen kann. Nachrichten für den Clustersenderkanal werden weiterhin in die ursprüngliche Übertragungswarteschlange gestellt. Ein Übergangsschalterprozess überträgt Nachrichten aus der ursprünglichen Übertragungswarteschlange in die neue Übertragungswarteschlange. Der Clustersenderkanal leitet die Nachrichten von der neuen Übertragungswarteschlange an den Clusterempfängerkanal weiter. Der Kanalstatus zeigt den Clustersenderkanal an, der immer noch der alten Übertragungswarteschlange zugeordnet ist.

Der Switch-Prozess überträgt auch weiterhin neu eingekommenes Nachrichten. Dieser Schritt wird fortgesetzt, bis die Anzahl der verbleibenden Nachrichten, die durch den Schaltvorgang weitergeleitet werden sollen, den Wert Null erreicht. Wenn die Anzahl der Nachrichten null erreicht, wird die Prozedur in Schritt 2 verschoben.

Während Schritt 1 wird die Plattenaktivität für den Kanal erhöht. Persistente Nachrichten werden von der ersten Übertragungswarteschlange und von der zweiten Übertragungswarteschlange aus festgeschrieben. Diese Plattenaktivität wird zusätzlich zu den Nachrichten, die festgeschrieben werden, wenn sie in die Übertragungswarteschlange gestellt und aus der Übertragungswarteschlange entfernt werden, als Teil der normalen Übertragung der Nachrichten festgeschrieben. Im Idealfall kommen während des Schaltvorganges keine Meldungen ein, so daß der Übergang so schnell wie möglich erfolgen kann. Wenn Nachrichten ankommen, werden sie vom Switch-Prozess verarbeitet.

**Schritt 2-Prozessnachrichten aus der neuen Übertragungswarteschlange**

Sobald keine Nachrichten in der ursprünglichen Übertragungswarteschlange für den Clustersenderkanal verbleiben, werden neue Nachrichten direkt in die neue Übertragungswarteschlange gestellt. Der Kanalstatus zeigt den Clustersenderkanal an, der der neuen Übertragungswarteschlange zugeordnet ist. Die folgende Nachricht wird in das Fehlerprotokoll des Warteschlangenmanagers geschrieben: " AMQ7341 `Die Übertragungswarteschlange für Kanal` *`Kanalname`* `ist` *`Warteschlangename`* ."

## Mehrere Clusterübertragungswarteschlangen und Clusterübertragungswarteschlangen-Attribute

Es besteht die Möglichkeit, Clusternachrichten an verschiedene Warteschlangenmanager weiterzuleiten, die die Nachrichten in einer einzelnen Clusterübertragungswarteschlange oder in mehreren Warteschlangen speichern. Bei einer Warteschlange verfügen Sie über eine Gruppe von Attributen für Clusterübertragungswarteschlangen zum Festlegen und Abfragen von; bei mehreren Warteschlangen, haben Sie mehrere Gruppen. Bei einigen Attributen ist die Verwendung mehrerer Gruppen ein Vorteil: z. B. die Warteschlangenlänge abfragen, wie viele Nachrichten darauf warten, von einem oder einer Gruppe von Kanälen weitergeleitet zu werden, und nicht von allen Kanälen. Für andere Attribute ist die Verwendung mehrerer Gruppen ein Nachteil: Sie möchten beispielsweise wahrscheinlich nicht dieselben Zugriffsberechtigungen für jede Clusterübertragungswarteschlange konfigurieren. Aus diesem Grund werden Zugriffsberechtigungen immer auf das Profil für `SYSTEM.CLUSTER.TRANSMIT.QUEUE` und nicht auf Profile für eine bestimmte Clusterübertragungswarteschlange überprüft. Wenn Sie differenziertere Sicherheitsprüfungen anwenden möchten, finden Sie weitere Informationen im Abschnitt Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen .

## Mehrere Clustersenderkanäle und mehrere Übertragungswarteschlangen

Ein WS-Manager speichert eine Nachricht in einer Clusterübertragungswarteschlange, bevor er sie an einen Clustersenderkanal weiterleitet. Er wählt einen Clustersenderkanal aus, der mit dem Ziel für die Nachricht verbunden ist. Es kann eine Auswahl an Clustersenderkanälen haben, die alle mit derselben Zieladresse verbunden sind. Das Ziel kann die gleiche physische Warteschlange sein, die über mehrere Clustersenderkanäle mit einem einzigen Warteschlangenmanager verbunden ist. Die Zieladresse kann auch viele physische Warteschlangen mit demselben Warteschlangennamen sein, die sich auf verschiedenen Warteschlangenmanagern in demselben Cluster befinden. Wenn es eine Auswahl an Clustersenderkanälen gibt, die mit einem Ziel verbunden sind, wählt der Algorithmus für die Lastverteilung eine

Auswahl aus. Die Auswahl hängt von einer Reihe von Faktoren ab; siehe Algorithmus für Clusterauslastungsmanagement .

In Abbildung 38 auf Seite 324 sind CL1.QM1, CL1.QM2 und CS.QM1 alle Kanäle, die zu demselben Ziel führen können. Wenn Sie beispielsweise Q1 in CL1 auf QM1 und QM2 definieren, stellen CL1.QM1 und CL1.QM2 beide Routen zu demselben Ziel, Q1, auf zwei verschiedenen Warteschlangenmanagern bereit. Wenn sich der Kanal CS.QM1 auch in CL1 befindet, ist es auch ein Kanal, den eine Nachricht für Q1 annehmen kann. Die Clusterzugehörigkeit von CS.QM1 kann durch eine Clusternamensliste definiert werden. Aus diesem Grund enthält der Kanalname keinen Clusternamen in seiner Konstruktion. Abhängig von den Parametern für den Lastausgleich und der sendenden Anwendung werden möglicherweise einige Nachrichten für Q1 in jeder der Übertragungswarteschlangen, XMITQ.CL1.QM1, XMITQ.CL1 und SYSTEM.CLUSTER.TRANSMIT.CS.QM1, platziert.

Wenn Sie den Nachrichtenverkehr voneinander trennen möchten, sodass Nachrichten für dasselbe Ziel keine Warteschlangen oder Kanäle mit Nachrichten für verschiedene Ziele gemeinsam nutzen, müssen Sie überlegen, wie der Datenverkehr auf verschiedene Clustersenderkanäle aufgeteilt werden soll, und wie die Nachrichten für einen bestimmten Kanal in eine andere Übertragungswarteschlange aufgeteilt werden. Clusterwarteschlangen in demselben Cluster, auf demselben Warteschlangenmanager, verwenden normalerweise dieselben Clusterkanäle. Das Definieren mehrerer Cluster-Übertragungswarteschlangen allein reicht nicht aus, um den Clusternachrichtenverkehr in verschiedene Warteschlangen zu trennen. Wenn Sie Nachrichten nicht für verschiedene Zielwarteschlangen auf unterschiedlichen Kanälen voneinander trennen, verwenden die Nachrichten dieselbe Clusterübertragungswarteschlange.

Eine einfache Möglichkeit, die Kanäle zu trennen, die Nachrichten annehmen, besteht darin, mehrere Cluster zu erstellen. Definieren Sie auf jedem WS-Manager in jedem Cluster nur eine Clusterwarteschlange. Wenn Sie dann einen anderen Clusterempfängerkanal für jede Cluster-/WS-Manager-Kombination definieren, verwenden die Nachrichten für jede Clusterwarteschlange keinen Clusterkanal mit Nachrichten für andere Clusterwarteschlangen. Wenn Sie separate Übertragungswarteschlangen für die Clusterkanäle definieren, speichert der sendende Warteschlangenmanager Nachrichten für nur eine Clusterwarteschlange in jeder Übertragungswarteschlange. Wenn beispielsweise zwei Clusterwarteschlangen Ressourcen nicht gemeinsam nutzen sollen, können Sie sie entweder in verschiedenen Clustern desselben Warteschlangenmanagers oder auf verschiedenen Warteschlangenmanagern in demselben Cluster platzieren.

Die Auswahl der Clusterübertragungswarteschlange wirkt sich nicht auf den Algorithmus für die Lastverteilung aus. Der Algorithmus für den Lastausgleich wählt den Clustersenderkanal aus, um eine Nachricht weiterzuleiten. Sie stellt die Nachricht in die Übertragungswarteschlange, die von diesem Kanal bedient wird. Wenn der Algorithmus für den Lastausgleich erneut aufgerufen werden soll, z. B. wenn der Kanal gestoppt wird, kann er möglicherweise einen anderen Kanal auswählen, um die Nachricht weiterzuleiten. Wenn er einen anderen Kanal wählt und der neue Kanal Nachrichten aus einer anderen Clusterübertragungswarteschlange weiterleitet, überträgt der Lastausgleichsalgorithmus die Nachricht an die andere Übertragungswarteschlange.

In Abbildung 38 auf Seite 324 werden zwei Clustersenderkanäle, CS.QM1 und CS.QM2, der Standardsystemübertragungswarteschlange zugeordnet. Wenn der Lastausgleichsalgorithmus eine Nachricht in SYSTEM.CLUSTER.TRANSMIT.QUEUE oder einer anderen Clusterübertragungswarteschlange speichert, wird der Name des Clustersenderkanals, der die Nachricht weiterleiten soll, in der Korrelations-ID der Nachricht gespeichert. Jeder Kanal leitet nur die Nachrichten weiter, die mit der Korrelations-ID mit dem Kanalnamen übereinstimmen.

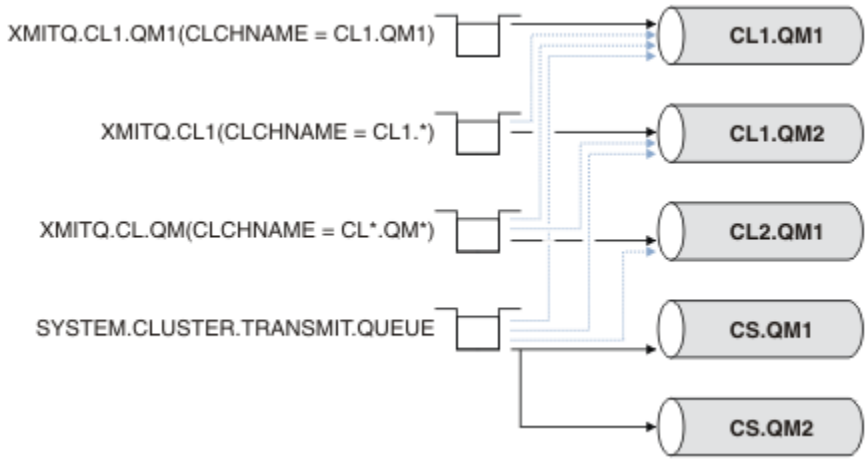



*Abbildung 38. Mehrere Clustersenderkanäle*

Wenn `CS.QM1` stoppt, werden die Nachrichten in der Übertragungswarteschlange für diesen Clustersenderkanal überprüft. Die Nachrichten, die von einem anderen Kanal weitergeleitet werden können, werden durch den Algorithmus für die Lastverteilung erneut verarbeitet. Ihre Korrelations-ID wird auf einen alternativen Clustersenderkanalnamen zurückgesetzt. Wenn es sich bei dem alternativen Clustersenderkanal um `CS.QM2` handelt, bleibt die Nachricht in `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Wenn der alternative Kanal `CL1.QM1` ist, überträgt der Lastausgleichsalgorithmus die Nachricht an `XMITQ.CL1.QM1`. Wenn der Clustersenderkanal erneut gestartet wird, werden neue Nachrichten und Nachrichten, die nicht für einen anderen Clustersenderkanal markiert wurden, durch den Kanal erneut übertragen.

Sie können die Zuordnung zwischen Übertragungswarteschlangen und Clustersenderkanälen auf einem aktiven System ändern. Sie können einen **CLCHNAME** -Parameter in einer Übertragungswarteschlange ändern oder den Parameter des Warteschlangenmanagers von **DEFCLXQ** ändern. Wenn ein Kanal, der von der Änderung betroffen ist, erneut gestartet wird, startet er den Übertragungswarteschlangen-Switching-Prozess (siehe „Funktionsweise des Prozesses zum Wechseln des Clustersenderkanals in eine andere Übertragungswarteschlange” auf Seite 321).

Der Prozess zum Umschalten der Übertragungswarteschlange beginnt, wenn der Kanal erneut gestartet wird. Der Prozess zur Neuverteilung von Workloads wird gestartet, wenn der Kanal gestoppt wird. Die beiden Prozesse können parallel ausgeführt werden.

Der einfache Fall ist, dass beim Stoppen eines Clustersenderkanals der Prozess des Neuausgleichs nicht dazu führt, dass der Clustersenderkanal geändert wird, um Nachrichten in der Warteschlange weiterzuleiten. Dies ist der Fall, wenn kein anderer Clustersenderkanal die Nachrichten an die richtige Zieladresse weiterleiten kann. Wenn kein alternativer Clustersenderkanal die Nachrichten an ihre Zieladresse weiterleiten soll, bleiben die Nachrichten für denselben Clustersenderkanal markiert, nachdem der Clustersenderkanal gestoppt wurde. Wenn der Kanal ansteht, wenn ein Switch ansteht, werden die Nachrichten von den Vermittlungsprozessen in eine andere Übertragungswarteschlange verschoben, in der sie von demselben Clustersenderkanal verarbeitet werden.

Der komplexere Fall ist, wenn mehr als ein Clustersenderkanal einige Nachrichten an ein und dasselbe Ziel verarbeiten kann. Sie können den Clustersenderkanal stoppen und erneut starten, um den Übertragungswarteschlangenschalter auszulösen. In vielen Fällen hat der Algorithmus für den Lastausgleich bereits Nachrichten aus der ursprünglichen Übertragungswarteschlange in verschiedene Übertragungswarteschlangen verschoben, die von verschiedenen Clustersenderkanälen bedient werden. Nur die Nachrichten, die nicht von einem anderen Clustersenderkanal weitergeleitet werden können, bleiben in die neue Übertragungswarteschlange übertragen. In einigen Fällen, wenn der Kanal schnell erneut gestartet wird, bleiben einige Nachrichten, die durch den Algorithmus für den Lastausgleich übertragen werden könnten, bestehen. In diesem Fall werden einige der verbleibenden Nachrichten durch den Lastausgleichsprozess und einige durch den Prozess zum Umschalten der Übertragungswarteschlange umgeschaltet.

**Zugehörige Konzepte**

Clusterkanäle

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

„Berechnen der Größe des Protokolls” auf Seite 698
Schätzen der Größe des Protokollwarteschlangenmanagers.

**Zugehörige Tasks**

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

„Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager” auf Seite 364
Befolgen Sie die Anweisungen in der Task, um überlappende Cluster mit einem Gateway-Warteschlangenmanager zu erstellen. Verwenden Sie die Cluster als Ausgangspunkt für die folgenden Beispiele, um Nachrichten in einer Anwendung von Nachrichten an andere Anwendungen in einem Cluster zu isolieren.

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

„Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde” auf Seite 373
Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine zusätzliche Clusterübertragungswarteschlange, um den Nachrichtenverkehr auf einen einzelnen Warteschlangenmanager in einem Cluster zu trennen.

„Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden” auf Seite 376
Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet einen zusätzlichen Cluster, um die Nachrichten in einer bestimmten Clusterwarteschlange zu isolieren.

## Kommunikation in einem Cluster einrichten

Ein Kanalinitiator wird benötigt, um einen Kommunikationskanal zu starten, wenn eine Nachricht ausgegeben werden soll. Ein Kanallistener wartet, bis das andere Ende eines Kanals gestartet wird, um die Nachricht zu empfangen.

### Vorbereitende Schritte

Wenn Sie die Kommunikation zwischen Warteschlangenmanagern in einem Cluster herstellen möchten, konfigurieren Sie einen Link mit einem der unterstützten Übertragungsprotokolle. Folgende Protokolle werden unterstützt:

- TCP oder LU 6.2 auf einer beliebigen Plattform
- Windows NetBIOS oder SPX auf Windows-Systemen

Im Rahmen dieser Konfiguration benötigen Sie außerdem Kanalinitiatoren und Kanallistener, wie Sie es mit der verteilten Steuerung von Warteschlangen tun.

### Informationen zu diesem Vorgang

Alle Clusterwarteschlangenmanager benötigen einen Kanalinitiator, um die systemdefinierte Initialisierungswarteschlange `SYSTEM.CHANNEL.INITQ` zu überwachen. `SYSTEM.CHANNEL.INITQ` ist die Initialisierungswarteschlange für alle Übertragungswarteschlangen, einschließlich der Clusterübertragungswarteschlange.

Jeder WS-Manager muss über einen Kanallistener verfügen. Ein Kanal-Listener-Programm wartet auf eingehende Netzanforderungen und startet den entsprechenden Empfängerkanal, wenn er benötigt wird.

Die Implementierung von Channel-Listenern ist plattformspezifisch, es gibt jedoch einige allgemeine Funktionen.

Auf allen IBM MQ-Plattformen kann der Listener mit dem Befehl **START LISTENER** gestartet werden.

Multi Unter "Multiplatforms" können Sie den Listener automatisch zur gleichen Zeit wie den Warteschlangenmanager starten. Wenn Sie den Listener automatisch starten möchten, setzen Sie das Attribut CONTROL des Objekts LISTENER auf QMGR oder STARTONLY .

z/OS Ein nicht gemeinsam genutzter Listener-Port (INDISP (QMGR)) muss für CLUSRCVR-Kanäle in z/OS und für CLUSSDR-Kanäle zu z/OS verwendet werden.

## Vorgehensweise

1. Starten Sie den Kanalinitiator.
   - z/OS Unter z/OS gibt es für jeden Warteschlangenmanager je einen Kanalinitiator, der als separater Adressraum ausgeführt wird. Sie starten ihn mit dem Befehl **MQSC** START CHINIT , den Sie beim Start des Warteschlangenmanagers ausgeben.
   - ALW Unter AIX, Linux, and Windows wird beim Starten eines Warteschlangenmanagers automatisch ein Kanalinitiator gestartet, wenn das Warteschlangenmanager-Attribut SCHINIT auf QMGR gesetzt ist. Andernfalls kann er mit dem Befehl **runmqsc** START CHINIT oder dem Steuerbefehl **runmqchi** gestartet werden.
   - IBM i Unter IBM i wird beim Starten eines Warteschlangenmanagers automatisch ein Kanalinitiator gestartet, wenn das Warteschlangenmanager-Attribut SCHINIT auf QMGR gesetzt ist. Andernfalls kann er mit dem Befehl **runmqsc** START CHINIT oder dem Steuerbefehl **runmqchi** gestartet werden.
2. Starten Sie den Kanal-Listener.
   - z/OS Verwenden Sie unter z/OS das von IBM MQ bereitgestellte Kanallistenerprogramm. Wenn Sie einen IBM MQ-Kanallistener starten möchten, verwenden Sie den **MQSC**-Befehl START LISTENER, den Sie als Teil Ihres Kanalinitiatorstarts ausgeben. For example:

     ```
     START LISTENER PORT(1414) TRPTYPE(TCP)
     ```

     oder:

     ```
     START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
     ```

     Mitglieder einer Gruppe mit gemeinsamer Warteschlange können einen gemeinsam genutzten Listener anstelle eines Listeners für jeden Warteschlangenmanager verwenden. Verwenden Sie keine gemeinsam genutzten Empfangsprogramme mit Clustern. Machen Sie insbesondere nicht CONNAME aus dem Kanal CLUSRCVR zur Adresse des gemeinsam genutzten Listeners der Gruppe mit gemeinsamer Warteschlange. Wenn dies der Fall ist, empfangen Warteschlangenmanager möglicherweise Nachrichten für Warteschlangen, für die sie keine Definition haben.
   - IBM i Verwenden Sie unter IBM i das von IBM MQ bereitgestellte Kanallistenerprogramm. Verwenden Sie den **CL** -Befehl STRMQMLSR, um einen IBM MQ -Kanallistener zu starten. For example:

     ```
     STRMQMLSR MQMNAME(QM1) PORT(1414)
     ```

   - Windows Verwenden Sie unter Windows entweder das von IBM MQ bereitgestellte Kanallistenerprogramm oder die vom Betriebssystem zur Verfügung gestellten Funktionen.

Verwenden Sie zum Starten des IBM MQ -Kanallisteners den Befehl RUNMQLSR . For example:

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

- Linux AIX Verwenden Sie unter AIX and Linux entweder das von IBM MQ bereitgestellte Kanallistenerprogramm oder die vom Betriebssystem zur Verfügung gestellten Funktionen; z. B. **inetd** für TCP-Verbindungen.

  Verwenden Sie den Befehl **runmqlsr** , um den IBM MQ -Kanallistener zu starten. For example:

  ```
  runmqlsr -t tcp -p 1414 -m QM1
  ```

  Wenn Sie **inetd** zum Starten von Kanälen verwenden möchten, konfigurieren Sie zwei Dateien:

  a. Bearbeiten Sie die Datei `/etc/services`. Sie müssen als Superuser oder als Root angemeldet sein. Wenn die folgende Zeile nicht in der Datei enthalten ist, fügen Sie sie wie folgt hinzu:

  ```
  MQSeries    1414/tcp   # WebSphere MQ channel listener
  ```

  Dabei ist 1414 die für IBM MQ erforderliche Portnummer. Sie können die Portnummer ändern, aber sie muss mit der Port-Nummer übereinstimmen, die auf der Senderseite angegeben wurde.

  b. Bearbeiten Sie die Datei `/etc/inetd.conf`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

  ```
  MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
  -m queue.manager.name
  ```

  Dabei wird *MQ_INSTALLATION_PATH* durch das übergeordnete Verzeichnis ersetzt, in dem IBM MQ installiert ist.

  Die Aktualisierungen werden aktiv, nachdem **inetd** die Konfigurationsdateien erneut gelesen hat. Geben Sie die folgenden Befehle von der Rootbenutzer-ID aus:

  AIX Unter AIX:

  ```
  refresh -s inetd
  ```

  Linux Unter Linux:

  a. Suchen Sie die Prozess-ID von **inetd** mit dem Befehl:

  ```
  ps -ef | grep inetd
  ```

  b. Führen Sie den entsprechenden Befehl aus.

  Für Linux:

  ```
  kill -1 inetd processid
  ```

## Neuen Cluster einrichten

Führen Sie die folgenden Anweisungen aus, um den Beispielcluster zu konfigurieren. In separaten Anweisungen wird beschrieben, wie der Cluster auf TCP/IP, LU 6.2 und mit einer einzelnen Übertragungswarteschlange oder mehreren Übertragungswarteschlangen eingerichtet wird. Testen Sie den Cluster, indem Sie eine Nachricht von einem WS-Manager an den anderen Warteschlangenmanager senden.

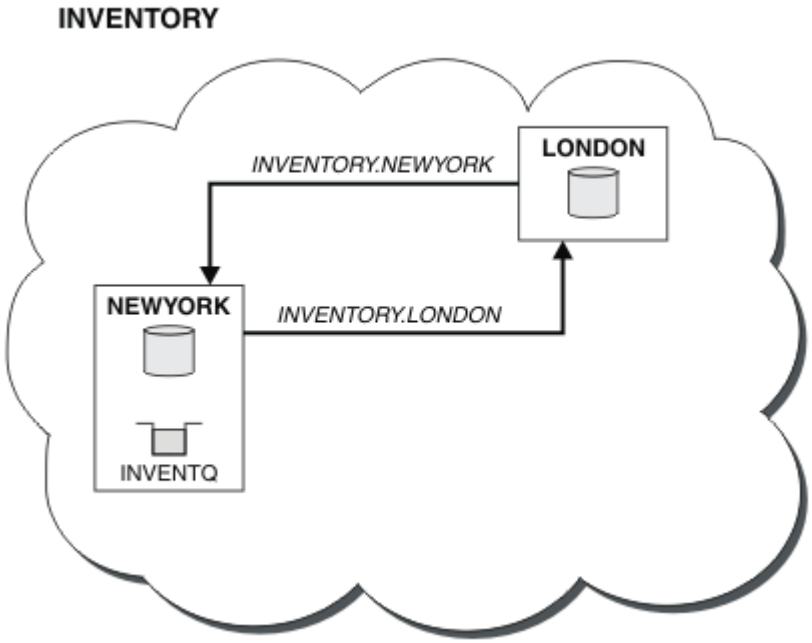
## Vorbereitende Schritte

- Anstatt diese Anweisungen zu befolgen, können Sie einen der Assistenten verwenden, die mit IBM MQ Explorer bereitgestellt werden, um einen Cluster wie den von dieser Task erstellten Cluster zu erstellen. Klicken Sie auf den Ordner WS-Manager-Cluster, klicken Sie dann auf **New** > **Queue Manager Cluster** und befolgen Sie die Anweisungen im Assistenten.
- Hintergrundinformationen zur Unterstützung Ihres Verständnisses für die Schritte zum Einrichten eines Clusters finden Sie in den Abschnitten „Clusterwarteschlangen definieren” auf Seite 313, Clusterkanäle und Listener.

## Informationen zu diesem Vorgang

Sie richten ein neues IBM MQ-Netz für eine Geschäftskette ein. Das Geschäft hat zwei Filialen, eine in London und eine in New York. Die Daten und Anwendungen für die einzelnen Filiale werden von Systemen gehostet, auf denen separate Warteschlangenmanager ausgeführt werden. Die beiden WS-Manager werden als LONDON und NEWYORK bezeichnet. Die Bestandsanwendung wird auf dem System in New York ausgeführt und ist mit dem Warteschlangenmanager NEWYORK verbunden. Die Anwendung wird durch den Eingang von Nachrichten in der INVENTQ -Warteschlange gesteuert, die von NEWYORK bereitgestellt wird. Die beiden WS-Manager LONDON und NEWYORK sollen in einem Cluster mit dem Namen INVENTORY verknüpft werden, damit beide Nachrichten an den INVENTQ stellen können.

Dieser Cluster sieht folgendermaßen




aus:

Sie können jeden WS-Manager im Cluster so konfigurieren, dass er Nachrichten an andere Warteschlangenmanager im Cluster sendet, die verschiedene Clusterübertragungswarteschlangen verwenden.

Die Anweisungen zum Einrichten des Clusters variieren je nach Transportprotokoll, Anzahl der Übertragungswarteschlangen oder Plattform. Sie haben die Wahl zwischen drei Kombinationen. Das Prüfungsverfahren bleibt für alle Kombinationen gleich.

INVENTORY ist ein kleiner Cluster. Sie ist jedoch als Proof of Concept nützlich. Das Wichtige an diesem Cluster ist der Umfang, den es für die zukünftige Erweiterung bietet.

## Prozedur

- „Cluster mit TCP/IP mit einer einzigen Übertragungswarteschlange pro WS-Manager konfigurieren” auf Seite 329

- „Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren” auf Seite 332
- „Cluster mit Logical Unit 6.2 unter z/OS einrichten” auf Seite 335
- „Cluster verifizieren” auf Seite 338

**Zugehörige Konzepte**

Cluster

Vergleich von Clustering und verteilter Steuerung von Warteschlangen

Komponenten eines Clusters

**Zugehörige Tasks**

„WS-Manager-Cluster konfigurieren” auf Seite 312

Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Sie können Cluster-Komponenten definieren und Cluster erstellen und verwalten.

## *Cluster mit TCP/IP mit einer einzigen Übertragungswarteschlange pro WS-Manager konfigurieren*

Dies ist einer von drei Themen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

### Vorbereitende Schritte

Eine Übersicht über den Cluster, der erstellt wird, finden Sie in „Neuen Cluster einrichten” auf Seite 327.

Das WS-Managerattribut **DEFCLXQ** muss als Standardwert SCTQ angegeben werden.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen Cluster unter Multiplatforms unter Verwendung des Transportprotokolls TCP/IP zu konfigurieren. z/OS Unter z/OSmüssen Sie die Anweisungen in „TCP-Verbindung unter z/OS definieren” auf Seite 1074 befolgen, um die TCP/IP-Verbindung einzurichten, anstatt die Listener in Schritt „4” auf Seite 330zu definieren. Ansonsten sind die Schritte für z/OS identisch, Fehlernachrichten werden jedoch in die Konsole geschrieben und nicht in das Fehlerprotokoll des Warteschlangenmanagers.

### Vorgehensweise

1. Entscheiden Sie sich für die Organisation des Clusters und dessen Namen.

   Sie haben entschieden, die beiden WS-Manager LONDON und NEWYORK in einen Cluster zu verlinken. Ein Cluster mit nur zwei Warteschlangenmanagern bietet nur einen marginalen Vorteil gegenüber einem Netz, das die verteilte Steuerung von Warteschlangen verwenden soll. Es ist ein guter Weg, um zu beginnen, und es bietet Raum für zukünftige Erweiterungen. Wenn Sie neue Filialen Ihres Geschäfts öffnen, können Sie die neuen WS-Manager problemlos dem Cluster hinzufügen. Durch das Hinzufügen neuer WS-Manager wird das vorhandene Netz nicht zerrüttten; siehe „WS-Manager zu einem Cluster hinzufügen” auf Seite 339.

   Die einzige Anwendung, die Sie gerade ausführen, ist die Bestandsanwendung. Der Clustername lautet INVENTORY.
2. Entscheiden Sie, welche WS-Manager vollständige Repositorys enthalten sollen.

   In jedem Cluster müssen Sie mindestens einen Warteschlangenmanager (oder vorzugsweise zwei) für die Aufnahme von vollständigen Repositorys benennen. In diesem Beispiel gibt es nur zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositorys enthalten.

   a. Sie können die restlichen Schritte in beliebiger Reihenfolge ausführen.

   b. Wenn Sie die Schritte ausführen, werden möglicherweise Warnungen in das WS-Manager-Protokoll geschrieben. Die Nachrichten sind ein Ergebnis fehlender Definitionen, die Sie noch nicht hinzugefügt haben.

```
Examples of the responses to the commands are shown in a box
like this after each step in this task.
These examples show the responses returned by IBM MQ for AIX.
The responses vary on other platforms.
```

c. Bevor Sie mit diesen Schritten fortfahren, müssen Sie sicherstellen, dass die WS-Manager gestartet wurden.

3. Ändern Sie die WS-Manager-Definitionen, um Repositorydefinitionen hinzuzufügen.

   Ändern Sie auf jedem WS-Manager, der ein vollständiges Repository aufnehmen soll, die Definition des lokalen Warteschlangenmanagers mit dem Befehl ALTER QMGR und geben Sie das Attribut REPOS an:

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

   Geben Sie Folgendes ein, wenn Sie Folgendes eingeben:

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   LONDON wird in ein vollständiges Repository geändert.

4. Definieren Sie die Empfangsprogramme.

   Definieren Sie einen Listener, der Netzanforderungen von anderen Warteschlangenmanagern für jeden WS-Manager im Cluster akzeptiert. Geben Sie auf den LONDON -Warteschlangenmanagern den folgenden Befehl aus:

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

   Mit dem Attribut CONTROL wird sichergestellt, dass das Empfangsprogramm gestartet und gestoppt wird, wenn der Warteschlangenmanager ausgeführt wird.

   Der Listener wird nicht gestartet, wenn er definiert ist. Daher muss er mit dem folgenden MQSC-Befehl zum ersten Mal manuell gestartet werden:

```
START LISTENER(LONDON_LS)
```

   Geben Sie ähnliche Befehle für alle anderen WS-Manager im Cluster ein, und ändern Sie den Namen des Listeners für die einzelnen WS-Manager.

   Es gibt mehrere Möglichkeiten, diese Listener zu definieren, wie in Listener gezeigt.

5. Definieren Sie den CLUSRCVR-Kanal für den WS-Manager von LONDON .

   Auf jedem WS-Manager in einem Cluster definieren Sie einen Clusterempfängerkanal, auf dem der Warteschlangenmanager Nachrichten empfangen kann. Siehe Clusterempfängerkanal: CLUSRCVR. Der Kanal CLUSRCVR definiert den Verbindungsnamen des Warteschlangenmanagers. Der Verbindungsname wird in den Repositorys gespeichert, auf die sich andere Warteschlangenmanager beziehen können. Das Schlüsselwort CLUSTER zeigt die Verfügbarkeit des Warteschlangenmanagers an, um Nachrichten von anderen Warteschlangenmanagern im Cluster zu empfangen.

   In diesem Beispiel lautet der Kanalname `INVENTORY.LONDON`, und der Verbindungsname (CONNAME) ist die Netzadresse der Maschine, auf der sich der Warteschlangenmanager befindet, `LONDON.CHSTORE.COM`. Die Netzadresse kann jeweils als alphanumerischer DNS-Hostname oder als IP-Adresse mit Trennzeichen gemäß IPv4 eingegeben werden. Beispiel: `192.0.2.0`, oder im Hexadezimalformat nach IPv6; Beispiel: `2001:DB8:0204:acff:fe97:2c34:fde0:3485`. Die Portnummer wurde nicht angegeben, daher wird der Standardport (1414) verwendet.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. Definieren Sie den CLUSRCVR-Kanal für den WS-Manager von NEWYORK .

   Wenn der Kanallistener den Standardport (in der Regel 1414) verwendet und der Cluster unter z/OS keinen Warteschlangenmanager enthält, kann CONNAME auch übergangen werden.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

7. Definieren Sie den CLUSSDR-Kanal auf dem LONDON -Warteschlangenmanager.

   Sie definieren manuell einen CLUSSDR -Kanal von jedem vollständigen Repository-WS-Manager zu jedem anderen Repository-WS-Manager im Cluster, der vollständig in Repository enthalten ist. Siehe Clustersenderkanal: CLUSSDR. In diesem Fall gibt es nur zwei WS-Manager, die beide vollständige Repositorys enthalten. Sie benötigen jeweils einen manuell definierten CLUSSDR -Kanal, der auf den Kanal CLUSRCVR verweist, der auf dem anderen Warteschlangenmanager definiert ist. Die Kanalnamen, die in den CLUSSDR -Definitionen angegeben sind, müssen mit den Kanalnamen in den entsprechenden CLUSRCVR -Definitionen übereinstimmen. Wenn ein Warteschlangenmanager Definitionen sowohl für einen Clusterempfängerkanal als auch für einen Clustersenderkanal in demselben Cluster enthält, wird der Clustersenderkanal gestartet.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18  Channel program started.
```

8. Definieren Sie den CLUSSDR-Kanal auf dem NEWYORK -Warteschlangenmanager.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. Definieren Sie die Clusterwarteschlange INVENTQ.

   Definieren Sie die INVENTQ -Warteschlange auf dem NEWYORK -Warteschlangenmanager und geben Sie dabei das Schlüsselwort CLUSTER an.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

   Das Schlüsselwort CLUSTER bewirkt, dass die Warteschlange für den Cluster zugänglich gemacht wird. Sobald die Warteschlange definiert ist, wird sie den anderen WS-Managern im Cluster zur Verfü-

gung gestellt. Sie können Nachrichten an sie senden, ohne eine Definition einer fernen Warteschlange für sie vornehmen zu müssen.

Alle Definitionen sind vollständig. Starten Sie auf allen Plattformen auf jedem WS-Manager ein Empfangsprogramm. Das Listenerprogramm wartet auf eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Nächste Schritte

Sie können jetzt den Cluster überprüfen.

**Zugehörige Tasks**

„Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren" auf Seite 332
Dies ist einer von drei Themen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

„Cluster mit Logical Unit 6.2 unter z/OS einrichten" auf Seite 335
Dies ist einer der Baumstrukturthemen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

# *Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren*

Dies ist einer von drei Themen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

## Vorbereitende Schritte

Eine Übersicht über den Cluster, der erstellt wird, finden Sie in „Neuen Cluster einrichten" auf Seite 327.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen Cluster unter Multiplatforms unter Verwendung des Transportprotokolls TCP/IP zu konfigurieren. Die Repository-Warteschlangenmanager sind so konfiguriert, dass sie eine andere Clusterübertragungswarteschlange verwenden, um Nachrichten an einander und an andere Warteschlangenmanager im Cluster zu senden. Wenn Sie dem Cluster Warteschlangenmanager hinzufügen, die auch andere Übertragungswarteschlangen verwenden, führen Sie die Task „WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen" auf Seite 342aus.

## Vorgehensweise

1. Entscheiden Sie sich für die Organisation des Clusters und dessen Namen.

   Sie haben entschieden, die beiden WS-Manager `LONDON` und `NEWYORK` in einen Cluster zu verlinken. Ein Cluster mit nur zwei Warteschlangenmanagern bietet nur einen marginalen Vorteil gegenüber einem Netz, das die verteilte Steuerung von Warteschlangen verwenden soll. Es ist ein guter Weg, um zu beginnen, und es bietet Raum für zukünftige Erweiterungen. Wenn Sie neue Filialen Ihres Geschäfts öffnen, können Sie die neuen WS-Manager problemlos dem Cluster hinzufügen. Durch das Hinzufügen neuer WS-Manager wird das vorhandene Netz nicht zerrüttten; siehe „WS-Manager zu einem Cluster hinzufügen" auf Seite 339.

   Die einzige Anwendung, die Sie gerade ausführen, ist die Bestandsanwendung. Der Clustername lautet `INVENTORY`.

2. Entscheiden Sie, welche WS-Manager vollständige Repositorys enthalten sollen.

   In jedem Cluster müssen Sie mindestens einen Warteschlangenmanager (oder vorzugsweise zwei) für die Aufnahme von vollständigen Repositorys benennen. In diesem Beispiel gibt es nur zwei WS-Manager, `LONDON` und `NEWYORK`, die beide vollständige Repositorys enthalten.

   a. Sie können die restlichen Schritte in beliebiger Reihenfolge ausführen.

   b. Wenn Sie die Schritte ausführen, werden möglicherweise Warnungen in das WS-Manager-Protokoll geschrieben. Die Nachrichten sind ein Ergebnis fehlender Definitionen, die Sie noch nicht hinzugefügt haben.

```
Examples of the responses to the commands are shown in a box
like this after each step in this task.
These examples show the responses returned by IBM MQ for AIX.
The responses vary on other platforms.
```

c. Bevor Sie mit diesen Schritten fortfahren, müssen Sie sicherstellen, dass die WS-Manager gestartet wurden.

3. Ändern Sie die WS-Manager-Definitionen, um Repositorydefinitionen hinzuzufügen.

   Ändern Sie auf jedem WS-Manager, der ein vollständiges Repository aufnehmen soll, die Definition des lokalen Warteschlangenmanagers mit dem Befehl ALTER QMGR und geben Sie das Attribut REPOS an:

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

   Geben Sie Folgendes ein, wenn Sie Folgendes eingeben:

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   LONDON wird in ein vollständiges Repository geändert.

4. Ändern Sie die WS-Manager-Definitionen, um separate Clusterübertragungswarteschlangen für die einzelnen Ziele zu erstellen.

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

   Auf jedem WS-Manager, den Sie dem Cluster hinzufügen, entscheiden Sie, ob separate Übertragungswarteschlangen verwendet werden sollen oder nicht. Weitere Informationen finden Sie in den Abschnitten „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 und „WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342.

5. Definieren Sie die Empfangsprogramme.

   Definieren Sie einen Listener, der Netzanforderungen von anderen Warteschlangenmanagern für jeden WS-Manager im Cluster akzeptiert. Geben Sie auf den LONDON -Warteschlangenmanagern den folgenden Befehl aus:

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

   Mit dem Attribut CONTROL wird sichergestellt, dass das Empfangsprogramm gestartet und gestoppt wird, wenn der Warteschlangenmanager ausgeführt wird.

   Der Listener wird nicht gestartet, wenn er definiert ist. Daher muss er mit dem folgenden MQSC-Befehl zum ersten Mal manuell gestartet werden:

```
START LISTENER(LONDON_LS)
```

   Geben Sie ähnliche Befehle für alle anderen WS-Manager im Cluster ein, und ändern Sie den Namen des Listeners für die einzelnen WS-Manager.

   Es gibt mehrere Möglichkeiten, diese Listener zu definieren, wie in Listener gezeigt.

6. Definieren Sie den CLUSRCVR-Kanal für den WS-Manager von LONDON .

   Auf jedem WS-Manager in einem Cluster definieren Sie einen Clusterempfängerkanal, auf dem der Warteschlangenmanager Nachrichten empfangen kann. Siehe Clusterempfängerkanal: CLUSRCVR. Der Kanal CLUSRCVR definiert den Verbindungsnamen des Warteschlangenmanagers. Der Verbin-

dungsname wird in den Repositorys gespeichert, auf die sich andere Warteschlangenmanager beziehen können. Das Schlüsselwort CLUSTER zeigt die Verfügbarkeit des Warteschlangenmanagers an, um Nachrichten von anderen Warteschlangenmanagern im Cluster zu empfangen.

In diesem Beispiel lautet der Kanalname INVENTORY.LONDON, und der Verbindungsname (CONNAME) ist die Netzadresse der Maschine, auf der sich der Warteschlangenmanager befindet, LONDON.CHSTORE.COM. Die Netzadresse kann jeweils als alphanumerischer DNS-Hostname oder als IP-Adresse mit Trennzeichen gemäß IPv4 eingegeben werden. Beispiel: 192.0.2.0, oder im Hexadezimalformat nach IPv6; Beispiel: 2001:DB8:0204:acff:fe97:2c34:fde0:3485. Die Portnummer wurde nicht angegeben, daher wird der Standardport (1414) verwendet.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. Definieren Sie den CLUSRCVR-Kanal für den WS-Manager von NEWYORK .

   Wenn der Kanallistener den Standardport (in der Regel 1414) verwendet und der Cluster unter z/OS keinen Warteschlangenmanager enthält, kann CONNAME auch übergangen werden.

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
   ```

8. Definieren Sie den CLUSSDR-Kanal auf dem LONDON -Warteschlangenmanager.

   Sie definieren manuell einen CLUSSDR -Kanal von jedem vollständigen Repository-WS-Manager zu jedem anderen Repository-WS-Manager im Cluster, der vollständig in Repository enthalten ist. Siehe Clustersenderkanal: CLUSSDR. In diesem Fall gibt es nur zwei WS-Manager, die beide vollständige Repositorys enthalten. Sie benötigen jeweils einen manuell definierten CLUSSDR -Kanal, der auf den Kanal CLUSRCVR verweist, der auf dem anderen Warteschlangenmanager definiert ist. Die Kanalnamen, die in den CLUSSDR -Definitionen angegeben sind, müssen mit den Kanalnamen in den entsprechenden CLUSRCVR -Definitionen übereinstimmen. Wenn ein Warteschlangenmanager Definitionen sowohl für einen Clusterempfängerkanal als auch für einen Clustersenderkanal in demselben Cluster enthält, wird der Clustersenderkanal gestartet.

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 13:00:18  Channel program started.
   ```

9. Definieren Sie den CLUSSDR-Kanal auf dem NEWYORK -Warteschlangenmanager.

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
   ```

10. Definieren Sie die Clusterwarteschlange INVENTQ.

Definieren Sie die INVENTQ -Warteschlange auf dem NEWYORK -Warteschlangenmanager und geben Sie dabei das Schlüsselwort CLUSTER an.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

Das Schlüsselwort CLUSTER bewirkt, dass die Warteschlange für den Cluster zugänglich gemacht wird. Sobald die Warteschlange definiert ist, wird sie den anderen WS-Managern im Cluster zur Verfügung gestellt. Sie können Nachrichten an sie senden, ohne eine Definition einer fernen Warteschlange für sie vornehmen zu müssen.

Alle Definitionen sind vollständig. Starten Sie auf allen Plattformen auf jedem WS-Manager ein Empfangsprogramm. Das Listenerprogramm wartet auf eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Nächste Schritte

Sie können jetzt den Cluster überprüfen.

**Zugehörige Tasks**

„Cluster mit TCP/IP mit einer einzigen Übertragungswarteschlange pro WS-Manager konfigurieren” auf Seite 329

Dies ist einer von drei Themen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

„Cluster mit Logical Unit 6.2 unter z/OS einrichten” auf Seite 335

Dies ist einer der Baumstrukturthemen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

## *Cluster mit Logical Unit 6.2 unter z/OS einrichten*

Dies ist einer der Baumstrukturthemen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

## Vorbereitende Schritte

Eine Übersicht über den Cluster, der erstellt wird, finden Sie in „Neuen Cluster einrichten” auf Seite 327.

## Vorgehensweise

1. Entscheiden Sie sich für die Organisation des Clusters und dessen Namen.

   Sie haben entschieden, die beiden WS-Manager LONDON und NEWYORK in einen Cluster zu verlinken. Ein Cluster mit nur zwei Warteschlangenmanagern bietet nur einen marginalen Vorteil gegenüber einem Netz, das die verteilte Steuerung von Warteschlangen verwenden soll. Es ist ein guter Weg, um zu beginnen, und es bietet Raum für zukünftige Erweiterungen. Wenn Sie neue Filialen Ihres Geschäfts öffnen, können Sie die neuen WS-Manager problemlos dem Cluster hinzufügen. Durch das Hinzufügen neuer WS-Manager wird das vorhandene Netz nicht zerrüttten; siehe „WS-Manager zu einem Cluster hinzufügen” auf Seite 339.

   Die einzige Anwendung, die Sie gerade ausführen, ist die Bestandsanwendung. Der Clustername lautet INVENTORY.
2. Entscheiden Sie, welche WS-Manager vollständige Repositorys enthalten sollen.

   In jedem Cluster müssen Sie mindestens einen Warteschlangenmanager (oder vorzugsweise zwei) für die Aufnahme von vollständigen Repositorys benennen. In diesem Beispiel gibt es nur zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositorys enthalten.

   a. Sie können die restlichen Schritte in beliebiger Reihenfolge ausführen.

b. Eventuell werden während der Ausführung der Schritte Warnungen auf der z/OS-Systemkonsole ausgegeben. Die Nachrichten sind ein Ergebnis fehlender Definitionen, die Sie noch nicht hinzugefügt haben.

c. Bevor Sie mit diesen Schritten fortfahren, müssen Sie sicherstellen, dass die WS-Manager gestartet wurden.

3. Ändern Sie die WS-Manager-Definitionen, um Repositorydefinitionen hinzuzufügen.

   Ändern Sie auf jedem WS-Manager, der ein vollständiges Repository aufnehmen soll, die Definition des lokalen Warteschlangenmanagers mit dem Befehl `ALTER QMGR` und geben Sie das Attribut `REPOS` an:

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

   ```
   1 : ALTER QMGR REPOS(INVENTORY)
   AMQ8005: IBM MQ queue manager changed.
   ```

   Geben Sie Folgendes ein, wenn Sie Folgendes eingeben:

   a. `runmqsc LONDON`

   b. `ALTER QMGR REPOS(INVENTORY)`

   `LONDON` wird in ein vollständiges Repository geändert.

4. Definieren Sie die Empfangsprogramme.

   z/OS Siehe Kanalinitiator unter z/OS und „Empfangen auf LU 6.2” auf Seite 1078.

   Der Listener wird nicht gestartet, wenn er definiert ist. Daher muss er mit dem folgenden MQSC-Befehl zum ersten Mal manuell gestartet werden:

   ```
   START LISTENER(LONDON_LS)
   ```

   Geben Sie ähnliche Befehle für alle anderen WS-Manager im Cluster ein, und ändern Sie den Namen des Listeners für die einzelnen WS-Manager.

5. Definieren Sie den CLUSRCVR-Kanal für den WS-Manager von `LONDON` .

   Auf jedem WS-Manager in einem Cluster definieren Sie einen Clusterempfängerkanal, auf dem der Warteschlangenmanager Nachrichten empfangen kann. Siehe Clusterempfängerkanal: CLUSRCVR. Der Kanal `CLUSRCVR` definiert den Verbindungsnamen des Warteschlangenmanagers. Der Verbindungsname wird in den Repositorys gespeichert, auf die sich andere Warteschlangenmanager beziehen können. Das Schlüsselwort `CLUSTER` zeigt die Verfügbarkeit des Warteschlangenmanagers an, um Nachrichten von anderen Warteschlangenmanagern im Cluster zu empfangen.

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
   CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
   MODENAME('#INTER') TPNAME('MQSERIES')
   DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
   ```

   ```
   1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
   CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
   MODENAME('#INTER') TPNAME('MQSERIES')
   DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
   AMQ8014: WebSphere MQ channel created.
   07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
   ```

6. Definieren Sie den CLUSRCVR-Kanal für den WS-Manager von `NEWYORK` .

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
   CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
   ```

```
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

7. Definieren Sie den CLUSSDR-Kanal auf dem LONDON -Warteschlangenmanager.

   Sie definieren manuell einen CLUSSDR -Kanal von jedem vollständigen Repository-WS-Manager zu jedem anderen Repository-WS-Manager im Cluster, der vollständig in Repository enthalten ist. Siehe Clustersenderkanal: CLUSSDR. In diesem Fall gibt es nur zwei WS-Manager, die beide vollständige Repositorys enthalten. Sie benötigen jeweils einen manuell definierten CLUSSDR -Kanal, der auf den Kanal CLUSRCVR verweist, der auf dem anderen Warteschlangenmanager definiert ist. Die Kanalnamen, die in den CLUSSDR -Definitionen angegeben sind, müssen mit den Kanalnamen in den entsprechenden CLUSRCVR -Definitionen übereinstimmen. Wenn ein Warteschlangenmanager Definitionen sowohl für einen Clusterempfängerkanal als auch für einen Clustersenderkanal in demselben Cluster enthält, wird der Clustersenderkanal gestartet.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18  Channel program started.
```

8. Definieren Sie den CLUSSDR-Kanal auf dem NEWYORK -Warteschlangenmanager.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. Definieren Sie die Clusterwarteschlange INVENTQ.

   Definieren Sie die INVENTQ -Warteschlange auf dem NEWYORK -Warteschlangenmanager und geben Sie dabei das Schlüsselwort CLUSTER an.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

   Das Schlüsselwort CLUSTER bewirkt, dass die Warteschlange für den Cluster zugänglich gemacht wird. Sobald die Warteschlange definiert ist, wird sie den anderen WS-Managern im Cluster zur Verfügung gestellt. Sie können Nachrichten an sie senden, ohne eine Definition einer fernen Warteschlange für sie vornehmen zu müssen.

   Alle Definitionen sind vollständig. Starten Sie auf allen Plattformen auf jedem WS-Manager ein Empfangsprogramm. Das Listenerprogramm wartet auf eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Nächste Schritte

Sie können jetzt den Cluster überprüfen.

**Zugehörige Tasks**

„Cluster mit TCP/IP mit einer einzigen Übertragungswarteschlange pro WS-Manager konfigurieren” auf Seite 329

Dies ist einer von drei Themen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

„Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren” auf Seite 332

Dies ist einer von drei Themen, die verschiedene Konfigurationen für einen einfachen Cluster beschreiben.

### *Cluster verifizieren*

Peer-Topics beschreiben drei verschiedene Konfigurationen für einen einfachen Cluster. In diesem Abschnitt wird erläutert, wie der Cluster überprüft wird.

## Vorbereitende Schritte

In diesem Abschnitt wird vorausgesetzt, dass Sie einen Cluster überprüfen, den Sie mit einer der folgenden Aufgaben erstellt haben:

- „Cluster mit TCP/IP mit einer einzigen Übertragungswarteschlange pro WS-Manager konfigurieren” auf Seite 329.
- „Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren” auf Seite 332.
- „Cluster mit Logical Unit 6.2 unter z/OS einrichten” auf Seite 335.

Eine Übersicht über den Cluster, der erstellt wurde, finden Sie unter „Neuen Cluster einrichten” auf Seite 327.

## Informationen zu diesem Vorgang

Sie können den Cluster auf eine oder mehrere der folgenden Arten überprüfen:

1. Ausführung von Verwaltungsbefehlen zum Anzeigen von Cluster-und Kanalattributen.
2. Führen Sie die Beispielprogramme aus, um Nachrichten in einer Clusterwarteschlange zu senden und zu empfangen.
3. Schreiben Sie Ihre eigenen Programme, um eine Anforderungsnachricht an eine Clusterwarteschlange zu senden und mit Antwortnachrichten an eine nicht in Gruppen zusammengefasste Antwortwarteschlange zu antworten.

## Vorgehensweise

Geben Sie DISPLAY **runmqsc** -Befehle aus, um den Cluster zu überprüfen.

Die Antworten, die Sie sehen, sollten wie die Antworten in den folgenden Schritten sein.

1. Führen Sie im Warteschlangenmanager NEWYORK den Befehl **DISPLAY CLUSQMGR** aus:

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)       CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)       CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. Führen Sie im Warteschlangenmanager NEWYORK den Befehl **DISPLAY CHANNEL STATUS** aus:

```
dis chstatus(*)
```

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK)  XMITQ( )
CONNAME(192.0.2.0)      CURRENT
CHLTYPE(CLUSRCVR)       STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNAME(192.0.2.1)      CURRENT
CHLTYPE(CLUSSDR)       STATUS(RUNNING)
RQMNAME(LONDON)
```

Senden Sie Nachrichten zwischen den beiden Warteschlangenmanagern mit **amqsput**.

3. Führen Sie unter LONDON den Befehl **amqsput INVENTQ LONDON**aus.

   Geben Sie einige Nachrichten ein, gefolgt von einer Leerzeile.
4. Führen Sie unter NEWYORK den Befehl **amqsget INVENTQ NEWYORK**aus.

   Die Nachrichten, die Sie in LONDON eingegeben haben, werden angezeigt. Nach 15 Sekunden wird das Programm beendet.

Senden Sie Nachrichten zwischen den beiden WS-Managern, die Ihre eigenen Programme verwenden.

In den folgenden Schritten stellt LONDON eine Nachricht an die INVENTQ in NEWYORK und empfängt eine Antwort in ihrer Warteschlange LONDON_reply.

5. Geben Sie in LONDON eine Nachricht in die Clusterwarteschlange ein.
   a) Definieren Sie eine lokale Warteschlange mit dem Namen LONDON_reply.
   b) Setzen Sie die Optionen für MQOPEN auf MQOO_OUTPUT.
   c) Geben Sie den Aufruf MQOPEN aus, um die Warteschlange INVENTQzu öffnen.
   d) Setzen Sie den Namen *ReplyToQ* im Nachrichtendeskriptor auf LONDON_reply.
   e) Geben Sie den Aufruf MQPUT aus, um die Nachricht einzureihen.
   f) Schreiben Sie die Nachricht fest.
6. Empfangen Sie auf NEWYORK die Nachricht in der Clusterwarteschlange und stellen Sie eine Antwort in die Antwortwarteschlange ein.
   a) Setzen Sie die Optionen für MQOPEN auf MQOO_BROWSE.
   b) Geben Sie den Aufruf MQOPEN aus, um die Warteschlange INVENTQzu öffnen.
   c) Geben Sie den MQGET-Aufruf aus, um die Nachricht von INVENTQabzurufen.
   d) Rufen Sie den Namen *ReplyToQ* aus dem Nachrichtendeskriptor ab.
   e) Geben Sie den Namen *ReplyToQ* im Feld ObjectName des Objektdeskriptors ein.
   f) Setzen Sie die Optionen für MQOPEN auf MQOO_OUTPUT.
   g) Geben Sie den Aufruf MQOPEN aus, um LONDON_reply im Warteschlangenmanager LONDONzu öffnen.
   h) Geben Sie den Aufruf MQPUT aus, um die Nachricht in LONDON_replyeinzureihen.
7. Empfangen Sie auf LONDON die Antwort.
   a) Setzen Sie die Optionen für MQOPEN auf MQOO_BROWSE.
   b) Geben Sie den Aufruf MQOPEN aus, um die Warteschlange LONDON_replyzu öffnen.
   c) Geben Sie den Aufruf MQGET aus, um die Nachricht von LONDON_replyabzurufen.

## WS-Manager zu einem Cluster hinzufügen

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung der einzigen Clusterübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE übertragen.

## Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wird wie in „Neuen Cluster einrichten” auf Seite 327 beschrieben konfiguriert. Es enthält zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositorys enthalten.
- Der WS-Manager PARIS ist Eigentum der primären Installation. Ist dies nicht der Fall, müssen Sie den Befehl **setmqenv** ausführen, um die Befehlsumgebung für die Installation einzurichten, zu der PARIS gehört.
- Die TCP-Verbindung besteht zwischen allen drei Systemen, und der Warteschlangenmanager wird mit einem TCP-Listener konfiguriert, der unter der Steuerung des Warteschlangenmanagers gestartet wird.

## Informationen zu diesem Vorgang

1. Eine neue Filiale des Filialgeschäfters wird in Paris eingerichtet und Sie möchten dem Cluster einen WS-Manager mit dem Namen PARIS hinzufügen.
2. Der Warteschlangenmanager PARIS sendet Bestandsaktualisierungen an die Anwendung, die auf dem System in New York ausgeführt wird, indem Nachrichten in die Warteschlange von INVENTQ gestellt werden.

Führen Sie die folgenden Schritte aus, um einen WS-Manager zu einem Cluster hinzuzufügen.

## Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository PARIS auf das erste Element verweist.

   Jeder WS-Manager in einem Cluster muss sich auf einen oder einen anderen der vollständigen Repositorys beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Wählen Sie eines der Repositorys als vollständiges Repository aus. Sobald ein neuer WS-Manager dem Cluster hinzugefügt wird, wird er sofort auch über das andere Repository informiert. Informationen zu Änderungen an einem WS-Manager werden direkt an zwei Repositorys gesendet. In diesem Beispiel verbinden Sie PARIS mit dem Warteschlangenmanager LONDON aus rein geographischen Gründen.

   **Anmerkung:** Führen Sie die verbleibenden Schritte in beliebiger Reihenfolge aus, nachdem der WS-Manager PARIS gestartet wurde.
2. Definieren Sie einen CLUSRCVR -Kanal auf WS-Manager PARIS.

   Jeder WS-Manager in einem Cluster muss einen Clusterempfängerkanal definieren, auf dem er Nachrichten empfangen kann. Definieren Sie unter PARIS Folgendes:

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   Der Clusterempfängerkanal wirbt für die Verfügbarkeit des Warteschlangenmanagers, um Nachrichten von anderen Warteschlangenmanagern im Cluster INVENTORY zu empfangen. Machen Sie keine Definitionen für andere WS-Manager für ein sendende Ende an den Clusterempfängerkanal INVENTORY.PARIS. Andere Definitionen werden bei Bedarf automatisch erstellt. Siehe Clusterkanäle.
3. z/OS
   Starten Sie den Kanalinitiator unter IBM MQ for z/OS.
4. Definieren Sie einen CLUSSDR -Kanal auf WS-Manager PARIS.

Wenn Sie einem Cluster einen WS-Manager hinzufügen, der kein vollständiges Repository ist, definieren Sie nur einen Clustersenderkanal, um eine erste Verbindung zu einem vollständigen Repository herzustellen. Siehe Clustersenderkanal: CLUSSDR.

Geben Sie unter PARIS die folgende Definition für einen CLUSSDR -Kanal mit dem Namen INVENTORY.LONDON mit der Netzadresse LONDON.CHSTORE.COM an den Warteschlangenmanager an.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

5. Optional: Wenn Sie einem Cluster einen WS-Manager hinzufügen, der zuvor aus demselben Cluster entfernt wurde, überprüfen Sie, ob er jetzt als Cluster-Member angezeigt wird. Ist dies nicht der Fall, führen Sie die folgenden zusätzlichen Schritte aus:
   a) Geben Sie den Befehl **REFRESH CLUSTER** auf dem Warteschlangenmanager aus, den Sie hinzufügen.

      Dieser Schritt stoppt die Clusterkanäle und gibt Ihrem lokalen Clustercache eine neue Gruppe von Folgenummern, die sichergestellt werden, dass sie im Rest des Clusters für das Up-to-Datum konfiguriert werden.

      ```
      REFRESH CLUSTER(INVENTORY) REPOS(YES)
      ```

      **Anmerkung:** Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager senden, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.
   b) Starten Sie den CLUSSDR-Kanal erneut.

      (z. B. mit dem Befehl START CHANNEL ).
   c) Starten Sie den CLUSRCVR-Kanal erneut.

## Ergebnisse

In der folgenden Abbildung ist der Cluster dargestellt, der von dieser Task eingerichtet wurde.

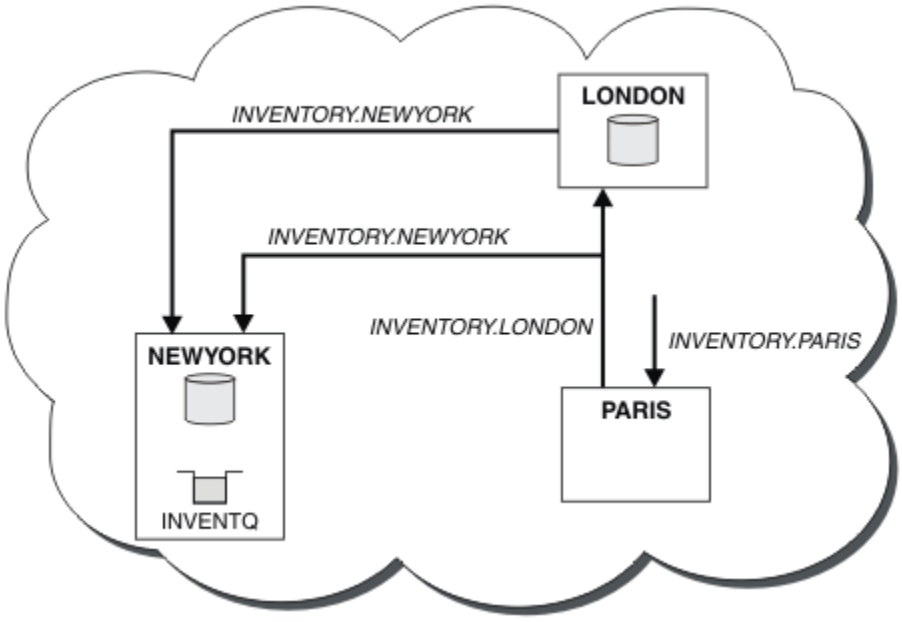



*Abbildung 39. Der INVENTORY -Cluster mit drei Warteschlangenmanagern*

Wenn Sie nur zwei Definitionen, eine CLUSRCVR -Definition und eine CLUSSDR -Definition, haben, haben wir den WS-Manager PARIS dem Cluster hinzugefügt.

Jetzt lernt der PARIS -Warteschlangenmanager aus dem vollständigen Repository in LONDON, dass die INVENTQ -Warteschlange vom WS-Manager NEWYORK gehostet wird. Wenn eine von dem System in Paris gehostete Anwendung versucht, Nachrichten an den INVENTQ zu stellen, definiert PARIS automatisch einen Clustersenderkanal, um eine Verbindung zum Clusterempfängerkanal INVENTORY.NEWYORK herzustellen. Die Anwendung kann Antworten empfangen, wenn der Name des Warteschlangenmanagers als Zielwarteschlangenmanager und eine Empfangswarteschlange für Antworten angegeben ist.

## *WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen*

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

### Vorbereitende Schritte

- Der WS-Manager ist kein Member von Clustern.
- Der Cluster ist vorhanden. Es gibt ein vollständiges Repository, zu dem dieser WS-Manager eine direkte Verbindung herstellen kann und das Repository verfügbar ist. Informationen zu den Schritten zum Erstellen des Clusters finden Sie in „Neuen Cluster einrichten” auf Seite 327.

### Informationen zu diesem Vorgang

Diese Task ist eine Alternative zu „WS-Manager zu einem Cluster hinzufügen” auf Seite 339, in der Sie einen WS-Manager zu einem Cluster hinzufügen, der Clusternachrichten in eine einzelne Übertragungswarteschlange stellt.

In dieser Task fügen Sie einen Warteschlangenmanager zu einem Cluster hinzu, der automatisch separate Clusterübertragungswarteschlangen für jeden Clustersenderkanal erstellt.

Um die Anzahl der Definitionen von Warteschlangen klein zu halten, ist die Standardeinstellung für die Verwendung einer einzelnen Übertragungswarteschlange. Die Verwendung separater Übertragungswarteschlangen ist von Vorteil, wenn Sie den Datenverkehr überwachen wollen, der für verschiedene Warteschlangenmanager und verschiedene Cluster bestimmt ist. Zudem kann auch die Trennung von Datenverkehr nach Empfängern wünschenswert sein, um diesen zu isolieren oder bestimmte Leistungsziele zu erreichen.

### Vorgehensweise

1. Ändern Sie den Standardtyp der Übertragungswarteschlange des Clusterkanals.

   Ändern Sie den WS-Manager PARIS:

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

   Jedesmal, wenn der WS-Manager einen Clustersenderkanal erstellt, um eine Nachricht an einen Warteschlangenmanager zu senden, wird eine Clusterübertragungswarteschlange erstellt. Die Übertragungswarteschlange wird nur von diesem Clustersenderkanal verwendet. Die Übertragungswarteschlange ist permanent dynamisch. Er wird aus der Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUEmit dem Namen SYSTEM.CLUSTER.TRANSMIT.*ChannelName*erstellt.

   **Achtung:** Wenn Sie eine dedizierte SYSTEM.CLUSTER.TRANSMIT.QUEUES -Instanz mit einem Warteschlangenmanager verwenden, für den ein Upgrade von einer früheren Produktversion als IBM WebSphere MQ 7.5durchgeführt wurde, müssen Sie sicherstellen, dass für SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE die Option SHARE/NOSHARE auf **SHARE** gesetzt ist.
2. Entscheiden Sie, welches vollständige Repository PARIS auf das erste Element verweist.

Jeder WS-Manager in einem Cluster muss sich auf einen oder einen anderen der vollständigen Repositorys beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Wählen Sie eines der Repositorys als vollständiges Repository aus. Sobald ein neuer WS-Manager dem Cluster hinzugefügt wird, wird er sofort auch über das andere Repository informiert. Informationen zu Änderungen an einem WS-Manager werden direkt an zwei Repositorys gesendet. In diesem Beispiel verbinden Sie PARIS mit dem Warteschlangenmanager LONDON aus rein geographischen Gründen.

**Anmerkung:** Führen Sie die verbleibenden Schritte in beliebiger Reihenfolge aus, nachdem der WS-Manager PARIS gestartet wurde.

3. Definieren Sie einen CLUSRCVR -Kanal auf WS-Manager PARIS.

   Jeder WS-Manager in einem Cluster muss einen Clusterempfängerkanal definieren, auf dem er Nachrichten empfangen kann. Definieren Sie unter PARIS Folgendes:

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   Der Clusterempfängerkanal wirbt für die Verfügbarkeit des Warteschlangenmanagers, um Nachrichten von anderen Warteschlangenmanagern im Cluster INVENTORY zu empfangen. Machen Sie keine Definitionen für andere WS-Manager für ein sendende Ende an den Clusterempfängerkanal INVENTORY.PARIS. Andere Definitionen werden bei Bedarf automatisch erstellt. Siehe Clusterkanäle.

4. Definieren Sie einen CLUSSDR -Kanal auf WS-Manager PARIS.

   Wenn Sie einem Cluster einen WS-Manager hinzufügen, der kein vollständiges Repository ist, definieren Sie nur einen Clustersenderkanal, um eine erste Verbindung zu einem vollständigen Repository herzustellen. Siehe Clustersenderkanal: CLUSSDR.

   Geben Sie unter PARIS die folgende Definition für einen CLUSSDR -Kanal mit dem Namen INVENTORY.LONDON mit der Netzadresse LONDON.CHSTORE.COM an den Warteschlangenmanager an.

   ```
   DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at LONDON')
   ```

   Der WS-Manager erstellt automatisch die permanente dynamische Clusterübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON aus der Modellwarteschlange SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. Sie setzt das Attribut CLCHNAME der Übertragungswarteschlange auf INVENTORY.LONDON.

## Ergebnisse

In der folgenden Abbildung ist der Cluster dargestellt, der von dieser Task eingerichtet wurde.

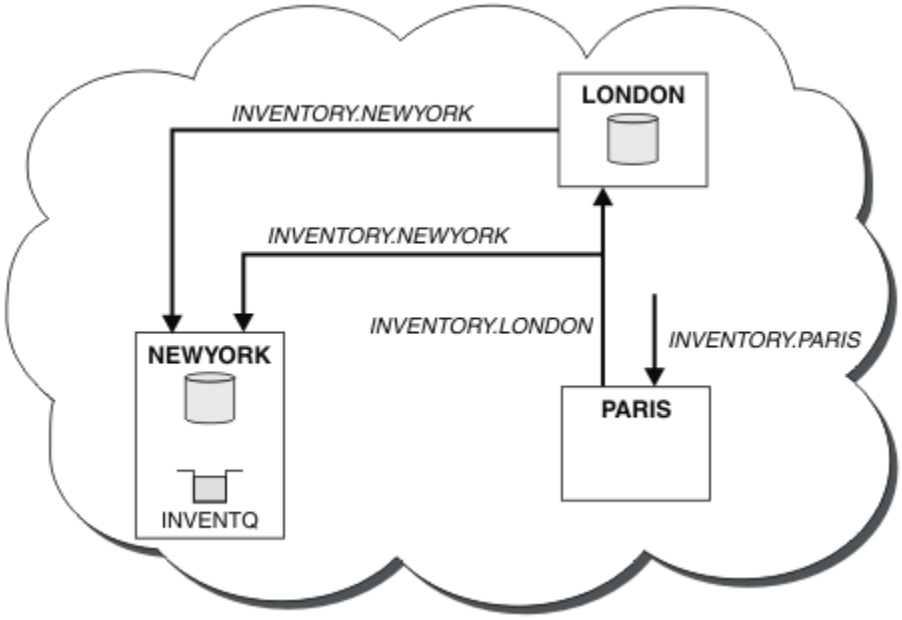



*Abbildung 40. Der INVENTORY -Cluster mit drei Warteschlangenmanagern*

Wenn Sie nur zwei Definitionen, eine CLUSRCVR -Definition und eine CLUSSDR -Definition, haben, haben wir den WS-Manager PARIS dem Cluster hinzugefügt.

Jetzt lernt der PARIS -Warteschlangenmanager aus dem vollständigen Repository in LONDON, dass die INVENTQ -Warteschlange vom WS-Manager NEWYORK gehostet wird. Wenn eine von dem System in Paris gehostete Anwendung versucht, Nachrichten an den INVENTQ zu stellen, definiert PARIS automatisch einen Clustersenderkanal, um eine Verbindung zum Clusterempfängerkanal INVENTORY.NEWYORK herzustellen. Die Anwendung kann Antworten empfangen, wenn der Name des Warteschlangenmanagers als Zielwarteschlangenmanager und eine Empfangswarteschlange für Antworten angegeben ist.

**Zugehörige Konzepte**

Art der zu verwendenden Clusterübertragungswarteschlange auswählen

**Zugehörige Tasks**

Mit DHCP einem Cluster einen WS-Manager hinzufügen

Fügen Sie einen Warteschlangenmanager zu einem Cluster unter Verwendung von DHCP hinzu. Die Task zeigt das Auslassen von CONNAME -Wert in einer CLUSRCVR -Definition an.

## *Mit DHCP einem Cluster einen WS-Manager hinzufügen*

Fügen Sie einen Warteschlangenmanager zu einem Cluster unter Verwendung von DHCP hinzu. Die Task zeigt das Auslassen von CONNAME -Wert in einer CLUSRCVR -Definition an.

## Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

In der Übung werden zwei spezielle Funktionen veranschaulicht:

- Die Möglichkeit, den Wert CONNAME in einer CLUSRCVR -Definition zu übergehen.
- Die Möglichkeit, +QMNAME+ in einer CLUSSDR -Definition zu verwenden.

Unter z/OS wird keine der Funktionen bereitgestellt.

Szenario:

- Der INVENTORY-Cluster wurde wie in „Neuen Cluster einrichten” auf Seite 327 beschrieben eingerichtet. Es enthält zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositorys enthalten.
- Eine neue Filiale des Filialgeschäfters wird in Paris eingerichtet und Sie möchten dem Cluster einen WS-Manager mit dem Namen PARIS hinzufügen.
- Warteschlangenmanager PARIS sendet Bestandsaktualisierungen an die Anwendung, die auf dem System in New York ausgeführt wird, indem Nachrichten in die Warteschlange INVENTQ gestellt werden.
- Die Netzkonnektivität besteht zwischen allen drei Systemen.
- Das Netzprotokoll ist TCP.
- Das WS-Manager-System von PARIS verwendet DHCP, d. es bedeutet, dass sich die IP-Adressen beim Systemneustart ändern können.
- Die Kanäle zwischen den Systemen PARIS und LONDON werden gemäß einer definierten Namenskonvention benannt. Die Konvention verwendet den WS-Manager-Namen des vollständigen Repository-WS-Managers auf LONDON.
- Administratoren des PARIS -Warteschlangenmanagers verfügen über keine Informationen zum Namen des Warteschlangenmanagers im LONDON -Repository. Der Name des WS-Managers auf dem LONDON -Repository unterliegt Änderungen.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen WS-Manager mit DHCP einem Cluster hinzuzufügen.

## Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository PARIS auf das erste Element verweist.

   Jeder WS-Manager in einem Cluster muss sich auf einen oder einen anderen der vollständigen Repositorys beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Wählen Sie eines der Repositorys als vollständiges Repository aus. Sobald ein neuer WS-Manager dem Cluster hinzugefügt wird, wird er sofort auch über das andere Repository informiert. Informationen zu Änderungen an einem WS-Manager werden direkt an zwei Repositorys gesendet. In diesem Beispiel wird PARIS aus geographischen Gründen mit dem WS-Manager LONDON verknüpft.

   **Anmerkung:** Führen Sie die verbleibenden Schritte in beliebiger Reihenfolge aus, nachdem der WS-Manager PARIS gestartet wurde.
2. Definieren Sie einen CLUSRCVR-Kanal auf WS-Manager PARIS.

   Jeder WS-Manager in einem Cluster muss einen Clusterempfängerkanal definieren, auf dem er Nachrichten empfangen kann. Definieren Sie unter PARIS Folgendes:

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
   TRPTYPE(TCP) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager PARIS')
   ```

   Der Clusterempfängerkanal wirbt für die Verfügbarkeit des Warteschlangenmanagers, um Nachrichten von anderen Warteschlangenmanagern im Cluster INVENTORY zu empfangen. Sie müssen den CONNAME nicht auf dem Clusterempfängerkanal angeben. Sie können IBM MQ anfordern, um den Verbindungsnamen aus dem System zu ermitteln, indem Sie entweder CONNAME weglassen oder CONNAME(' ') angeben. IBM MQ generiert den Wert für CONNAME unter Verwendung der aktuellen IP-Adresse des Systems. Weitere Informationen finden Sie im Abschnitt CONNAME. Es ist nicht erforderlich, Definitionen auf anderen Warteschlangenmanagern für ein sendende Ende an den Clusterempfängerkanal INVENTORY.PARIS zu setzen. Andere Definitionen werden bei Bedarf automatisch erstellt.
3. Definieren Sie einen CLUSSDR-Kanal auf WS-Manager PARIS.

   Jeder WS-Manager in einem Cluster muss einen Clustersenderkanal definieren, auf dem er Nachrichten an sein erstes vollständiges Repository senden kann. Geben Sie unter PARIS die folgende Definiti-

on für einen Kanal mit dem Namen INVENTORY.+QMNAME+ an den Warteschlangenmanager mit der Netzadresse LONDON.CHSTORE.COM an.

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. Optional: Wenn Sie einem Cluster einen WS-Manager hinzufügen, der zuvor aus demselben Cluster entfernt wurde, überprüfen Sie, ob er jetzt als Cluster-Member angezeigt wird. Ist dies nicht der Fall, führen Sie die folgenden zusätzlichen Schritte aus:
   a) Geben Sie den Befehl **REFRESH CLUSTER** auf dem Warteschlangenmanager aus, den Sie hinzufügen.

      Dieser Schritt stoppt die Clusterkanäle und gibt Ihrem lokalen Clustercache eine neue Gruppe von Folgenummern, die sichergestellt werden, dass sie im Rest des Clusters für das Up-to-Datum konfiguriert werden.

      ```
      REFRESH CLUSTER(INVENTORY) REPOS(YES)
      ```

      **Anmerkung:** Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager senden, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.
   b) Starten Sie den CLUSSDR-Kanal erneut.

      (z. B. mit dem Befehl START CHANNEL ).
   c) Starten Sie den CLUSRCVR-Kanal erneut.

## Ergebnisse

Der Cluster, der von dieser Task eingerichtet wird, entspricht dem von „WS-Manager zu einem Cluster hinzufügen” auf Seite 339:

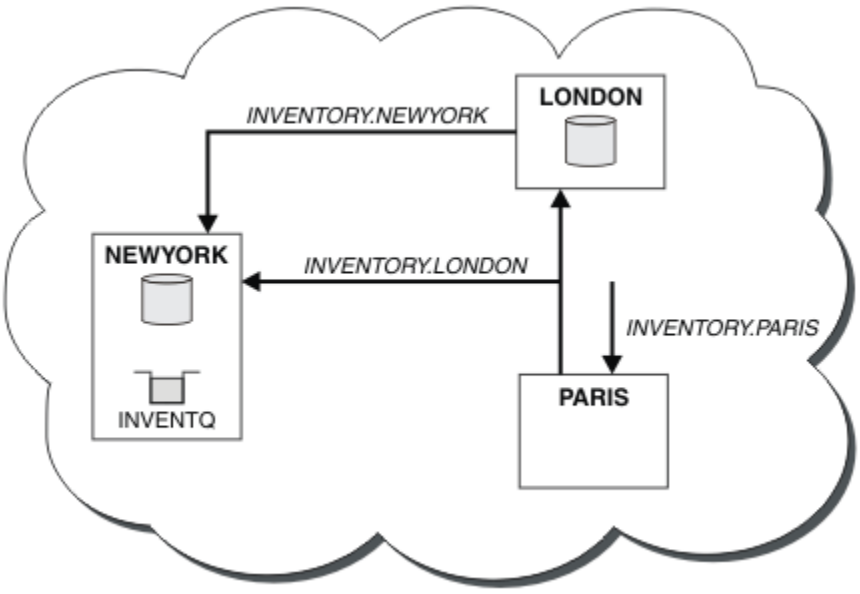



*Abbildung 41. Cluster BESTANDSFÜHRUNG mit drei Warteschlangenmanagern*

Wenn Sie nur zwei Definitionen, eine CLUSRCVR -Definition und eine CLUSSDR -Definition angeben, haben wir den WS-Manager PARIS dem Cluster hinzugefügt.

Auf dem PARIS -Warteschlangenmanager wird der CLUSSDR mit der Zeichenfolge +QMNAME+ gestartet. Auf dem LONDON-System löst IBM MQ den +QMNAME+ auf den Warteschlangenmanagernamen ( LONDON). IBM MQ stimmt dann mit der Definition für einen Kanal mit dem Namen INVENTORY.LONDON für die entsprechende CLUSRCVR-Definition überein.

IBM MQ sendet den aufgelösten Kanalnamen an den PARIS-Warteschlangenmanager zurück. In PARIS wird die Kanaldefinition CLUSSDR für den Kanal mit dem Namen INVENTORY.+QMNAME+ durch eine intern generierte CLUSSDR -Definition für INVENTORY.LONDON ersetzt. Diese Definition enthält den aufgelösten Kanalnamen, ist aber ansonsten mit der von Ihnen vorgenommenen Definition von +QMNAME+ identisch. Die Cluster-Repositorys werden auch mit der Kanaldefinition mit dem neu aufgelösten Kanalnamen auf dem neuesten Stand gebracht.

**Anmerkung:**

1. Der Kanal, der mit dem Namen +QMNAME+ erstellt wurde, wird sofort inaktiv. Es wird nie verwendet, um Daten zu übertragen.
2. Kanalexits sehen möglicherweise die Änderung des Kanalnamens zwischen einem Aufruf und dem nächsten.

Jetzt lernt der PARIS -Warteschlangenmanager aus dem Repository in LONDON, dass die INVENTQ -Warteschlange vom WS-Manager NEWYORK gehostet wird. Wenn eine von dem System in Paris gehostete Anwendung versucht, Nachrichten an den INVENTQ, PARIS zu stellen, definiert automatisch einen Clustersenderkanal, um eine Verbindung zum Clusterempfängerkanal INVENTORY.NEWYORK herzustellen. Die Anwendung kann Antworten empfangen, wenn der Name des Warteschlangenmanagers als Zielwarteschlangenmanager und eine Empfangswarteschlange für Antworten angegeben ist.

**Zugehörige Tasks**

WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

**Zugehörige Verweise**

CHANNEL DEFINE CHANNEL

## Hinzufügen eines Warteschlangenmanagers, der eine Warteschlange enthält

Fügen Sie einen weiteren WS-Manager zum Cluster hinzu, um eine weitere INVENTQ -Warteschlange zu hosten. Anforderungen werden abwechselnd an die Warteschlangen in den einzelnen Warteschlangenmanagern gesendet. Es müssen keine Änderungen an dem vorhandenen INVENTQ -Host vorgenommen werden.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY-Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben eingerichtet. Es enthält drei Warteschlangenmanager; LONDON und NEWYORK enthalten beide vollständige Repositorys, PARIS enthält ein Teilrepository. Die Bestandsanwendung wird auf dem System in New York ausgeführt und ist mit dem NEWYORK -Warteschlangenmanager verbunden. Die Anwendung wird durch den Eingang von Nachrichten in der INVENTQ -Warteschlange gesteuert.
- In Toronto wird ein neues Geschäft aufgebaut. Um zusätzliche Kapazitäten bereitzustellen, möchten Sie die Inventaranwendung auf dem System in Toronto sowie in New York ausführen.
- Die Netzkonnektivität besteht zwischen allen vier Systemen.
- Das Netzprotokoll ist TCP.

**Anmerkung:** Der WS-Manager TORONTO enthält nur ein Teilrepository. Wenn Sie einen WS-Manager mit vollem Repository zu einem Cluster hinzufügen möchten, lesen Sie den Abschnitt „Vollrepositorys in einen anderen WS-Manager verschieben” auf Seite 352.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen WS-Manager hinzuzufügen, der eine Warteschlange enthält.

## Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository TORONTO auf das erste Element verweist.

   Jeder WS-Manager in einem Cluster muss sich auf einen oder einen anderen der vollständigen Repositorys beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Es ist nicht besonders wichtig, welches Repository Sie auswählen. In diesem Beispiel wählen Sie NEWYORK aus. Sobald der neue WS-Manager dem Cluster beigetreten ist, kommuniziert er mit beiden Repositorys.
2. Definieren Sie den Kanal CLUSRCVR .

   Jeder WS-Manager in einem Cluster muss einen Clusterempfängerkanal definieren, auf dem er Nachrichten empfangen kann. Definieren Sie unter TORONTO einen CLUSRCVR -Kanal:

   ```
   DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for TORONTO')
   ```

   Der TORONTO -Warteschlangenmanager macht seine Verfügbarkeit für den Empfang von Nachrichten von anderen WS-Managern im INVENTORY -Cluster mit Hilfe seines Clusterempfängerkanals bekannt.
3. Definieren Sie einen CLUSSDR -Kanal auf WS-Manager TORONTO.

   Jeder Warteschlangenmanager in einem Cluster muss einen Clustersenderkanal definieren, auf dem er Nachrichten an sein erstes vollständiges Repository senden kann. Wählen Sie in diesem Fall NEWYORK aus. TORONTO benötigt die folgende Definition:

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
   ```
4. Optional: Wenn Sie einem Cluster einen WS-Manager hinzufügen, der zuvor aus demselben Cluster entfernt wurde, überprüfen Sie, ob er jetzt als Cluster-Member angezeigt wird. Ist dies nicht der Fall, führen Sie die folgenden zusätzlichen Schritte aus:
   a) Geben Sie den Befehl **REFRESH CLUSTER** auf dem Warteschlangenmanager aus, den Sie hinzufügen.

      Dieser Schritt stoppt die Clusterkanäle und gibt Ihrem lokalen Clustercache eine neue Gruppe von Folgenummern, die sichergestellt werden, dass sie im Rest des Clusters für das Up-to-Datum konfiguriert werden.

      ```
      REFRESH CLUSTER(INVENTORY) REPOS(YES)
      ```

      **Anmerkung:** Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager senden, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.
   b) Starten Sie den CLUSSDR-Kanal erneut.

      (z. B. mit dem Befehl START CHANNEL ).
   c) Starten Sie den CLUSRCVR-Kanal erneut.

5. Überprüfen Sie die Bestandsanwendung auf Nachrichtenaffinitäten.

   Bevor Sie fortfahren, stellen Sie sicher, dass die Inventaranwendung keine Abhängigkeiten von der Reihenfolge der Verarbeitung von Nachrichten hat und die Anwendung auf dem System in Toronto installiert.
6. Definieren Sie die Clusterwarteschlange INVENTQ.

   Die INVENTQ -Warteschlange, die bereits vom NEWYORK -Warteschlangenmanager gehostet wird, befindet sich ebenfalls in TORONTO. Definieren Sie sie auf dem TORONTO -Warteschlangenmanager wie folgt:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

## Ergebnisse

In Abbildung 42 auf Seite 349 wird der INVENTORY -Cluster angezeigt, der von dieser Task eingerichtet wurde.

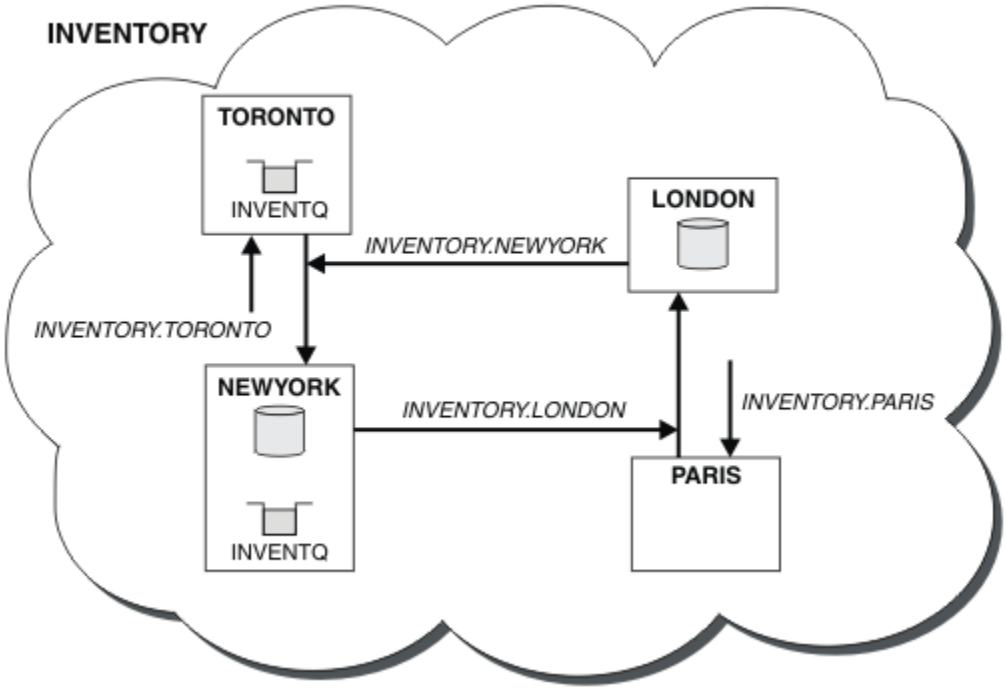



*Abbildung 42. Der INVENTORY -Cluster mit vier Warteschlangenmanagern*

Die INVENTQ -Warteschlange und die Inventaranwendung werden jetzt auf zwei Warteschlangenmanagern im Cluster gehostet. Dies erhöht die Verfügbarkeit, beschleunigt den Durchsatz von Nachrichten und ermöglicht die Verteilung der Auslastung zwischen den beiden Warteschlangenmanagern. Nachrichten, die entweder von TORONTO oder NEWYORK an INVENTQ gestellt werden, werden, wenn möglich, von der Instanz auf dem lokalen WS-Manager bearbeitet. Nachrichten, die von LONDON oder PARIS gestellt werden, werden abwechselnd an TORONTO oder NEWYORK weitergeleitet, so dass die Auslastung ausgeglichen ist.

Diese Änderung am Cluster wurde ausgeführt, ohne dass die Definitionen in den Warteschlangenmanagern NEWYORK, LONDON und PARIS geändert werden müssen. Die vollständigen Repositorys in diesen Warteschlangenmanagern werden automatisch mit den Informationen aktualisiert, die sie benötigen, um Nachrichten an INVENTQ in TORONTO senden zu können. Die Inventaranwendung funktioniert weiterhin, wenn einer der NEWYORK oder der WS-Manager von TORONTO nicht mehr verfügbar ist und die Kapazität ausreicht. Die Bestandsanwendung muss in der Lage sein, ordnungsgemäß zu arbeiten, wenn sie an beiden Standorten gehostet wird.

Wie Sie im Ergebnis dieser Task sehen können, können Sie dieselbe Anwendung in mehr als einem Warteschlangenmanager ausführen. Sie können das Clustering gleichmäßig auf die Verteilungsauslastung verteilen.

Eine Anwendung ist möglicherweise nicht in der Lage, Datensätze an beiden Standorten zu verarbeiten. Angenommen, Sie möchten eine Abfrage zum Kunden-Account hinzufügen und die Anwendung, die in LONDON und NEWYORK ausgeführt wird, hinzufügen. Ein Accountendatensatz kann nur an einem Ort gehalten werden. Sie können die Verteilung von Anforderungen mit Hilfe eines Datenpartitionierungsverfahrens steuern. Sie können die Verteilung der Datensätze aufteilen. Sie können die Hälfte der Datensätze anordnen, z. B. für die Kontonummern 00000-49999, die in LONDON gehalten werden sollen. Die andere Hälfte (im Bereich 50000-99999 ) wird in NEWYORK gehalten. Anschließend können Sie ein Exitprogramm für die Clusterauslastung schreiben, um das Accountfeld in allen Nachrichten zu untersuchen und die Nachrichten an den entsprechenden Warteschlangenmanager weiterzuleiten.

### Nächste Schritte

Nachdem Sie alle Definitionen erstellt haben, starten Sie nun unter IBM MQ for z/OS den Kanalinitiator, sofern dies noch nicht geschehen ist. Starten Sie auf allen Plattformen ein Listenerprogramm auf dem Warteschlangenmanager TORONTO. Das Listenerprogramm wartet auf eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## z/OS Gruppe mit gemeinsamer Warteschlange zu vorhandenen Clustern hinzufügen

Sie können eine Gruppe mit gemeinsamer Warteschlange unter z/OS zu vorhandenen Clustern hinzufügen.

### Vorbereitende Schritte

**Anmerkung:**

1. Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.
2. Gruppen mit gemeinsamer Warteschlange werden nur unter IBM MQ for z/OS unterstützt. Diese Task ist nicht auf andere Plattformen anwendbar.

Szenario:

- Der INVENTORY -Cluster wurde wie in „Neuen Cluster einrichten” auf Seite 327beschrieben eingerichtet. Er enthält zwei WS-Manager, LONDON und NEWYORK.
- Sie möchten diesem Cluster eine Gruppe mit gemeinsamer Warteschlange hinzufügen. Die Gruppe QSGP besteht aus drei Warteschlangenmanagern, P1, P2 und P3. Sie nutzen eine Instanz der INVENTQ -Warteschlange, die von P1 definiert werden soll.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um neue WS-Manager hinzuzufügen, die eine gemeinsam genutzte Warteschlange enthalten.

### Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository die WS-Manager zuerst lesen sollen.

   Jeder WS-Manager in einem Cluster muss sich auf einen oder einen anderen der vollständigen Repositorys beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Es ist nicht besonders wichtig, welches vollständige Repository Sie auswählen. Wählen Sie in diesem Beispiel NEWYORK aus. Sobald die Gruppe mit gemeinsamer Warteschlange mit dem Cluster verknüpft wurde, kommuniziert sie mit beiden vollständigen Repositorys.

2. Definieren Sie die CLUSRCVR -Kanäle.

   Jeder WS-Manager in einem Cluster muss einen Clusterempfängerkanal definieren, auf dem er Nachrichten empfangen kann. Definieren Sie in P1, P2 und P3 Folgendes:

   ```
   DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for sharing queue manager')
   ```

   Der Clusterempfängerkanal wirbt für die Verfügbarkeit jedes Warteschlangenmanagers, um Nachrichten von anderen Warteschlangenmanagern im Cluster INVENTORY zu empfangen.
3. Definieren Sie einen CLUSSDR-Kanal für die Gruppe mit gemeinsamer Warteschlange.

   Jedes Mitglied eines Clusters muss einen Clustersenderkanal definieren, auf dem er Nachrichten an sein erstes vollständiges Repository senden kann. In diesem Fall haben wir NEWYORK ausgewählt. Für einen der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange ist die folgende Gruppendefinition erforderlich. Die Definition stellt sicher, dass jeder WS-Manager über eine Clustersenderkanaldefinition verfügt.

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
   DESCR('Cluster-sender channel to repository at NEWYORK')
   ```
4. Definieren Sie die gemeinsam genutzte Warteschlange.

   Definieren Sie die Warteschlange INVENTQ in P1 wie folgt:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
   ```

   Starten Sie den Kanalinitiator und ein Listenerprogramm auf dem neuen Warteschlangenmanager. Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

Abbildung 43 auf Seite 352 zeigt den durch diese Task konfigurierten Cluster.

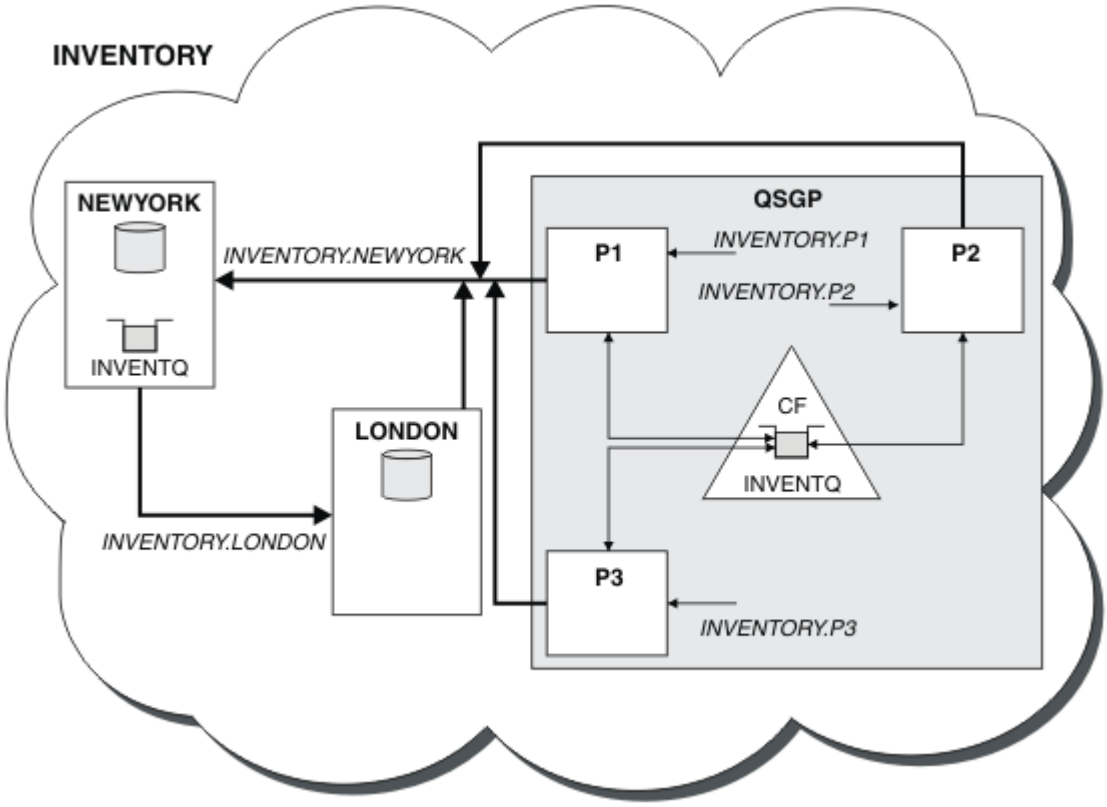



*Abbildung 43. Cluster und Gruppe mit gemeinsamer Warteschlange*

Nun werden Nachrichten, die von LONDON in die INVENTQ -Warteschlange gestellt werden, abwechselnd um die vier Warteschlangenmanager weitergeleitet, die als Host für die Warteschlange beworben wurden.

### Nächste Schritte

Ein Vorteil bei der Verwendung von Mitgliedern einer Gruppe mit gemeinsamer Warteschlange als Host in einer Clusterwarteschlange ist, dass jedes Mitglied der Gruppe auf eine Anforderung antworten kann. In diesem Fall ist P1 möglicherweise nicht mehr verfügbar, nachdem eine Nachricht in der gemeinsam genutzten Warteschlange empfangen wurde. Ein anderes Mitglied der Gruppe mit gemeinsamer Warteschlange kann stattdessen antworten.

## Vollrepositorys in einen anderen WS-Manager verschieben

Versetzen Sie ein vollständiges Repository von einem WS-Manager in einen anderen, und erstellen Sie das neue Repository aus den Informationen, die im zweiten Repository enthalten sind.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY-Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben eingerichtet.
- Aus geschäftlichen Gründen möchten Sie jetzt das vollständige Repository aus dem WS-Manager LONDON entfernen und durch ein vollständiges Repository im Warteschlangenmanager PARIS ersetzen. Der WS-Manager von NEWYORK muss weiterhin ein vollständiges Repository enthalten.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um ein vollständiges Repository in einen anderen WS-Manager zu verschieben.

## Vorgehensweise

1. Ändern Sie PARIS , um es zu einem vollständigen WS-Manager-Repository zu machen.

   Geben Sie unter PARIS den folgenden Befehl aus:

   ```
   ALTER QMGR REPOS(INVENTORY)
   ```

2. Fügen Sie einen CLUSSDR -Kanal unter PARIS hinzu.

   PARIS hat derzeit einen Clustersenderkanal, der auf LONDON zeigt. LONDON ist nicht mehr zum Speichern eines vollständigen Repositorys für den Cluster mehr erforderlich. PARIS muss über einen neuen Clustersenderkanal verfügen, der auf NEWYORK verweist, wo das andere vollständige Repository jetzt angehalten ist.

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
   ```

3. Definieren Sie einen CLUSSDR -Kanal in NEWYORK , der auf PARIS verweist.

   Derzeit verfügt NEWYORK über einen Clustersenderkanal, der auf LONDON zeigt. Nun, da das andere vollständige Repository in PARIS verschoben wurde, müssen Sie einen neuen Clustersenderkanal in NEWYORK hinzufügen, der auf PARIS verweist.

   ```
   DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
   ```

   Wenn Sie den Clustersenderkanal zu PARIS hinzufügen, erfährt PARIS von NEWYORK über den Cluster. Es baut sein eigenes vollständiges Repository mit Hilfe der Informationen aus NEWYORK auf.

4. Überprüfen Sie, ob der Warteschlangenmanager PARIS jetzt über ein vollständiges Repository verfügt.

   Überprüfen Sie, ob der WS-Manager PARIS ein eigenes vollständiges Repository aus dem vollständigen Repository auf dem Warteschlangenmanager NEWYORK erstellt hat. Setzen Sie die folgenden Befehle ab:

   ```
   DIS QCLUSTER(*) CLUSTER (INVENTORY)
   DIS CLUSQMGR(*) CLUSTER (INVENTORY)
   ```

   Überprüfen Sie, ob diese Befehle Details zu denselben Ressourcen in diesem Cluster wie in NEWYORK enthalten.

   **Anmerkung:** Wenn der WS-Manager NEWYORK nicht verfügbar ist, kann dieses Informationsgebäude nicht vollständig ausgeführt werden. Bewegen Sie sich nicht zum nächsten Schritt, bis die Task abgeschlossen ist.

5. Warteschlangenmanagerdefinition in LONDON ändern

   Ändern Sie schließlich den Warteschlangenmanager in LONDON , so dass er kein vollständiges Repository mehr für den Cluster enthält. Geben Sie unter LONDON den folgenden Befehl aus:

   ```
   ALTER QMGR REPOS(' ')
   ```

   Der WS-Manager erhält keine Clusterinformationen mehr. Nach 30 Tagen laufen die Informationen ab, die in ihrem vollständigen Repository gespeichert sind. Der WS-Manager LONDON erstellt jetzt ein eigenes Teilrepository.

6. Entfernen oder ändern Sie alle ausstehenden Definitionen.

   Wenn Sie sicher sind, dass die neue Anordnung Ihres Clusters wie erwartet funktioniert, entfernen oder ändern Sie manuell definierte CLUSSDR-Definitionen, die nicht mehr korrekt sind.

   - Auf dem PARIS -Warteschlangenmanager müssen Sie den Clustersenderkanal auf LONDON stoppen und löschen und anschließend den Befehl start channel absetzen, damit der Cluster die automatischen Kanäle erneut verwenden kann:

     ```
     STOP CHANNEL(INVENTORY.LONDON)
     DELETE CHANNEL(INVENTORY.LONDON)
     START CHANNEL(INVENTORY.LONDON)
     ```

   - Auf dem NEWYORK -Warteschlangenmanager müssen Sie den Clustersenderkanal auf LONDON stoppen und löschen und anschließend den Befehl start channel absetzen, damit der Cluster die automatischen Kanäle erneut verwenden kann:

     ```
     STOP CHANNEL(INVENTORY.LONDON)
     DELETE CHANNEL(INVENTORY.LONDON)
     START CHANNEL(INVENTORY.LONDON)
     ```

   - Ersetzen Sie alle anderen manuell definierten Clustersenderkanäle, die auf LONDON verweisen, auf allen WS-Managern im Cluster mit Kanälen, die auf NEWYORK oder PARIS verweisen. Setzen Sie nach dem Löschen eines Kanals immer den Befehl **start channel** ab, damit der Cluster die automatischen Kanäle erneut verwenden kann. In diesem kleinen Beispiel gibt es keine anderen. Um zu überprüfen, ob andere Benutzer vergessen haben, geben Sie den Befehl DISPLAY CHANNEL von jedem WS-Manager aus und geben Sie dabei TYPE(CLUSSDR) an. For example:

     ```
     DISPLAY CHANNEL(*) TYPE(CLUSSDR)
     ```

   Es ist wichtig, dass Sie diese Task so schnell wie möglich ausführen, nachdem Sie das vollständige Repository von LONDON in PARIS verschoben haben. In der Zeit, bevor Sie diese Task ausführen, können Warteschlangenmanager, die über manuell definierte CLUSSDR-Kanäle mit dem Namen INVENTORY.LONDON verfügen, Anforderungen für Informationen senden, die diesen Kanal verwenden.

   Nachdem LONDON nicht mehr ein vollständiges Repository ist, wird es, wenn es solche Anforderungen empfängt, Fehlernachrichten in das Fehlerprotokoll des Warteschlangenmanagers schreiben. Die folgenden Beispiele zeigen, welche Fehlernachrichten in LONDON angezeigt werden:

   - AMQ9428: Unexpected publication of a cluster queue object received
   - AMQ9432: Query received by a non-repository queue manager

   Der Warteschlangenmanager LONDON antwortet nicht auf die Anforderungen für Informationen, da er kein vollständiges Repository mehr ist. Die Warteschlangenmanager, die Informationen von LONDON anfordern, müssen sich auf NEWYORK für Clusterinformationen verlassen, bis ihre manuell definierten CLUSSDR-Definitionen korrigiert werden, um auf PARIS zu verweisen. Diese Situation darf auf lange Sicht nicht als gültige Konfiguration toleriert werden.

## Ergebnisse

In Abbildung 44 auf Seite 355 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

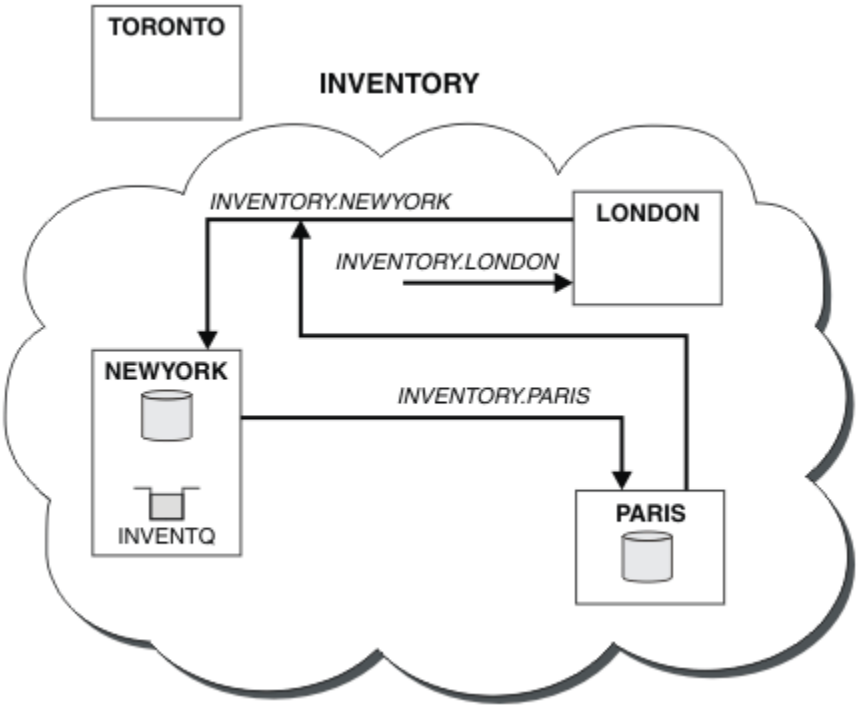



*Abbildung 44. Der INVENTORY -Cluster mit dem vollständigen Repository wurde in PARIS verschoben.*

## Konvertieren eines vorhandenen Netzes in einen Cluster

Konvertieren Sie ein existischtes verteiltes Warteschlangennetz in einen Cluster und fügen Sie einen zusätzlichen Warteschlangenmanager hinzu, um die Kapazität zu erhöhen.

### Vorbereitende Schritte

In „Neuen Cluster einrichten” auf Seite 327 über „Vollrepositorys in einen anderen WS-Manager verschieben” auf Seite 352 haben Sie einen neuen Cluster erstellt und erweitert. Die nächsten beiden Tasks untersuchen einen anderen Ansatz: die Konvertierung eines vorhandenen Netzes von Warteschlangenmanagern in einen Cluster.

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Es ist bereits ein IBM MQ-Netz vorhanden, das die landesweiten Filialen eines Filialgeschäfts verbindet. Es verfügt über eine Hub-und Speichenstruktur: Alle Warteschlangenmanager sind mit einem zentralen Warteschlangenmanager verbunden. Der zentrale WS-Manager befindet sich auf dem System, auf dem die Bestandsanwendung ausgeführt wird. Die Anwendung wird durch das Eintreffen von Nachrichten in die Warteschlange von INVENTQ gesteuert, für die jeder Warteschlangenmanager über eine Definition einer fernen Warteschlange verfügt.

  Dieses Netz ist in Abbildung 45 auf Seite 356 dargestellt.

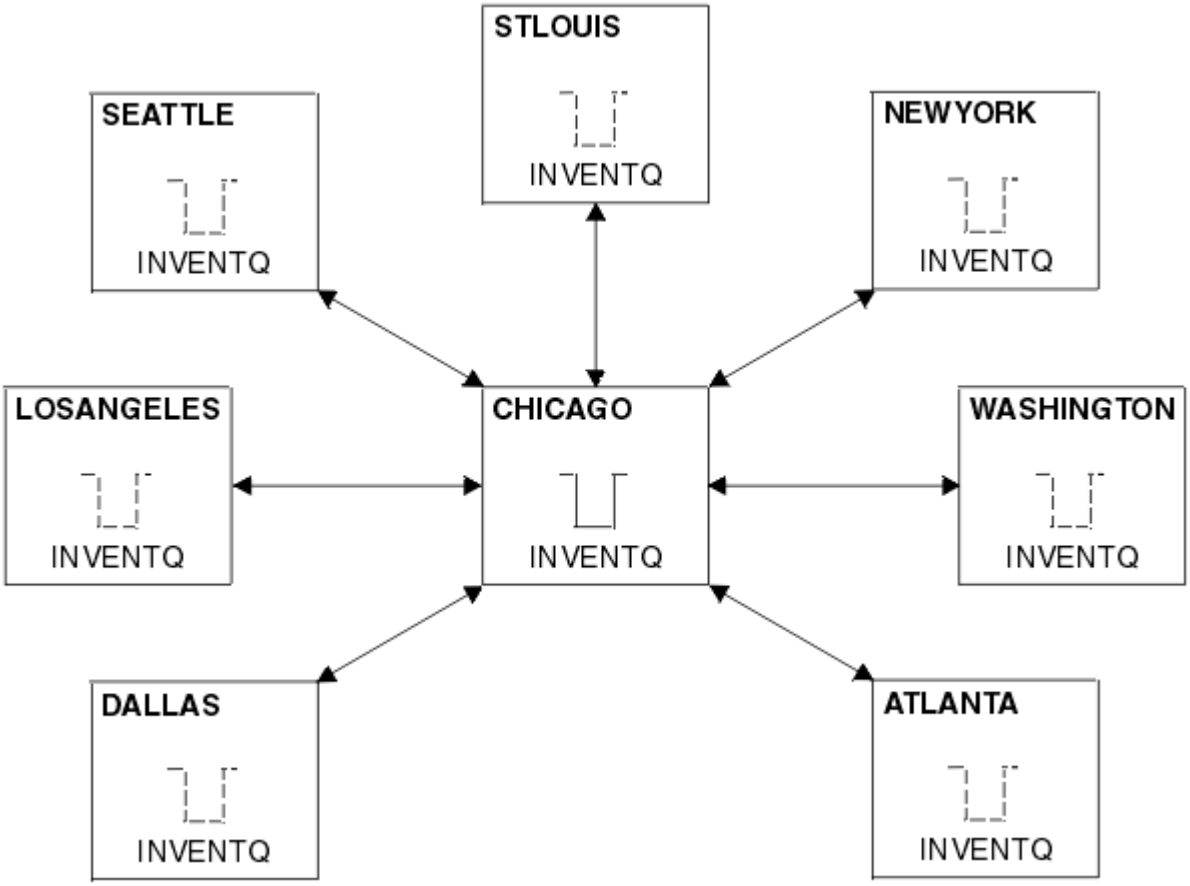



*Abbildung 45. Ein Hub-und Spoke-Netz*

- Um die Verwaltung zu vereinfachen, werden Sie dieses Netz in einen Cluster konvertieren und einen anderen Warteschlangenmanager am zentralen Standort erstellen, um die Workload gemeinsam zu nutzen.

  Der Clustername lautet CHNSTORE.

  **Anmerkung:** Der Clustername CHNSTORE wurde ausgewählt, damit Clusterempfängerkanalnamen mit Namen im Format *cluster_name. queue_manager_name* erstellt werden können, die die maximale Länge von 20 Zeichen nicht überschreiten, z. B. CHNSTORE.WASHINGTON.
- Beide zentralen Warteschlangenmanager sollen vollständige Repositorys als Host für die Bestandsanwendung verwenden.
- Die Inventaranwendung soll durch das Eintreffen von Nachrichten in die Warteschlange von INVENTQ gesteuert werden, die von einem der zentralen WS-Manager gehostet wird.
- Die Bestandsanwendung ist die einzige Anwendung, die parallel ausgeführt wird und von mehr als einem Warteschlangenmanager zugänglich ist. Alle anderen Anwendungen werden weiterhin wie zuvor ausgeführt.
- Alle Filialen verfügen über Netzkonnektivität zu den beiden zentralen WS-Managern.
- Das Netzprotokoll ist TCP.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um ein vorhandenes Netz in einen Cluster zu konvertieren.

## Vorgehensweise

1. Überprüfen Sie die Bestandsanwendung auf Nachrichtenaffinitäten.

Bevor Sie fortfahren, stellen Sie sicher, dass die Anwendung Nachrichtenaffinitäten verarbeiten kann. Nachrichtenaffinitäten sind die Beziehungen zwischen Dialognachrichten, die zwischen zwei Anwendungen ausgetauscht werden, wobei die Nachrichten von einem bestimmten WS-Manager oder in einer bestimmten Reihenfolge verarbeitet werden müssen. Weitere Informationen zu Nachrichtenaffinitäten finden Sie in „Nachrichtenaffinitäten bearbeiten" auf Seite 439.

2. Ändern Sie die beiden zentralen WS-Manager, um sie vollständig in Repository-WS-Managern zu erstellen.

   Die beiden WS-Manager CHICAGO und CHICAGO2 befinden sich im Hub dieses Netzes. Sie haben entschieden, alle Aktivitäten, die dem Filialcluster zugeordnet sind, auf diese beiden WS-Manager zu konzentrieren. Neben der Bestandsanwendung und den Definitionen für die INVENTQ -Warteschlange sollen diese WS-Manager die beiden vollständigen Repositorys für den Cluster hosten. Geben Sie an jedem der beiden Warteschlangenmanager den folgenden Befehl aus:

   ```
   ALTER QMGR REPOS(CHNSTORE)
   ```

3. Definieren Sie einen CLUSRCVR -Kanal auf jedem Warteschlangenmanager.

   Definieren Sie in jedem WS-Manager im Cluster einen Clusterempfängerkanal und einen Clustersenderkanal. Es spielt keine Rolle, welchen Kanal Sie zuerst definieren.

   Erstellen Sie eine CLUSRCVR -Definition, um jeden Warteschlangenmanager, seine Netzadresse und andere Informationen für den Cluster zugänglich zu machen. Beispiel für WS-Manager ATLANTA:

   ```
   DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)
   DESCR('Cluster-receiver channel')
   ```

4. Definieren Sie einen CLUSSDR -Kanal auf jedem WS-Manager.

   Erstellen Sie in jedem WS-Manager eine CLUSSDR -Definition, um diesen Warteschlangenmanager mit einem oder einem der vollständigen WS-Manager-Repositorys zu verbinden. Sie können z. B. ATLANTA mit CHICAGO2 verknüpfen:

   ```
   DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)
   DESCR('Cluster-sender channel to repository queue manager')
   ```

5. Installieren Sie die Bestandsanwendung unter CHICAGO2.

   Sie haben bereits die Bestandsanwendung auf WS-Manager CHICAGO. Jetzt müssen Sie eine Kopie dieser Anwendung auf WS-Manager CHICAGO2 erstellen.

6. Definieren Sie die INVENTQ -Warteschlange auf den zentralen WS-Managern.

   Ändern Sie unter CHICAGO die lokale Warteschlangendefinition für die Warteschlange INVENTQ , um die Warteschlange für den Cluster verfügbar zu machen. Geben Sie den folgenden Befehl aus:

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
   ```

   Geben Sie unter CHICAGO2 eine Definition für die gleiche Warteschlange an:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
   ```

   Unter z/OS können Sie die Option MAKEDEF der Funktion COMMAND von **CSQUTIL** verwenden, um auf CHICAGO2 eine exakte Kopie von INVENTQ auf CHICAGO zu erstellen.

   Wenn Sie diese Definitionen erstellen, wird eine Nachricht an die vollständigen Repositorys in CHICAGO und CHICAGO2 gesendet, und die darin enthaltenen Informationen werden aktualisiert. Der Warteschlangenmanager ermittelt aus den vollständigen Repositorys, wenn er eine Nachricht an den INVENTQ einreiht, dass es eine Auswahl an Zieladressen für die Nachrichten gibt.

7. Überprüfen Sie, ob die Clusteränderungen weitergegeben wurden.

Überprüfen Sie, ob die Definitionen, die Sie im vorherigen Schritt erstellt haben, durch den Cluster weitergegeben wurden. Geben Sie den folgenden Befehl in einem vollständigen WS-Manager-Repository aus:

```
DIS QCLUSTER(INVENTQ)
```

### *Hinzufügen eines neuen, miteinander verbundenen Clusters*

Fügen Sie einen neuen Cluster hinzu, der einige WS-Manager mit einem vorhandenen Cluster gemeinsam nutzt.

## Vorbereitende Schritte

**Anmerkung:**

1. Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.
2. Bevor Sie mit dieser Task beginnen, müssen Sie nach Warteschlangennamen suchen und die Auswirkungen kennen. Möglicherweise müssen Sie eine Warteschlange umbenennen oder Warteschlangenaliasnamen einrichten, bevor Sie fortfahren können.

Szenario:

- Ein IBM MQ-Cluster wurde wie in „Konvertieren eines vorhandenen Netzes in einen Cluster” auf Seite 355 beschrieben konfiguriert.
- Es soll ein neuer Cluster mit dem Namen MAILORDER implementiert werden. Dieser Cluster umfasst vier WS-Manager, die sich im CHNSTORE -Cluster befinden, CHICAGO, CHICAGO2, SEATTLE und ATLANTA sowie zwei zusätzliche Warteschlangenmanager; HARTFORD und OMAHA. Die Anwendung MAILORDER wird auf dem System in Omaha ausgeführt, das mit dem Warteschlangenmanager OMAHA verbunden ist. Es wird von den anderen Warteschlangenmanagern im Cluster gesteuert, die Nachrichten in die Warteschlange von MORDERQ stellen.
- Die vollständigen Repositorys für den MAILORDER-Cluster werden auf den beiden Warteschlangenmanagern CHICAGO and und CHICAGO2 verwaltet.
- Das Netzprotokoll ist TCP.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen neuen, miteinander verbundenen Cluster hinzuzufügen.

## Vorgehensweise

1. Erstellen Sie eine Namensliste der Clusternamen.

   Die vollständigen Repository-WS-Manager in CHICAGO und CHICAGO2 werden jetzt die vollständigen Repositorys für die beiden Cluster CHNSTORE und MAILORDER enthalten. Erstellen Sie zunächst eine Namensliste, die die Namen der Cluster enthält. Definieren Sie die Namensliste unter CHICAGO und CHICAGO2 wie folgt:

   ```
   DEFINE NAMELIST(CHAINMAIL)
   DESCR('List of cluster names')
   NAMES(CHNSTORE, MAILORDER)
   ```

2. Ändern Sie die beiden WS-Manager-Definitionen.

   Ändern Sie nun die beiden WS-Manager-Definitionen in CHICAGO und CHICAGO2. Derzeit zeigen diese Definitionen an, dass die WS-Manager vollständige Repositorys für den Cluster CHNSTORE enthalten. Ändern Sie diese Definition, um anzuzeigen, dass die WS-Manager vollständige Repositorys

für alle in der CHAINMAIL -Namensliste aufgeführten Cluster enthalten. Ändern Sie die WS-Manager-Definitionen für CHICAGO und CHICAGO2 :

```
ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
```

3. Ändern Sie die CLUSRCVR -Kanäle in CHICAGO und CHICAGO2.

   Die Kanaldefinitionen von CLUSRCVR in CHICAGO und CHICAGO2 zeigen, dass die Kanäle im Cluster CHNSTORE verfügbar sind. Sie müssen die Clusterempfängerdefinition ändern, um anzuzeigen, dass die Kanäle für alle Cluster verfügbar sind, die in der Namensliste von CHAINMAIL aufgelistet sind. Ändern Sie die Clusterempfängerdefinition in CHICAGO:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   Geben Sie in CHICAGO2 den folgenden Befehl ein:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

4. Ändern Sie die CLUSSDR -Kanäle in CHICAGO und CHICAGO2.

   Ändern Sie die beiden CLUSSDR -Kanaldefinitionen, um die Namensliste hinzuzufügen. Geben Sie in CHICAGO den folgenden Befehl ein:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   Geben Sie in CHICAGO2 den folgenden Befehl ein:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

5. Erstellen Sie eine Namensliste unter SEATTLE und ATLANTA.

   Da SEATTLE und ATLANTA Member mehrerer Cluster sein werden, müssen Sie eine Namensliste erstellen, die die Namen der Cluster enthält. Definieren Sie die Namensliste unter SEATTLE und ATLANTA wie folgt:

   ```
   DEFINE NAMELIST(CHAINMAIL)
   DESCR('List of cluster names')
   NAMES(CHNSTORE, MAILORDER)
   ```

6. Ändern Sie die CLUSRCVR -Kanäle in SEATTLE und ATLANTA.

   Die Kanaldefinitionen von CLUSRCVR in SEATTLE und ATLANTA zeigen, dass die Kanäle im Cluster CHNSTORE verfügbar sind. Ändern Sie die Clusterempfangskanaldefinitionen, um zu zeigen, dass die Kanäle für alle Cluster verfügbar sind, die in der CHAINMAIL -Namensliste aufgeführt sind. Geben Sie in SEATTLE den folgenden Befehl ein:

   ```
   ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

   Geben Sie in ATLANTA den folgenden Befehl ein:

   ```
   ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
   CLUSTER(' ') CLUSNL(CHAINMAIL)
   ```

7. Ändern Sie die CLUSSDR -Kanäle in SEATTLE und ATLANTA.

Ändern Sie die beiden CLUSSDR -Kanaldefinitionen, um die Namensliste hinzuzufügen. Geben Sie in SEATTLE den folgenden Befehl ein:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

Geben Sie in ATLANTA den folgenden Befehl ein:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

8. Definieren Sie CLUSRCVR -und CLUSSDR -Kanäle in HARTFORD und OMAHA.

   Definieren Sie auf den beiden neuen Warteschlangenmanagern HARTFORD und OMAHA Cluster-Empfänger-und Clustersenderkanäle. Es spielt keine Rolle, in welcher Reihenfolge Sie die Definitionen machen. Geben Sie in HARTFORD Folgendes ein:

   ```
   DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-receiver channel for HARTFORD')

   DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
   ```

   Geben Sie in OMAHA Folgendes ein:

   ```
   DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-receiver channel for OMAHA')

   DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
   DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
   ```

9. Definieren Sie die MORDERQ -Warteschlange unter OMAHA.

   Der letzte Schritt zum Ausführen dieser Task ist die Definition der Warteschlange MORDERQ auf dem WS-Manager OMAHA. Geben Sie in OMAHA Folgendes ein:

   ```
   DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
   ```

10. Überprüfen Sie, ob die Clusteränderungen weitergegeben wurden.

    Überprüfen Sie, ob die Definitionen, die Sie mit den vorherigen Schritten erstellt haben, durch den Cluster weitergegeben wurden. Geben Sie die folgenden Befehle in einem Repository-WS-Manager aus:

    ```
    DIS QCLUSTER (MORDERQ)
    DIS CLUSQMGR
    ```

11.

## Ergebnisse

Der Cluster, der von dieser Task konfiguriert wird, wird in Abbildung 46 auf Seite 361 angezeigt.

Jetzt haben wir zwei überlappende Cluster. Die vollständigen Repositorys für beide Cluster finden Sie unter CHICAGO und CHICAGO2. Die Anwendung für die Mail-Bestellung, die auf OMAHA ausgeführt wird, ist unabhängig von der Bestandsanwendung, die unter CHICAGO ausgeführt wird. Einige der Warteschlangenmanager, die sich im CHNSTORE -Cluster befinden, befinden sich jedoch auch im MAILORDER -Cluster, sodass sie Nachrichten an beide Anwendungen senden können. Bevor Sie diese Task zur Überlappung

von zwei Clustern durchführen können, müssen Sie die Möglichkeit von Warteschlangennamenkollisionen kennen.

Nehmen Sie an, dass auf NEWYORK im Cluster CHNSTORE und auf OMAHA in Cluster MAILORDER eine Warteschlange mit dem Namen ACCOUNTQ vorhanden ist. Wenn Sie die Cluster überlappen und eine Anwendung in SEATTLE eine Nachricht in die Warteschlange ACCOUNTQ einreiht, kann die Nachricht in eine der beiden Instanzen der ACCOUNTQ gestellt werden.

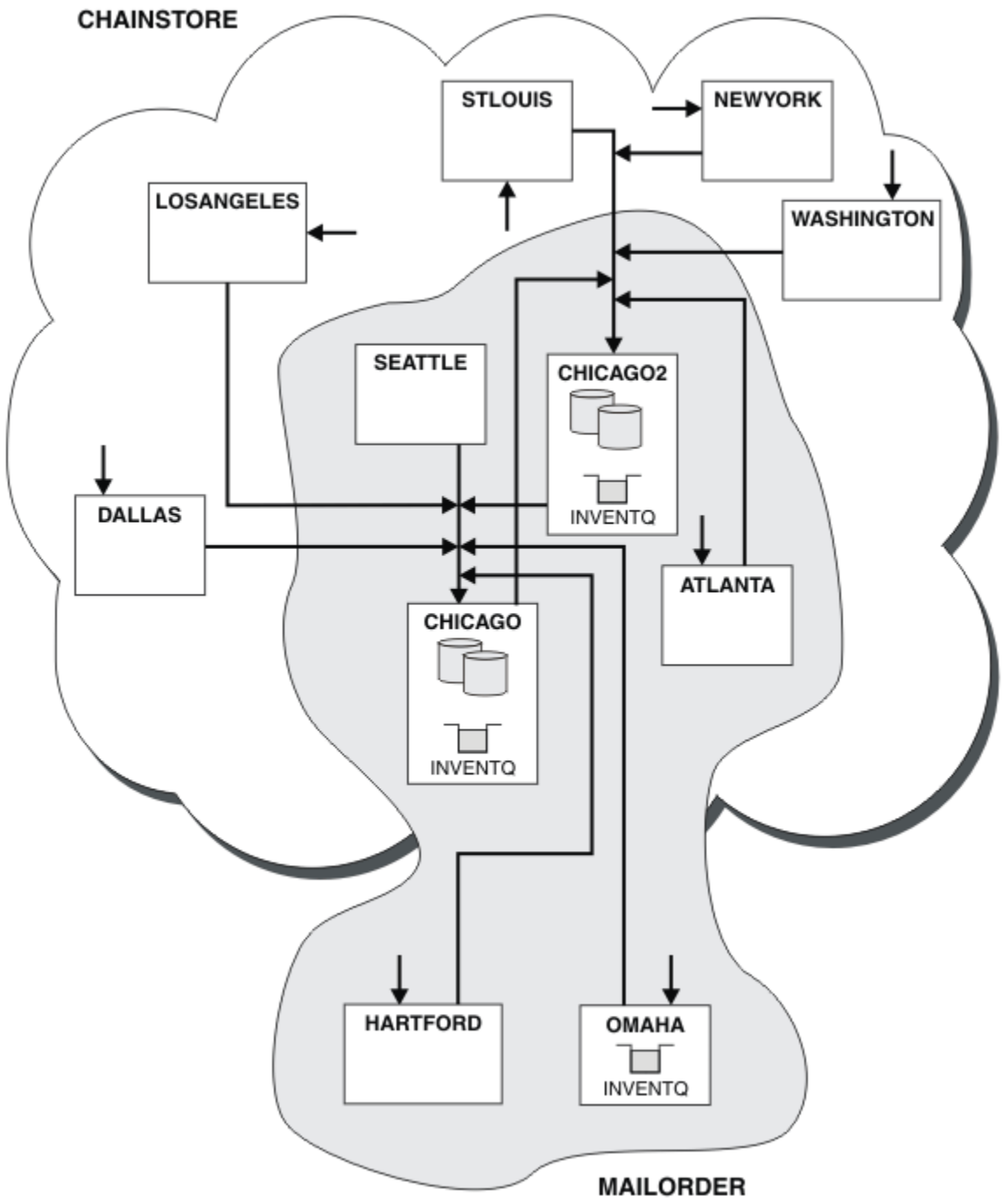



*Abbildung 46. Interverbundene Cluster*

## Nächste Schritte

Angenommen, Sie möchten den MAILORDER-Cluster mit dem CHNSTORE-Cluster zusammenführen, um einen großen Cluster mit dem Namen CHNSTORE zu bilden.

Gehen Sie wie folgt vor, um den MAILORDER -Cluster mit dem CHNSTORE -Cluster zusammenzuführen, so dass CHICAGO und CHICAGO2 die vollständigen Repositorys enthalten:

- Ändern Sie die Warteschlangenmanagerdefinitionen für CHICAGO und CHICAGO2, entfernen Sie das Attribut REPOSNL, das die Namensliste angibt ( CHAINMAIL), und ersetzen Sie es durch ein REPOS-Attribut, das den Clusternamen angibt ( CHNSTORE). Beispiel:

  ```
  ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
  ```

- Ändern Sie auf jedem WS-Manager im MAILORDER -Cluster alle Kanaldefinitionen und Warteschlangendefinitionen, um den Wert des Attributs CLUSTER von MAILORDER in CHNSTORE zu ändern. Geben Sie z. B. in HARTFORD Folgendes ein:

  ```
  ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
  ```

  Geben Sie bei OMAHA Folgendes ein:

  ```
  ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
  ```

- Ändern Sie alle Definitionen, die die Clusternamensliste CHAINMAIL angeben, also die Kanaldefinitionen CLUSRCVR und CLUSSDR in CHICAGO, CHICAGO2, SEATTLE und ATLANTA, um stattdessen den Cluster CHNSTORE anzugeben.

In diesem Beispiel sehen Sie den Vorteil der Verwendung von Namenslisten. Anstatt die WS-Manager-Definitionen für CHICAGO und CHICAGO2 zu ändern, können Sie den Wert der Namensliste CHAINMAIL ändern. Ebenso können Sie, anstatt die Kanaldefinitionen CLUSRCVR und CLUSSDR in CHICAGO, CHICAGO2, SEATTLE und ATLANTA zu ändern, das erforderliche Ergebnis erzielen, indem Sie die Namensliste ändern.

**Zugehörige Tasks**

Clusternetzwerk entfernen

Entfernen Sie einen Cluster aus einem Netz und stellen Sie die verteilte Warteschlangenkonfiguration wieder her.

## *Clusternetzwerk entfernen*

Entfernen Sie einen Cluster aus einem Netz und stellen Sie die verteilte Warteschlangenkonfiguration wieder her.

## Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Ein IBM MQ-Cluster wurde wie in „Konvertieren eines vorhandenen Netzes in einen Cluster” auf Seite 355 beschrieben konfiguriert.
- Dieser Cluster soll jetzt aus dem System entfernt werden. Das Netz von Warteschlangenmanagern soll so funktionieren, wie es vor der Implementierung des Clusters ausgeführt wurde.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um ein Clusternetz zu entfernen.

## Vorgehensweise

1. Entfernen Sie Clusterwarteschlangen aus dem CHNSTORE -Cluster.

Ändern Sie sowohl in CHICAGO als auch in CHICAGO2 die lokale Warteschlangendefinition für die Warteschlange INVENTQ , um die Warteschlange aus dem Cluster zu entfernen. Geben Sie den folgenden Befehl aus:

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

Wenn Sie die Warteschlange ändern, werden die Informationen in den vollständigen Repositorys aktualisiert und im gesamten Cluster repliziert. Aktive Anwendungen, die MQOO_BIND_NOT_FIXEDverwenden, und Anwendungen, die MQOO_BIND_AS_Q_DEF verwenden, bei denen die Warteschlange mit DEFBIND(NOTFIXED) definiert wurde, schlagen beim nächsten versuchten Aufruf MQPUT oder MQPUT1 fehl. Der Ursachencode MQRC_UNKNOWN_OBJECT_NAME wird zurückgegeben.

Sie müssen Schritt 1 nicht zuerst ausführen, aber wenn Sie dies nicht tun, führen Sie es stattdessen nach Schritt 4 aus.

2. Stoppen Sie alle Anwendungen, die Zugriff auf die Clusterwarteschlange haben.

   Stoppen Sie alle Anwendungen, die Zugriff auf Clusterwarteschlangen haben. Wenn Sie dies nicht tun, bleiben einige Clusterinformationen möglicherweise auf dem lokalen Warteschlangenmanager, wenn Sie den Cluster in Schritt 5 aktualisieren. Diese Informationen werden entfernt, wenn alle Anwendungen gestoppt wurden und die Clusterkanäle getrennt wurden.

3. Entfernen Sie das Repository-Attribut aus den vollständigen WS-Managern des Repositorys.

   Ändern Sie in CHICAGO und CHICAGO2 die WS-Manager-Definitionen, um das Repository-Attribut zu entfernen. Geben Sie dazu den folgenden Befehl aus:

   ```
   ALTER QMGR REPOS(' ')
   ```

   Die WS-Manager informieren die anderen WS-Manager im Cluster, dass sie nicht mehr die vollständigen Repositorys enthalten. Wenn die anderen WS-Manager diese Informationen empfangen, wird eine Nachricht angezeigt, die angibt, dass das vollständige Repository beendet wurde. Es wird auch eine oder mehrere Nachrichten angezeigt, die darauf hinweisen, dass für den Cluster CHNSTORE keine Repositorys mehr verfügbar sind.

4. Entfernen Sie die Clusterkanäle.

   Entfernen Sie auf CHICAGO die Clusterkanäle:

   ```
   ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')
   ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
   ```

   **Anmerkung:** Es ist wichtig, zuerst den Befehl CLUSSDR und dann den Befehl CLUSRCVR auszugeben. Geben Sie zuerst den Befehl CLUSRCVR und dann den Befehl CLUSSDR nicht aus. Dadurch werden unbestätigte Kanäle erstellt, die über den Status STOPPED verfügen. Anschließend müssen Sie einen START CHANNEL -Befehl absetzen, um die gestoppten Kanäle wiederherzustellen, z. B. START CHANNEL(CHNSTORE.CHICAGO).

   Es werden Nachrichten angezeigt, die darauf hinweisen, dass es keine Repositorys für den Cluster CHNSTORE gibt.

   Wenn Sie die Clusterwarteschlangen nicht wie in Schritt 1 beschrieben entfernt haben, tun Sie dies jetzt.

5. Stoppen Sie die Clusterkanäle.

   Stoppen Sie auf CHICAGO die Clusterkanäle mit den folgenden Befehlen:

   ```
   STOP CHANNEL(CHNSTORE.CHICAGO2)
   STOP CHANNEL(CHNSTORE.CHICAGO)
   ```

6. Wiederholen Sie die Schritte 4 und 5 für die einzelnen WS-Manager im Cluster.

7. Stoppen Sie die Clusterkanäle, und entfernen Sie anschließend alle Definitionen für die Clusterkanäle und Clusterwarteschlangen von jedem Warteschlangenmanager.
8. Optional: Löschen Sie die zwischengespeicherten Clusterinformationen, die vom WS-Manager gehalten werden.

   Obwohl die WS-Manager nicht mehr Mitglieder des Clusters sind, behalten sie jeweils eine zwischengespeicherte Kopie von Informationen zum Cluster. Wenn Sie diese Daten entfernen möchten, lesen Sie die Task „WS-Manager in den Status vor dem Cluster zurückschreiben” auf Seite 394.
9. Ferne Warteschlangendefinitionen für `INVENTQ` ersetzen

   Damit das Netz weiterhin funktionieren kann, ersetzen Sie die Definition der fernen Warteschlange für den `INVENTQ` in jedem WS-Manager.
10. Aufkreichen Sie den Cluster.

    Löschen Sie keine Warteschlangen-oder Kanaldefinitionen, die nicht mehr benötigt werden.

**Zugehörige Tasks**

Hinzufügen eines neuen, miteinander verbundenen Clusters
Fügen Sie einen neuen Cluster hinzu, der einige WS-Manager mit einem vorhandenen Cluster gemeinsam nutzt.

## Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager

Befolgen Sie die Anweisungen in der Task, um überlappende Cluster mit einem Gateway-Warteschlangenmanager zu erstellen. Verwenden Sie die Cluster als Ausgangspunkt für die folgenden Beispiele, um Nachrichten in einer Anwendung von Nachrichten an andere Anwendungen in einem Cluster zu isolieren.

### Informationen zu diesem Vorgang

In Abbildung 47 auf Seite 365 wird die Beispiel-Clusterkonfiguration gezeigt, die zur Veranschaulichung des Eingrenzung von Datenverkehr auf Clusternachrichten verwendet wird Das Beispiel wird im Abschnitt Clustering: Application isolation using multiple cluster transmission queues beschrieben.

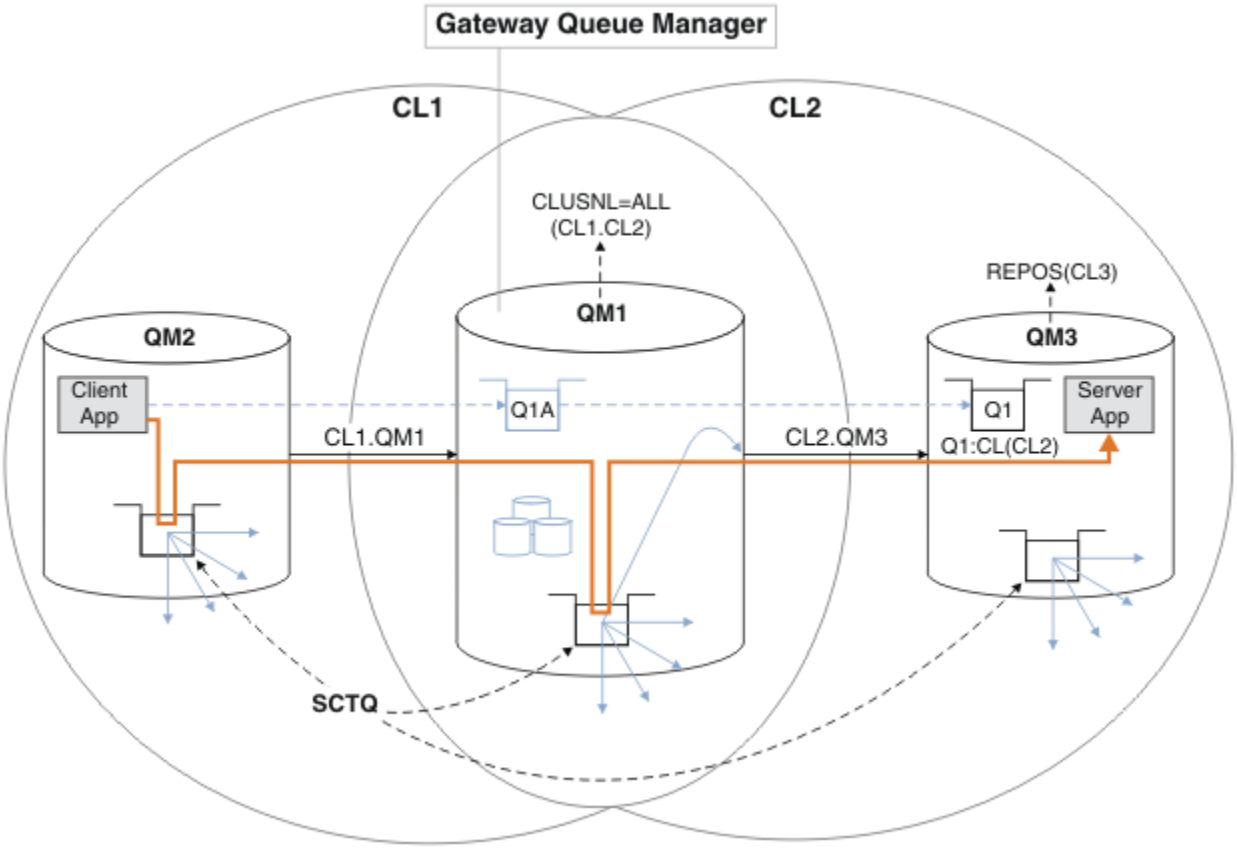



*Abbildung 47. In Hub- und Spoke-Architektur mit IBM MQ-Clustern implementierte Client/Server-Anwendung*

Um die Anzahl der Schritte zum Konstruieren des Beispiels so gering wie möglich zu halten, wird die Konfiguration einfach gehalten und nicht realistisch gehalten. Das Beispiel könnte die Integration von zwei Clustern darstellen, die von zwei separaten Organisationen erstellt wurden. Ein realistischeres Szenario finden Sie in Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen .

Führen Sie die Schritte aus, um die Cluster zu erstellen. Die Cluster werden in den folgenden Beispielen verwendet, um den Nachrichtenverkehr von der Clientanwendung auf die Serveranwendung zu isolieren.

Die Anweisungen fügen ein paar zusätzliche Warteschlangenmanager hinzu, so dass jeder Cluster über zwei Repositorys verfügt. Der Gateway-Warteschlangenmanager wird aus Leistungsgründen nicht als Repository verwendet.

## Vorgehensweise

1. Erstellen und starten Sie die Warteschlangenmanager QM1, QM2, QM3, QM4, QM5.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QM n
strmqm QmgrName
```

   **Anmerkung:** QM4 und QM5 sind die vollständigen Sicherungsrepositorys für die Cluster.

2. Definieren und starten Sie die Empfangsprogramme für jeden der Warteschlangenmanager.

```
*... On QM n
DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141 n)
```

3. Erstellen Sie eine Clusternamensliste für alle Cluster.

```
*... On QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. Erstellen Sie QM2 und QM4 vollständige Repositorys für CL1, QM3 und QM5 vollständige Repositorys für CL2.

   a) Für CL1:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

   b) Für CL2:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. Fügen Sie die Kanäle "Clustersender" und "Clusterempfänger" für jeden Warteschlangenmanager und Cluster hinzu.

   Führen Sie die folgenden Befehle unter QM2, QM3, QM4 und QM5 aus, wobei *c*, *n* und *m* die in Tabelle 26 auf Seite 366 gezeigten Werte für jeden Warteschlangenmanager annehmen:

*Tabelle 26. Parameterwerte für die Erstellung von Clustern 1 und 2*

| **Warteschlangenmanager** | **Cluster** ***c*** | **Anderes Repository** ***n*** | **Dieses Repository** ***m*** |
|---|---|---|---|
| QM2 | 1 | 4 | 2 |
| QM4 | 1 | 2 | 4 |
| QM3 | 2 | 5 | 3 |
| QM5 | 2 | 3 | 5 |

```
*... On QM m
DEFINE CHANNEL(CL c.QM n) CHLTYPE(CLUSSDR) CONNAME('localhost(141 n)') CLUSTER(CL c) REPLACE
DEFINE CHANNEL(CL c.QM m) CHLTYPE(CLUSRCVR) CONNAME('localhost(141 m)') CLUSTER(CL c) REPLACE
```

6. Fügen Sie den Gateway-Warteschlangenmanager QM1 zu jedem Cluster hinzu.

```
*... On QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. Fügen Sie die lokale Warteschlange Q1 dem Warteschlangenmanager QM3 im Cluster CL2 hinzu.

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. Fügen Sie den Cluster-WS-Manager-Aliasnamen Q1A zum Gateway-Warteschlangenmanager hinzu.

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

**Anmerkung:** Anwendungen, die den Aliasnamen des Warteschlangenmanagers auf einem anderen Warteschlangenmanager verwenden, aber QM1, müssen DEFBIND(NOTFIXED) angeben, wenn sie die Aliaswarteschlange öffnen. **DEFBIND** gibt an, ob die Routing-Informationen im Nachrichtenheader festgelegt werden, wenn die Warteschlange von der Anwendung geöffnet wird. Wenn sie auf den

Standardwert ÖFFNEN gesetzt ist, werden Nachrichten an Q1@QM1weitergeleitet. Q1@QM1 ist nicht vorhanden, sodass Nachrichten von anderen Warteschlangenmanagern in einer Warteschlange für nicht zustellbare Nachrichten enden. Wenn Sie das Warteschlangenattribut auf DEFBIND(NOTFIXED) setzen, verhalten sich Anwendungen wie **amqsput**, die standardmäßig die Warteschlangeneinstellung **DEFBIND**verwenden, ordnungsgemäß.

9. Fügen Sie die Clusterwarteschlangenmanageraliasdefinitionen für alle Clusterwarteschlangenmanager zum Gateway-Warteschlangenmanager QM1 hinzu.

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

**Tipp:** Die Warteschlangenmanager-Aliasnamendefinitionen auf dem Gateway-Warteschlangenmanager übertragen Nachrichten, die auf einen Warteschlangenmanager in einem anderen Cluster verweisen; siehe Clusterwarteschlangenmanager-Aliasnamen.

## Nächste Schritte

1. Testen Sie die Definition des Warteschlangenaliasnamens, indem Sie unter QM3 unter Verwendung der Warteschlangenaliasdefinition Q1A eine Nachricht von QM2 an Q1 senden.

   a. Führen Sie das Beispielprogramm **amqsput** unter QM2 aus, um eine Nachricht einzureihen.

   ```
   C:\IBM\MQ>amqsput Q1A QM2
   Sample AMQSPUT0 start
   target queue is Q1A
   Sample request message from QM2 to Q1 using Q1A

   Sample AMQSPUT0 end
   ```

   b. Führen Sie das Beispielprogramm **amqsget** aus, um die Nachricht von Q1 unter QM3 abzurufen

   ```
   C:\IBM\MQ>amqsget Q1 QM3
   Sample AMQSGET0 start
   message <Sample request message from QM2 to Q1 using Q1A>
   no more messages
   Sample AMQSGET0 end
   ```

2. Testen Sie die Aliasdefinitionen des Warteschlangenmanagers, indem Sie eine Anforderungsnachricht senden und eine Antwortnachricht in einer temporären Antwortwarteschlange empfangen.

   Das Diagramm zeigt den Pfad, den die Antwortnachricht zurück zu einer temporären dynamischen Warteschlange mit dem Namen RQ nimmt. Die Serveranwendung, die mit QM3 verbunden ist, öffnet die Antwortwarteschlange unter Verwendung des Warteschlangenmanagernamens QM2. Der Warteschlangenmanagername QM2 ist unter QM1 als Aliasname eines Clusterwarteschlangenmanagers definiert. QM3 leitet die Antwortnachricht an QM1 weiter. QM1 leitet die Nachricht an QM2 weiter.

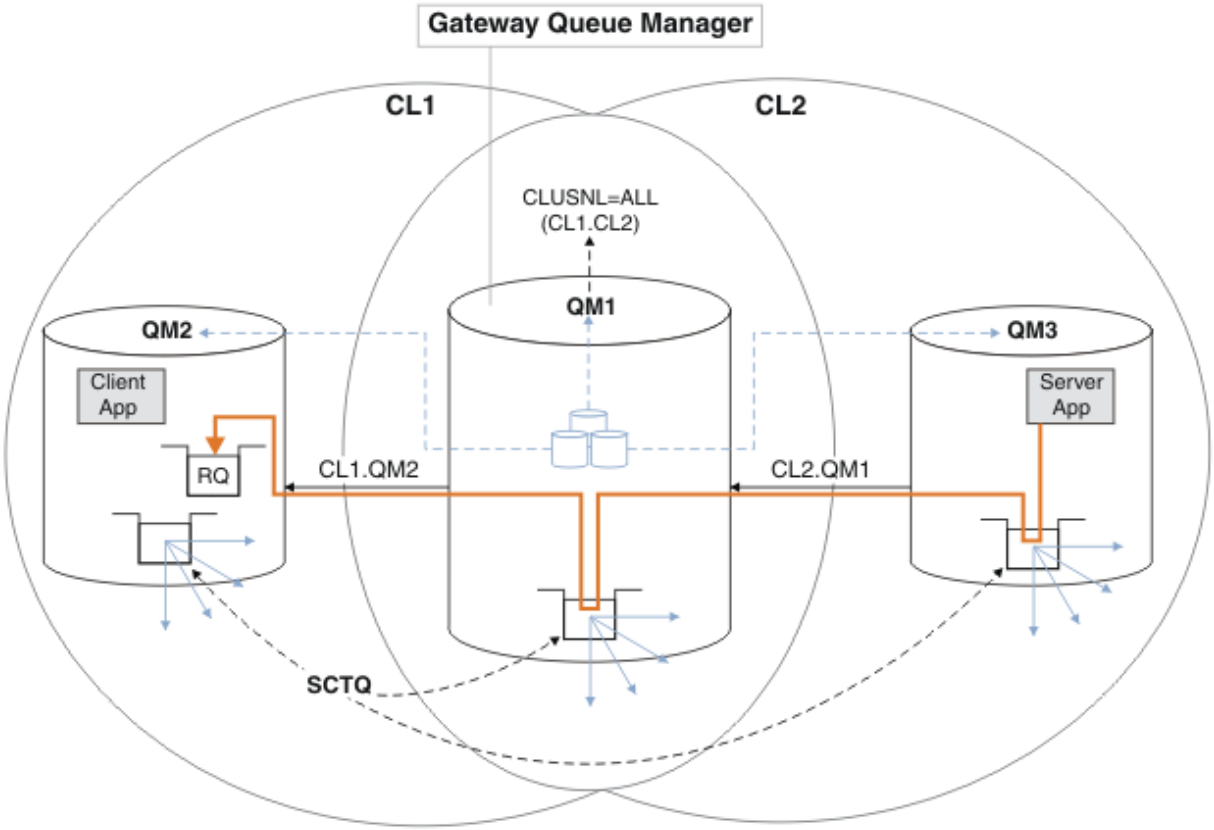



*Abbildung 48. Verwenden eines Warteschlangenmanager-Aliasnamens, um die Antwortnachricht an einen anderen Cluster zurückzugeben*

Die Art und Weise, wie das Routing funktioniert, ist wie folgt. Jeder Warteschlangenmanager in jedem Cluster verfügt über eine Warteschlangenmanager-Aliasnamensdefinition unter QM1. Die Aliasnamen werden in allen Clustern gruppiert. Die grauen gestrichelten Pfeile von jedem der Aliasnamen zu einem Warteschlangenmanager zeigen, dass jeder Warteschlangenmanager-Aliasname in mindestens einem der Cluster in einen echten Warteschlangenmanager aufgelöst wird. In diesem Fall wird der Aliasname QM2 sowohl in Cluster CL1 als auch in CL2 zusammengefasst und in den realen Warteschlangenmanager QM2 in CL1 aufgelöst. Die Serveranwendung erstellt die Antwortnachricht unter Verwendung des Namens der Empfangswarteschlange für Antworten RQ und des Namens des Antwortwarteschlangenmanagers QM2. Die Nachricht wird an QM1 weitergeleitet, da die Warteschlangenmanager-Aliasdefinition QM2 in QM1 im Cluster CL2 definiert ist und der Warteschlangenmanager QM2 nicht im Cluster CL2 ist. Da die Nachricht nicht an den Zielwarteschlangenmanager gesendet werden kann, wird sie an den Warteschlangenmanager gesendet, der die Aliasdefinition enthält.

QM1 stellt die Nachricht in die Clusterübertragungswarteschlange unter QM1 zur Übertragung an QM2. QM1 leitet die Nachricht an QM2 weiter, weil die Warteschlangenmanager-Aliasdefinition unter QM1 für QM2 QM2 als realen Zielwarteschlangenmanager definiert. Die Definition ist nicht kreisförmig, da die Aliasdefinitionen nur auf reale Definitionen verweisen können. Der Aliasname kann nicht auf sich selbst verweisen. Die reale Definition wird von QM1 aufgelöst, da sich sowohl QM1 als auch QM2 in demselben Cluster befinden: CL1. QM1 ermittelt die Verbindungsinformationen für QM2 aus dem Repository für CL1 und leitet die Nachricht an QM2 weiter. Damit die Nachricht von QM1 weitergeleitet wird, muss die Serveranwendung die Antwortwarteschlange mit der Option DEFBIND auf MQBND_BIND_NOT_FIXED geöffnet haben. Wenn die Serveranwendung die Antwortwarteschlange mit der Option MQBND_BIND_ON_OPEN geöffnet hat, wird die Nachricht nicht weitergeleitet und in eine Warteschlange für nicht zustellbare Nachrichten eingereiht.

a. Erstellen Sie eine Clusteranforderungswarteschlange mit einem Auslöser unter QM3.

```
*... On QM3
```

```
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE) PRO
CESS(ECHO) REPLACE
```

b. Erstellen Sie die Clusterwarteschlangenaliasdefinition QR auf dem Gateway-Warteschlangenmanager QM1.

```
*... On QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

c. Erstellen Sie eine Prozessdefinition, um das Beispielecho-Programm **amqsech** unter QM3zu starten.

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

d. Erstellen Sie unter QM2 eine Modellwarteschlange für das Beispielprogramm **amqsreq** , um die temporäre dynamische Antwortwarteschlange zu erstellen.

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

e. Testen Sie die Definition des WS-Manager-Aliasnamens, indem Sie eine Anforderung von QM2 an QR unter QM3 unter Verwendung der Warteschlangenaliasdefinition QRA senden.

i) Führen Sie das Auslösemonitorprogramm unter QM3 aus.

```
runmqtrm -m QM3
```

Die Ausgabe ist

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
01/02/2012  16:17:15: IBM MQ trigger monitor started.

_____________________________________________________
01/02/2012  16:17:15: Waiting for a trigger message
```

ii) Führen Sie das Beispielprogramm **amqsreq** in QM2 aus, um eine Anforderung zu stellen und warten Sie auf eine Antwort.

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3

response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

**Zugehörige Konzepte**

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

**Zugehörige Tasks**

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

### *Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden*

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine ferne Definition einer Clusterwarteschlange und einen separaten Senderkanal und eine separate Übertragungswarteschlange.

## Vorbereitende Schritte

Erstellen Sie die in In Hub-and-Spoke-Architekturmit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager" auf Seite 364 dargestellten überlappenden Cluster, indem Sie die in dieser Task genannten Schritte ausführen.

## Informationen zu diesem Vorgang

Die Lösung verwendet die verteilte Steuerung von Warteschlangen, um die Nachrichten für die `Server App` -Anwendung von einem anderen Nachrichtenverkehr auf dem Gateway-Warteschlangenmanager zu trennen. Sie müssen eine Definition einer fernen Clusterwarteschlange auf `QM1` definieren, um die Nachrichten in eine andere Übertragungswarteschlange und einen anderen Kanal umzuleiten. Die Definition der fernen Warteschlange muss einen Verweis auf die spezifische Übertragungswarteschlange enthalten, in der Nachrichten nur für `Q1` auf `QM3` gespeichert werden. In Abbildung 49 auf Seite 371 wird der Aliasname der Clusterwarteschlange `Q1A` durch eine ferne Warteschlangendefinition `Q1R` sowie eine Übertragungswarteschlange und einen Senderkanal ergänzt.

In dieser Lösung werden alle Antwortnachrichten unter Verwendung der allgemeinen `SYSTEM.CLUSTER.TRANSMIT.QUEUE` zurückgegeben.

Der Vorteil dieser Lösung ist, dass es leicht ist, den Datenverkehr für mehrere Zielwarteschlangen auf demselben WS-Manager im selben Cluster zu trennen. Der Nachteil der Lösung besteht darin, dass Sie keinen Cluster-Workload-Ausgleich zwischen mehreren Kopien von `Q1` auf verschiedenen Warteschlangenmanagern verwenden können. Informationen zum Überwinden dieses Nachteils finden Sie in „Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde" auf Seite 373. Außerdem müssen Sie den Switch von einer Übertragungswarteschlange in die andere verwalten.

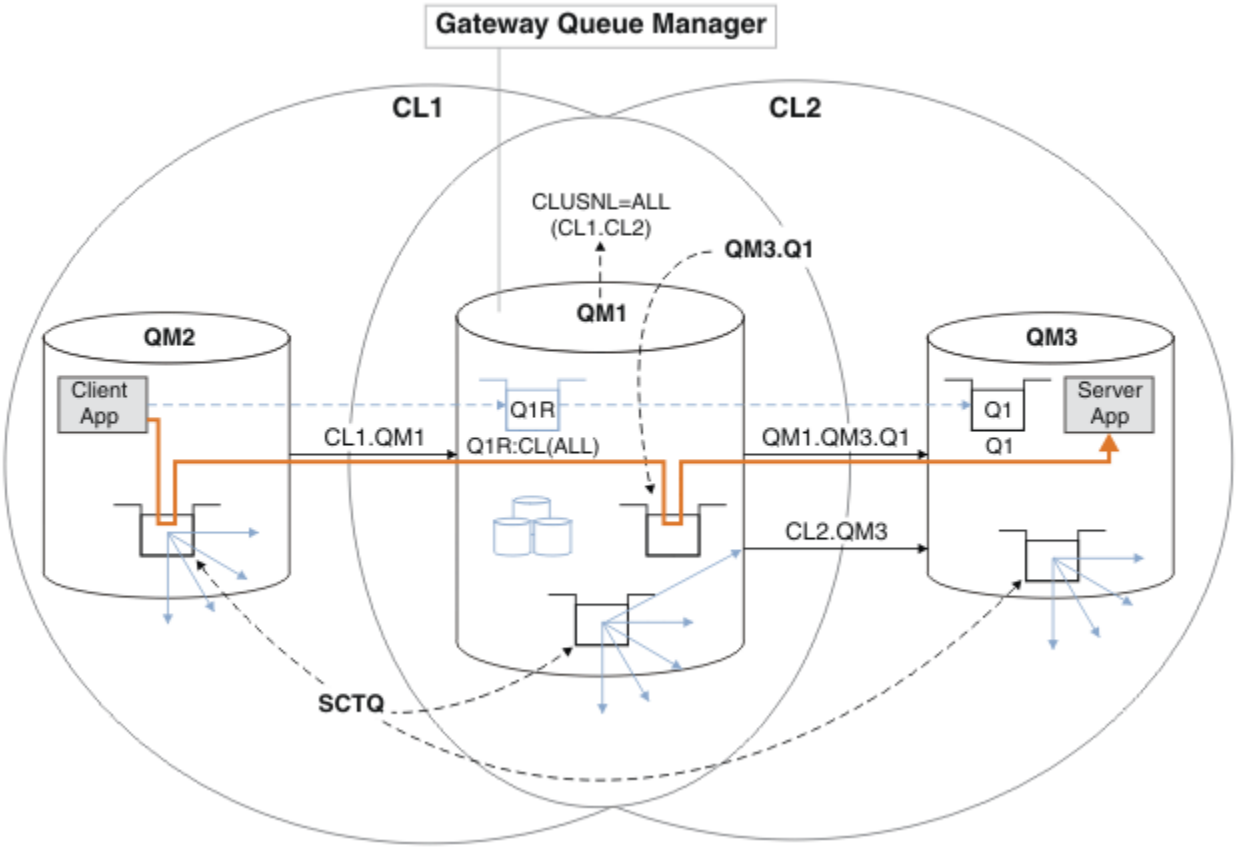



*Abbildung 49. Anwendung 'Client-Server' in Hub-und Spoke-Cluster-Architektur unter Verwendung ferner Warteschlangendefinitionen implementiert*

## Vorgehensweise

1. Erstellen Sie einen Kanal, um den Nachrichtenverkehr für Q1 vom Gateway-Warteschlangenmanager zu trennen.
   a) Erstellen Sie einen Senderkanal auf dem Gateway-Warteschlangenmanager, QM1, an den Zielwarteschlangenmanager QM3.

   ```
   DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
   ```

   b) Erstellen Sie einen Empfängerkanal auf dem Zielwarteschlangenmanager, QM3.

   ```
   DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
   ```

2. Erstellen Sie eine Übertragungswarteschlange auf dem Gateway-Warteschlangenmanager für den Nachrichtenverkehr in Q1.

   ```
   DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE
   START CHANNEL(QM1.QM3.Q1)
   ```

   Wenn der Kanal gestartet wird, der der Übertragungswarteschlange zugeordnet ist, wird die Übertragungswarteschlange dem Kanal zugeordnet. Der Kanal wird automatisch gestartet, sobald die Übertragungswarteschlange dem Kanal zugeordnet wurde.
3. Ergänzen Sie die Clusterwarteschlangenaliasdefinition für Q1 auf dem Gateway-Warteschlangenmanager mit einer Definition der fernen Clusterwarteschlange.

```
DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNAME(QM3) XMITQ(QM3.Q1) REPLACE
```

## Nächste Schritte

Testen Sie die Konfiguration, indem Sie eine Nachricht an Q1 unter QM3 von QM2 unter Verwendung der Clusterwarteschlangendefinition Q1R auf dem Gateway-WS-Manager QM1 senden.

1. Führen Sie das Beispielprogramm **amqsput** unter QM2 aus, um eine Nachricht einzureihen.

```
C:\IBM\MQ>amqsput Q1R QM2
Sample AMQSPUT0 start
target queue is Q1R
Sample request message from QM2 to Q1 using Q1R

Sample AMQSPUT0 end
```

2. Führen Sie das Beispielprogramm **amqsget** aus, um die Nachricht von Q1 unter QM3 abzurufen

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1R>
no more messages
Sample AMQSGET0 end
```

**Zugehörige Konzepte**

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

**Zugehörige Tasks**

Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine zusätzliche Clusterübertragungswarteschlange, um den Nachrichtenverkehr auf einen einzelnen Warteschlangenmanager in einem Cluster zu trennen.

Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet einen zusätzlichen Cluster, um die Nachrichten in einer bestimmten Clusterwarteschlange zu isolieren.

Ändern der Standardeinstellung in separate Clusterübertragungswarteschlangen, um den Nachrichtendatenverkehr zu isolieren

Sie können die Standardweise ändern, in der ein WS-Manager Nachrichten für eine Clusterwarteschlange oder ein Topic in einer Übertragungswarteschlange speichert. Wenn Sie den Standardwert ändern, können Sie Clusternachrichten auf einem Gateway-Warteschlangenmanager isolieren.

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

### *Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde*

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine zusätzliche Clusterübertragungswarteschlange, um den Nachrichtenverkehr auf einen einzelnen Warteschlangenmanager in einem Cluster zu trennen.

## Vorbereitende Schritte

1. Der Gateway-Warteschlangenmanager muss sich unter IBM MQbefinden.
2. Erstellen Sie die in In Hub-and-Spoke-Architekturmit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager" auf Seite 364 dargestellten überlappenden Cluster, indem Sie die in dieser Task genannten Schritte ausführen.

## Informationen zu diesem Vorgang

Fügen Sie auf dem Gateway-Warteschlangenmanager QM1 eine Übertragungswarteschlange hinzu, und legen Sie das zugehörige Warteschlangenattribut CLCHNAME fest. Setzen Sie CLCHNAME auf den Namen des Clusterempfängerkanals unter QM3 (siehe Abbildung 50 auf Seite 374).

Diese Lösung bietet eine Reihe von Vorteilen gegenüber der Lösung, die in „Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden" auf Seite 370 beschrieben wird:

- Sie erfordert weniger zusätzliche Definitionen.
- Er unterstützt den Lastausgleich zwischen mehreren Kopien der Zielwarteschlange, Q1, auf verschiedenen WS-Managern in demselben Cluster, CL2.
- Der Gateway-Warteschlangenmanager wechselt automatisch in die neue Konfiguration, wenn der Kanal neu gestartet wird, ohne dass Nachrichten zu verlieren sind.
- Der Warteschlangenmanager des Gateways setzt die Nachrichten in derselben Reihenfolge fort, in der er sie empfangen hat. Dies gilt auch dann, wenn der Switch mit Nachrichten für die Warteschlange Q1 unter QM3 noch auf SYSTEM.CLUSTER.TRANSMIT.QUEUE ausgeführt wird.

Die Konfiguration zum Isolieren des Clusternachrichtenverkehrs in Abbildung 50 auf Seite 374 führt nicht zu einer starken Isolation des Datenverkehrs als die Konfiguration, die ferne Warteschlangen in „Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden" auf Seite 370 verwendet. Wenn der Warteschlangenmanager QM3 in CL2 eine Reihe unterschiedlicher Clusterwarteschlangen und Serveranwendungen hostet, nutzen alle diese Warteschlangen den Clusterkanal CL2.QM3 und verbinden QM1 mit QM3. Die zusätzlichen Datenflüsse werden in Abbildung 50 auf Seite 374 durch den grauen Pfeil dargestellt, der den potenziellen Clusternachrichtenverkehr von der SYSTEM.CLUSTER.TRANSMIT.QUEUE zum Clustersenderkanal CL2.QM3 darstellt.

Die Abhilfe besteht darin, den WS-Manager auf die Aufnahme einer Clusterwarteschlange in einem bestimmten Cluster zu beschränken. Wenn der Warteschlangenmanager bereits eine Reihe von Clusterwarteschlangen bereitstellt, müssen Sie diese Einschränkung erfüllen, indem Sie entweder einen anderen Warteschlangenmanager erstellen oder einen anderen Cluster erstellen. Weitere Informationen finden Sie im Abschnitt „Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden" auf Seite 376.

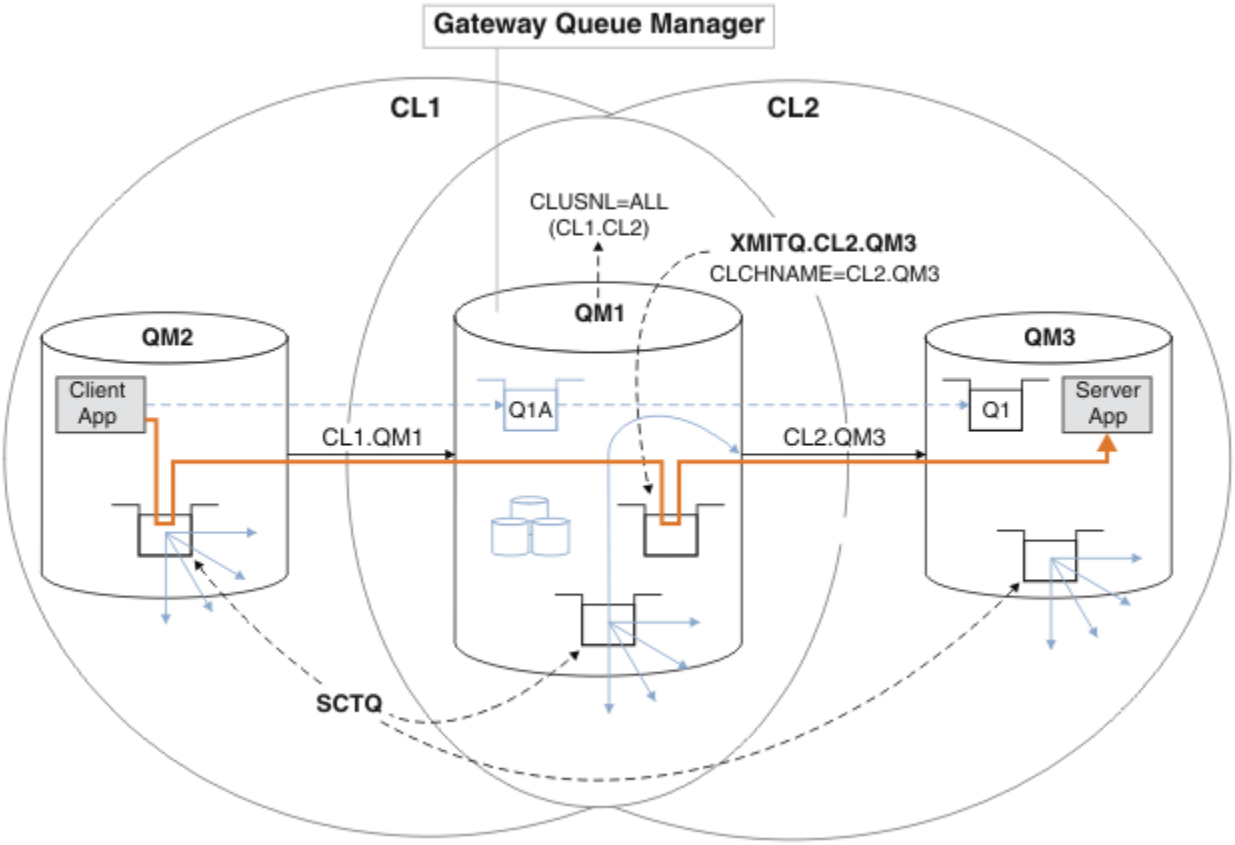



*Abbildung 50. Die Client-Server-Anwendung wurde in Hub-und Spoke-Architektur unter Verwendung einer zusätzlichen Clusterübertragungswarteschlange implementiert.*

## Vorgehensweise

1. Erstellen Sie eine zusätzliche Clusterübertragungswarteschlange für den Clustersenderkanal `CL2.QM3` auf dem Gateway-WS-Manager, `QM1`.

```
 *... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. Wechseln Sie zur Verwendung der Übertragungswarteschlange `XMITQ.CL2.QM3`.
   a) Stoppen Sie den Clustersenderkanal `CL2.QM3`.

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

   Die Antwort ist, dass der Befehl akzeptiert wird:

```
AMQ8019: Stop IBM MQ channel accepted.
```

   b) Überprüfen Sie, ob der Kanal `CL2.QM3` gestoppt wurde.

   Wenn der Kanal nicht gestoppt wird, können Sie den Befehl **STOP CHANNEL** erneut mit der Option `FORCE` ausführen. Ein Beispiel für die Einstellung der Option `FORCE` wäre, wenn der Kanal nicht gestoppt wird, und Sie können den anderen Warteschlangenmanager nicht erneut starten, um den Kanal zu synchronisieren.

```
*... On QM1
start
```

Die Antwort ist eine Zusammenfassung des Kanalstatus.

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)             CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))     CURRENT
RQMNAME(QM3)                 STATUS(STOPPED)
SUBSTATE(MQGET)              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) Starten Sie den Kanal `CL2.QM3`.

```
*... On QM1
START CHANNEL(CL2.QM3)
```

Die Antwort ist, dass der Befehl akzeptiert wird:

```
AMQ8018: Start IBM MQ channel accepted.
```

d) Überprüfen Sie, ob der Kanal gestartet wurde

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

Die Antwort ist eine Zusammenfassung des Kanalstatus:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                 CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))         CURRENT
RQMNAME(QM3)                     STATUS(RUNNING)
SUBSTATE(MQGET)                  XMITQ(XMITQ.CL2.QM3)
```

e) Überprüfen Sie, ob die Übertragungswarteschlange gewechselt wurde.

Überwachen Sie das Fehlerprotokoll des Gateway-WS-Managers für die Nachricht " AMQ7341 `Die Übertragungswarteschlange für Kanal CL2.QM3 ist XMITQ.CL2.QM3 `".

## Nächste Schritte

Testen Sie die separate Übertragungswarteschlange, indem Sie eine Nachricht von `QM2` an `Q1` auf `QM3` unter Verwendung der Warteschlangenaliasdefinition `Q1A` senden.

1. Führen Sie das Beispielprogramm **amqsput** unter `QM2` aus, um eine Nachricht einzureihen.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. Führen Sie das Beispielprogramm **amqsget** aus, um die Nachricht von `Q1` unter `QM3` abzurufen

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

**Zugehörige Konzepte**

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

„Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen" auf Seite 318
Nachrichten zwischen Clustering-WS-Managern werden in Clusterübertragungswarteschlangen gespeichert und von Clustersenderkanälen weitergeleitet. Zu einem beliebigen Zeitpunkt ist ein Clustersenderkanal einer Übertragungswarteschlange zugeordnet. Wenn Sie die Konfiguration des Kanals ändern, kann es beim nächsten Start zu einer anderen Übertragungswarteschlange wechseln. Die Verarbeitung dieses Switches ist automatisiert und transaktionsorientiert.

**Zugehörige Tasks**

Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden
Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine ferne Definition einer Clusterwarteschlange und einen separaten Senderkanal und eine separate Übertragungswarteschlange.

Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden
Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet einen zusätzlichen Cluster, um die Nachrichten in einer bestimmten Clusterwarteschlange zu isolieren.

Ändern der Standardeinstellung in separate Clusterübertragungswarteschlangen, um den Nachrichtendatenverkehr zu isolieren
Sie können die Standardweise ändern, in der ein WS-Manager Nachrichten für eine Clusterwarteschlange oder ein Topic in einer Übertragungswarteschlange speichert. Wenn Sie den Standardwert ändern, können Sie Clusternachrichten auf einem Gateway-Warteschlangenmanager isolieren.

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen" auf Seite 342
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

## *Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden*

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet einen zusätzlichen Cluster, um die Nachrichten in einer bestimmten Clusterwarteschlange zu isolieren.

## Vorbereitende Schritte

Die Schritte in der Task werden geschrieben, um die in Abbildung 50 auf Seite 374 dargestellte Konfiguration zu ändern.

1. Der Gateway-Warteschlangenmanager muss sich unter IBM MQbefinden.
2. Erstellen Sie die in In Hub-and-Spoke-Architekturmit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager" auf Seite 364 dargestellten überlappenden Cluster, indem Sie die in dieser Task genannten Schritte ausführen.

3. Führen Sie die Schritte unter Abbildung 50 auf Seite 374 in „Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde" auf Seite 373 aus, um die Lösung ohne den zusätzlichen Cluster zu erstellen. Verwenden Sie diese Option als Basis für die Schritte in dieser Task.

## Informationen zu diesem Vorgang

Die Lösung zum Eingrenzen des Nachrichtenverkehrs auf eine einzelne Anwendung in „Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde" auf Seite 373 funktioniert, wenn die Ziel-Clusterwarteschlange die einzige Clusterwarteschlange auf einem Warteschlangenmanager ist. Wenn dies nicht der Fall ist, haben Sie zwei Möglichkeiten. Verschieben Sie die Warteschlange entweder in einen anderen Warteschlangenmanager oder erstellen Sie einen Cluster, der die Warteschlange aus anderen Clusterwarteschlangen auf dem Warteschlangenmanager isoliert.

Diese Task führt Sie durch die Schritte zum Hinzufügen eines Clusters zum Isolieren der Zielwarteschlange. Der Cluster wird zu diesem Zweck hinzugefügt. In der Praxis ist es die Aufgabe, bestimmte Anwendungen systematisch zu isolieren, wenn Sie Cluster-und Clusterbenennungsschemata entwerfen. Wenn Sie einen Cluster jedes Mal hinzufügen, wenn eine Warteschlange isoliert wird, kann die Isolation möglicherweise mit vielen Clustern enden. In dieser Task ändern Sie die Konfiguration in „Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde" auf Seite 373 , indem Sie einen Cluster `CL3` hinzufügen, um `Q1` auf `QM3` zu isolieren. Anwendungen werden während der gesamten Änderung weiterhin ausgeführt.

Die neuen und geänderten Definitionen werden in Abbildung 51 auf Seite 378 hervorgehoben. Die Zusammenfassung der Änderungen lautet wie folgt: Erstellen Sie einen Cluster. Dies bedeutet, dass Sie auch ein neues vollständiges Clusterrepository erstellen müssen. Im Beispiel wird `QM3` zu einem der vollständigen Repositorys für `CL3` gemacht. Erstellen Sie Clustersenderkanäle und Clusterempfängerkanäle für `QM1` , um den Gateway-WS-Manager dem neuen Cluster hinzuzufügen. Ändern Sie die Definition von `Q1` , um sie in `CL3` zu wechseln. Ändern Sie die Clusternamensliste auf dem Gateway-Warteschlangenmanager, und fügen Sie eine Clusterübertragungswarteschlange hinzu, um den neuen Clusterkanal zu verwenden. Schalten Sie schließlich den Warteschlangenalias `Q1A` in die neue Clusternamensliste ein.

IBM MQ kann keine Nachrichten aus der Übertragungswarteschlange `XMITQ.CL2.QM3`, die Sie in „Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde" auf Seite 373 hinzugefügt haben, automatisch in die neue Übertragungswarteschlange `XMITQ.CL3.QM3` übertragen. Es kann Nachrichten nur dann automatisch übertragen, wenn beide Übertragungswarteschlangen vom selben Clustersenderkanal bedient werden. Stattdessen beschreibt die Task eine Möglichkeit, den Switch manuell auszuführen. Dies kann für Sie geeignet sein. Wenn die Übertragung abgeschlossen ist, haben Sie die Möglichkeit, die Standard-Clusterübertragungswarteschlange für andere `CL2` -Clusterwarteschlangen unter `QM3` zu verwenden. Oder Sie können `XMITQ.CL2.QM3` weiterhin verwenden. Wenn Sie sich dafür entscheiden, auf eine Standard-Cluster-Übertragungswarteschlange zurückzusetzen, verwaltet der Gateway-Warteschlangenmanager den Switch automatisch für Sie.

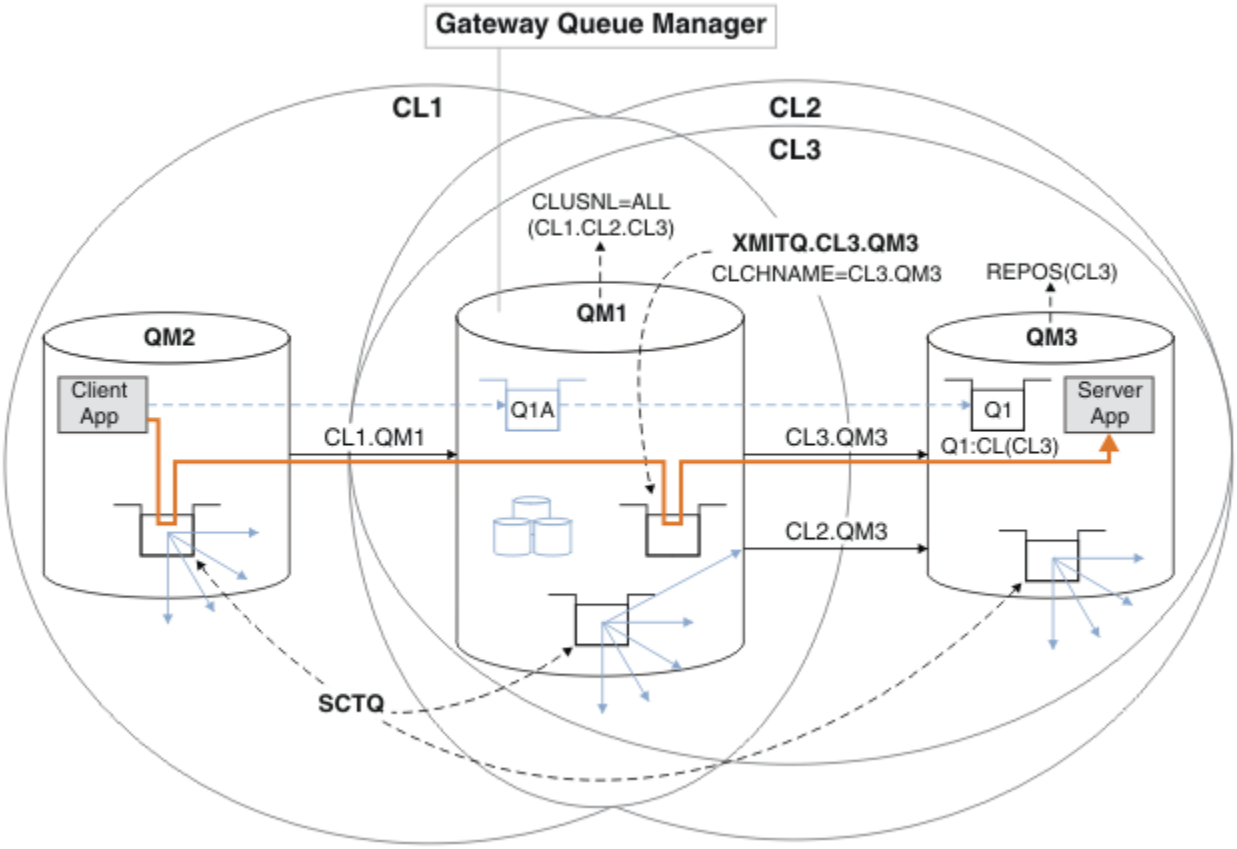



*Abbildung 51. Verwenden eines zusätzlichen Clusters zum Trennen des Nachrichtenverkehrs im Gateway-Warteschlangenmanager, der zu einer Anzahl von Clusterwarteschlangen auf demselben Warteschlangenmanager wechselt*

## Vorgehensweise

1. Ändern Sie die Warteschlangenmanager QM3 und QM5 , um sie sowohl für CL2 als auch für CL3 zu erstellen.

   Um einen Warteschlangenmanager zu einem Member mehrerer Cluster zu machen, muss er eine Clusternamensliste verwenden, um die Cluster zu identifizieren, zu der er gehört.

   ```
   *... On QM3 and QM5
   DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
   ALTER QMGR REPOS(' ') REPOSNL(CL23)
   ```

2. Definieren Sie die Kanäle zwischen den Warteschlangenmanagern QM3 und QM5 für CL3.

   ```
   *... On QM3
   DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

   *... On QM5
   DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
   DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
   ```

3. Fügen Sie den Gateway-Warteschlangenmanager zu CL3 hinzu.

   Fügen Sie den Gateway-WS-Manager hinzu, indem Sie QM1 als Teilrepository zu CL3 hinzufügen. Erstellen Sie ein Teilrepository, indem Sie Clustersenderkanäle und Clusterempfängerkanäle zu QM1 hinzufügen.

   Fügen Sie außerdem CL3 zur Namensliste aller Cluster hinzu, die mit dem Gateway-WS-Manager verbunden sind.

```
*... On QM1
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
```

4. Fügen Sie eine Clusterübertragungswarteschlange für Nachrichten an CL3 unter QM3zum Gateway-Warteschlangenmanager QM1hinzu.

   Stoppen Sie zunächst den Clustersenderkanal, der Nachrichten aus der Übertragungswarteschlange überträgt, bis Sie bereit sind, Übertragungswarteschlangen zu wechseln.

   ```
   *... On QM1
   DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
   ```

5. Bereinigen Sie Nachrichten aus der vorhandenen Clusterübertragungswarteschlange XMITQ.CL2.QM3.

   Diese Unterprozedur dient dazu, die Reihenfolge der Nachrichten in Q1 beizubehalten, damit sie mit der Reihenfolge übereinstimmt, in der sie im Gateway-Warteschlangenmanager angekommen sind. Bei Clustern ist die Nachrichtenreihenfolge nicht vollständig gewährleistet, ist aber wahrscheinlich. Wenn eine garantierte Nachrichtenreihenfolge erforderlich ist, müssen Anwendungen die Reihenfolge der Nachrichten definieren. Weitere Informationen finden Sie in Die Reihenfolge, in der Nachrichten aus einer Warteschlange abgerufen werden .

   a) Ändern Sie die Zielwarteschlange Q1 in QM3 von CL2 in CL3.

      ```
      *... On QM3
      ALTER QLOCAL(Q1) CLUSTER(CL3)
      ```

   b) Überwachen Sie XMITQ.CL3.QM3 , bis Nachrichten gestartet werden, die an ihn gesendet werden.

      Die Zustellung von Nachrichten an XMITQ.CL3.QM3 wird gestartet, wenn der Wechsel von Q1 in CL3 an den Gateway-Warteschlangenmanager weitergegeben wird.

      ```
      *... On QM1
      DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
      ```

   c) Überwachen Sie XMITQ.CL2.QM3 , bis keine Nachrichten vorhanden sind, die auf die Zustellung an Q1 auf QM3 warten.

      **Anmerkung:** XMITQ.CL2.QM3 speichert möglicherweise Nachrichten für andere Warteschlangen in QM3 , die Mitglieder von CL2 sind. In diesem Fall wird die Tiefe möglicherweise nicht auf null gehen.

      ```
      *... On QM1
      DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
      ```

   d) Abrufen aus der neuen Clusterübertragungswarteschlange aktivieren, XMITQ.CL3.QM3

      ```
      *... On QM1
      ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
      ```

6. Entfernen Sie die alte Clusterübertragungswarteschlange XMITQ.CL2.QM3, falls sie nicht mehr benötigt wird.

   Nachrichten für Clusterwarteschlangen in CL2 auf QM3 werden auf die Verwendung der Standard-Cluster-Übertragungswarteschlange auf dem Gateway-WS-Manager QM1 zurückgesetzt. Die Standard-Cluster-Übertragungswarteschlange ist entweder SYSTEM.CLUSTER.TRANSMIT.QUEUE oder SYSTEM.CLUSTER.TRANSMIT.CL2.QM3. Welche davon abhängt, ob der Wert des Warteschlangenmanagerattributs **DEFCLXQ** in QM1 SCTQ oder CHANNEL ist. Der Warteschlangenmanager überträgt Nachrichten automatisch von XMITQ.CL2.QM3 , wenn der Clustersenderkanal CL2.QM3 als Nächstes gestartet wird.

   a) Ändern Sie die Übertragungswarteschlange XMITQ.CL2.QM3 aus einer Cluster-Übertragungswarteschlange, um eine normale Übertragungswarteschlange zu sein.

Dadurch wird die Zuordnung der Übertragungswarteschlange zu beliebigen Clustersenderkanälen unterbrochen. Im Gegenzug überträgt IBM MQ Nachrichten automatisch von XMITQ.CL2.QM3 in die Standard-Cluster-Übertragungswarteschlange, wenn der Clustersenderkanal beim nächsten Mal gestartet wird. Bis dahin werden die Nachrichten für CL2 unter QM3 weiterhin auf XMITQ.CL2.QM3 gesetzt.

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME(' ')
```

b) Stoppen Sie den Clustersenderkanal CL2.QM3.

Wenn Sie den Clustersenderkanal stoppen und erneut starten, wird die Übertragung von Nachrichten von XMITQ.CL2.QM3 in die Standardübertragungswarteschlange des Clusters eingeleitet. In der Regel müssen Sie den Kanal manuell stoppen und starten, um die Übertragung zu starten. Die Übertragung wird automatisch gestartet, wenn der Kanal nach dem Abschließen des Unterbrechungsintervalls erneut gestartet wird.

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

Die Antwort ist, dass der Befehl akzeptiert wird:

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) Überprüfen Sie, ob der Kanal CL2.QM3 gestoppt wurde.

Wenn der Kanal nicht gestoppt wird, können Sie den Befehl **STOP CHANNEL** erneut mit der Option FORCE ausführen. Ein Beispiel für die Einstellung der Option FORCE wäre, wenn der Kanal nicht gestoppt wird, und Sie können den anderen Warteschlangenmanager nicht erneut starten, um den Kanal zu synchronisieren.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

Die Antwort ist eine Zusammenfassung des Kanalstatus.

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))                CURRENT
RQMNAME(QM3)                            STATUS(STOPPED)
SUBSTATE(MQGET)                         XMITQ(XMITQ.CL2.QM3)
```

d) Starten Sie den Kanal CL2.QM3.

```
*... On QM1
START CHANNEL(CL2.QM3)
```

Die Antwort ist, dass der Befehl akzeptiert wird:

```
AMQ8018: Start IBM MQ channel accepted.
```

e) Überprüfen Sie, ob der Kanal gestartet wurde

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

Die Antwort ist eine Zusammenfassung des Kanalstatus:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)      CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))  CURRENT
RQMNAME(QM3)        STATUS(RUNNING)
SUBSTATE(MQGET)      XMITQ(SYSTEM.CLUSTER.TRANSMIT. QUEUE|CL2.QM3)
```

f) Überwachen Sie das Fehlerprotokoll des Gateway-WS-Managers für die Nachricht " AMQ7341 Die Übertragungswarteschlange für Kanal CL2.QM3 ist SYSTEM.CLUSTER.TRANSMIT. *QUEUE|CL2.QM3* ".

g) Löschen Sie die Clusterübertragungswarteschlange XMITQ.CL2.QM3.

```
*... On QM1
DELETE QLOCAL(XMITQ.CL2.QM3)
```

## Nächste Schritte

Testen Sie die separate Clusterwarteschlange, indem Sie eine Nachricht von QM2 an Q1 auf QM3 unter Verwendung der Warteschlangenaliasdefinition Q1A senden.

1. Führen Sie das Beispielprogramm **amqsput** unter QM2 aus, um eine Nachricht einzureihen.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. Führen Sie das Beispielprogramm **amqsget** aus, um die Nachricht von Q1 unter QM3 abzurufen

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

**Zugehörige Konzepte**

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

„Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen” auf Seite 318

Nachrichten zwischen Clustering-WS-Managern werden in Clusterübertragungswarteschlangen gespeichert und von Clustersenderkanälen weitergeleitet. Zu einem beliebigen Zeitpunkt ist ein Clustersenderkanal einer Übertragungswarteschlange zugeordnet. Wenn Sie die Konfiguration des Kanals ändern, kann es beim nächsten Start zu einer anderen Übertragungswarteschlange wechseln. Die Verarbeitung dieses Switches ist automatisiert und transaktionsorientiert.

**Zugehörige Tasks**

Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine ferne Definition einer Clusterwarteschlange und einen separaten Senderkanal und eine separate Übertragungswarteschlange.

Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine zusätzliche Clusterübertragungswarteschlange, um den Nachrichtenverkehr auf einen einzelnen Warteschlangenmanager in einem Cluster zu trennen.

Ändern der Standardeinstellung in separate Clusterübertragungswarteschlangen, um den Nachrichtendatenverkehr zu isolieren
Sie können die Standardweise ändern, in der ein WS-Manager Nachrichten für eine Clusterwarteschlange oder ein Topic in einer Übertragungswarteschlange speichert. Wenn Sie den Standardwert ändern, können Sie Clusternachrichten auf einem Gateway-Warteschlangenmanager isolieren.

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

## *Ändern der Standardeinstellung in separate Clusterübertragungswarteschlangen, um den Nachrichtendatenverkehr zu isolieren*

Sie können die Standardweise ändern, in der ein WS-Manager Nachrichten für eine Clusterwarteschlange oder ein Topic in einer Übertragungswarteschlange speichert. Wenn Sie den Standardwert ändern, können Sie Clusternachrichten auf einem Gateway-Warteschlangenmanager isolieren.

### Vorbereitende Schritte

1. Der Gateway-Warteschlangenmanager muss sich unter IBM MQbefinden.
2. Erstellen Sie die in In Hub-and-Spoke-Architekturmit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager” auf Seite 364 dargestellten überlappenden Cluster, indem Sie die in dieser Task genannten Schritte ausführen.

### Informationen zu diesem Vorgang

Um die Architektur mit mehreren Clusterwarteschlangen zu implementieren, muss sich Ihr Gateway-Warteschlangenmanager in IBM MQbefinden. Alle für die Verwendung mehrerer Clusterübertragungswarteschlangen verwendeten Warteschlangen müssen den Standardwarteschlangentyp der Clusterübertragung im Gateway-Warteschlangenmanager ändern. Ändern Sie den Wert des Warteschlangenmanagerattributs **DEFCLXQ** unter `QM1` von `SCTQ` in `CHANNEL` . Siehe Abbildung 52 auf Seite 383. Das Diagramm zeigt einen Nachrichtenfluss. Für Datenflüsse zu anderen Warteschlangenmanagern oder zu anderen Clustern erstellt der Warteschlangenmanager zusätzliche permanente dynamische Clusterübertragungswarteschlangen. Jeder Clustersenderkanal überträgt Nachrichten aus einer anderen Clusterübertragungswarteschlange.

Die Änderung wird nicht sofort wirksam, es sei denn, Sie verbinden den Gateway-WS-Manager zum ersten Mal mit Clustern. Die Task enthält Schritte für den typischen Fall, dass eine Änderung an einer vorhandenen Konfiguration verwaltet wird. Informationen zum Festlegen eines Warteschlangenmanagers für die Verwendung separater Clusterübertragungswarteschlangen beim ersten Joins eines Clusters finden Sie in „WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342.

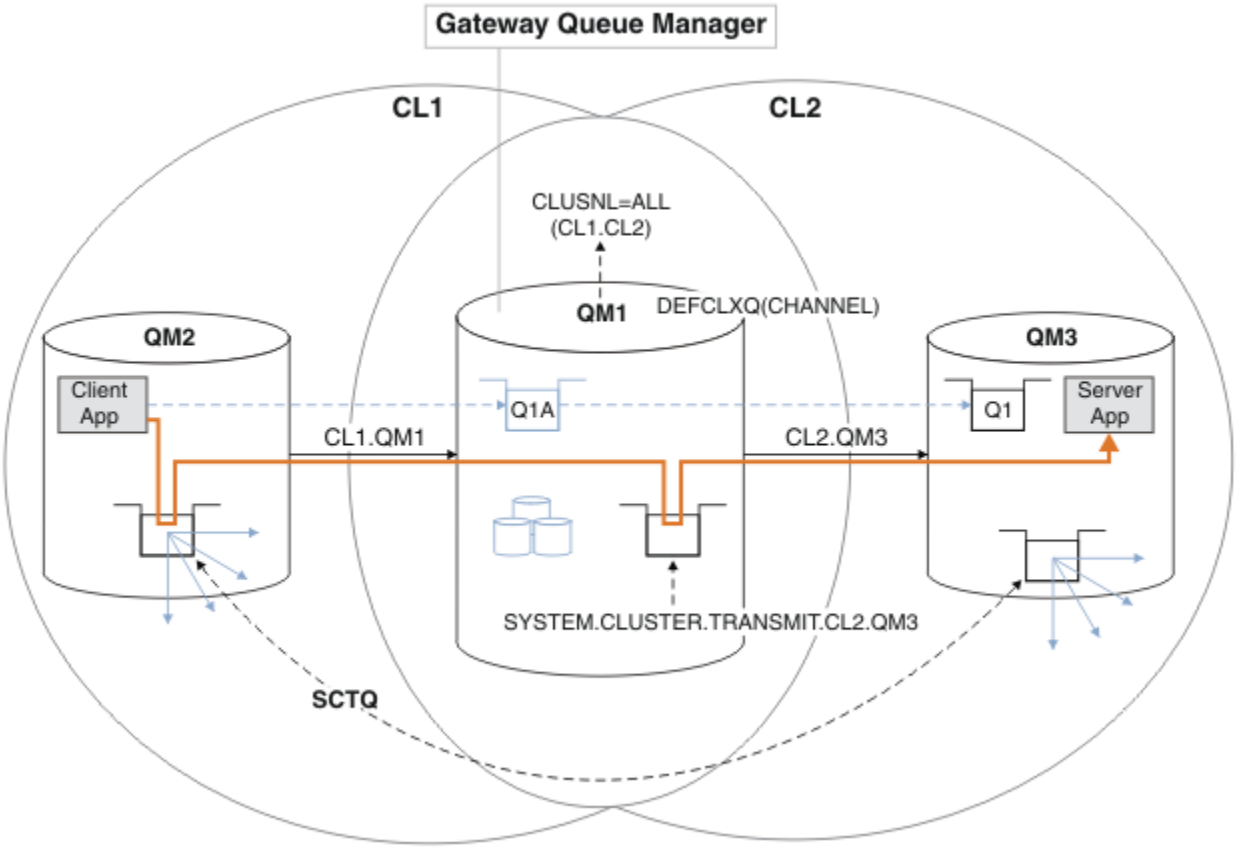



*Abbildung 52. Die Client-Server-Anwendung, die in Hub-und Spoke-Architektur implementiert ist, mit separaten Clusterübertragungswarteschlangen auf dem Gateway-Warteschlangenmanager.*

## Vorgehensweise

1. Ändern Sie den Gateway-WS-Manager, um separate Clusterübertragungswarteschlangen zu verwenden.

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. Wechseln Sie zu den separaten Clusterübertragungswarteschlangen.

   Jeder Clustersenderkanal, der keine Switches ausführt, um separate Clusterübertragungswarteschlangen zu verwenden, wenn er als Nächstes gestartet wird.

   Um die aktiven Kanäle umzuschalten, müssen Sie entweder den Warteschlangenmanager erneut starten oder die folgenden Schritte ausführen:

   a) Listen Sie die Clustersenderkanäle auf, die mit `SYSTEM.CLUSTER.TRANSMIT.QUEUE` ausgeführt werden.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

   Die Antwort ist eine Liste der Kanalstatusberichte:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))    CURRENT
RQMNAME(QM2)                STATUS(RUNNING)
```

```
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))    CURRENT
RQMNAME(QM3)                STATUS(RUNNING)
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))    CURRENT
RQMNAME(QM5)                STATUS(RUNNING)
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)            CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))    CURRENT
RQMNAME(QM4)                STATUS(RUNNING)
SUBSTATE(MQGET)             XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

b) Kanäle stoppen, die ausgeführt werden

Führen Sie für jeden Kanal in der Liste den folgenden Befehl aus:

```
*... On QM1
STOP CHANNEL(ChannelName)
```

Dabei ist *ChannelName* jeder `CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5`.

Die Antwort ist, dass der Befehl akzeptiert wird:

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) Überwachen, welche Kanäle gestoppt sind

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

Die Antwort ist eine Liste der Kanäle, die noch aktiv sind, und Kanäle, die gestoppt wurden:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)           CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))   CURRENT
RQMNAME(QM2)               STATUS(STOPPED)
SUBSTATE( )                XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))   CURRENT
RQMNAME(QM3)               STATUS(STOPPED)
SUBSTATE( )                XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)           CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))   CURRENT
RQMNAME(QM5)               STATUS(STOPPED)
SUBSTATE( )                XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)           CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))   CURRENT
RQMNAME(QM4)               STATUS(STOPPED)
SUBSTATE( )                XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

d) Starten Sie jeden gestoppten Kanal.

Führen Sie diesen Schritt für alle Kanäle aus, die ausgeführt wurden. Wenn ein Kanal nicht gestoppt wird, kann der Befehl **STOP CHANNEL** mit der Option FORCE erneut ausgeführt werden. Ein Beispiel für die Einstellung der Option FORCE wäre, wenn der Kanal nicht gestoppt wird, und Sie können den anderen Warteschlangenmanager nicht erneut starten, um den Kanal zu synchronisieren.

```
*... On QM1
START CHANNEL(CL2.QM5)
```

Die Antwort ist, dass der Befehl akzeptiert wird:

```
AMQ8018: Start IBM MQ channel accepted.
```

e) Überwachen Sie die Übertragungswarteschlangen, die umgeschaltet werden.

Überwachen Sie das Fehlerprotokoll des Gateway-WS-Managers für die Nachricht " AMQ7341 Die Übertragungswarteschlange für Kanal CL2.QM3 ist SYSTEM.CLUSTER.TRANSMIT.*QUEUE*|*CL2.QM3* ".

f) Vergewissern Sie sich, dass SYSTEM.CLUSTER.TRANSMIT.QUEUE nicht mehr verwendet wird.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

Die Antwort ist eine Liste der Kanalstatusberichte und die Tiefe von SYSTEM.CLUSTER.TRANSMIT.QUEUE:

```
AMQ8420: Channel Status not found.
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)    TYPE(QLOCAL)
CURDEPTH(0)
```

g) Kanäle überwachen, die gestartet werden

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

Die Antwort ist eine Liste der Kanäle, die in diesem Fall bereits mit den neuen Standard-Cluster-Übertragungswarteschlangen ausgeführt werden:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))                CURRENT
RQMNAME(QM2)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))                CURRENT
RQMNAME(QM3)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))                CURRENT
RQMNAME(QM5)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
```

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)                        CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))                CURRENT
RQMNAME(QM4)                            STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

## Nächste Schritte

1. Testen Sie die automatisch definierte Clusterübertragungswarteschlange, indem Sie eine Nachricht von QM2 an Q1 unter QM3 senden und den Warteschlangennamen mit der Warteschlangenaliasdefinition Q1A auflösen.

   a. Führen Sie das Beispielprogramm **amqsput** unter QM2 aus, um eine Nachricht einzureihen.

   ```
   C:\IBM\MQ>amqsput Q1A QM2
   Sample AMQSPUT0 start
   target queue is Q1A
   Sample request message from QM2 to Q1 using Q1A

   Sample AMQSPUT0 end
   ```

   b. Führen Sie das Beispielprogramm **amqsget** aus, um die Nachricht von Q1 unter QM3 abzurufen

   ```
   C:\IBM\MQ>amqsget Q1 QM3
   Sample AMQSGET0 start
   message <Sample request message from QM2 to Q1 using Q1A>
   no more messages
   Sample AMQSGET0 end
   ```

2. Überlegen Sie, ob die Sicherheit durch die Konfiguration der Sicherheit für die Clusterwarteschlangen auf den Warteschlangenmanagern, in denen Nachrichten für die Clusterwarteschlangen stammen, neu konfiguriert werden soll.

**Zugehörige Konzepte**

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

**Zugehörige Tasks**

Definition einer fernen Warteschlange hinzufügen, um Nachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet wurden

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine ferne Definition einer Clusterwarteschlange und einen separaten Senderkanal und eine separate Übertragungswarteschlange.

Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Clusternachrichten zu verwenden. Die Lösung verwendet eine zusätzliche Clusterübertragungswarteschlange, um den Nachrichtenverkehr auf einen einzelnen Warteschlangenmanager in einem Cluster zu trennen.

Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden

Ändern Sie die Konfiguration von überlappenden Clustern, die einen Gateway-Warteschlangenmanager verwenden. Nachdem die Änderungsnachrichten von dem Gateway-Warteschlangenmanager an eine Anwendung übertragen wurden, ohne dieselbe Übertragungswarteschlange oder Kanäle wie andere Cluster-

nachrichten zu verwenden. Die Lösung verwendet einen zusätzlichen Cluster, um die Nachrichten in einer bestimmten Clusterwarteschlange zu isolieren.

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen” auf Seite 342
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

## Clusterwarteschlange aus einem WS-Manager entfernen

Inaktivieren Sie die INVENTQ -Warteschlange in Toronto. Senden Sie alle Bestandsnachrichten an New York, und löschen Sie die INVENTQ -Warteschlange in Toronto, wenn sie leer ist.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY-Cluster wurde wie in „Hinzufügen eines Warteschlangenmanagers, der eine Warteschlange enthält” auf Seite 347 beschrieben eingerichtet. Sie enthält vier WS-Manager. LONDON und NEWYORK enthalten vollständige Repositorys. PARIS und TORONTO enthalten Teilrepositorys. Die Inventaranwendung wird auf den Systemen in New York und Toronto ausgeführt und wird durch das Eintreffen von Nachrichten in der INVENTQ -Warteschlange gesteuert.
- Aufgrund der geringeren Auslastung möchten Sie die Inventaranwendung in Toronto nicht mehr ausführen. Sie möchten die INVENTQ -Warteschlange, die vom Warteschlangenmanager TORONTO gehostet wird, inaktivieren und TORONTO -Feednachrichten in die INVENTQ -Warteschlange in NEWYORK stellen.
- Die Netzkonnektivität besteht zwischen allen vier Systemen.
- Das Netzprotokoll ist TCP.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um eine Clusterwarteschlange zu entfernen.

### Vorgehensweise

1. Geben Sie an, dass die Warteschlange nicht mehr verfügbar ist.

   Wenn Sie eine Warteschlange aus einem Cluster entfernen möchten, entfernen Sie den Clusternamen aus der lokalen Warteschlangendefinition. Ändern Sie die INVENTQ unter TORONTO so, dass sie für den Rest des Clusters nicht zugänglich ist:

   ```
   ALTER QLOCAL(INVENTQ) CLUSTER(' ')
   ```

2. Überprüfen Sie, ob die Warteschlange nicht mehr verfügbar ist.

   Überprüfen Sie in einem vollständigen WS-Manager-Repository entweder LONDON oder NEWYORK, ob die Warteschlange nicht mehr vom Warteschlangenmanager TORONTO per Hosting bereitgestellt wird, indem Sie den folgenden Befehl ausgeben:

   ```
   DIS QCLUSTER (INVENTQ)
   ```

   TORONTO ist in den Ergebnissen nicht aufgeführt, wenn der Befehl ALTER erfolgreich ausgeführt wurde.
3. Inaktivieren Sie die Warteschlange.

Inaktivieren Sie die INVENTQ -Warteschlange in TORONTO , so dass keine weiteren Nachrichten in diese Warteschlange geschrieben werden können:

```
ALTER QLOCAL(INVENTQ) PUT(DISABLED)
```

Nun werden Nachrichten, die mit MQOO_BIND_ON_OPEN in diese Warteschlange gesendet werden, in die Warteschlange für dead-letter (dead-letter) gesendet Sie müssen alle Anwendungen stoppen, um Nachrichten explizit in die Warteschlange dieses Warteschlangenmanagers zu stellen.

4. Überwachen Sie die Warteschlange, bis sie leer ist.

   Überwachen Sie die Warteschlange mit dem Befehl DISPLAY QUEUE , indem Sie die Attribute IP-PROCS, OPPROCSund CURDEPTHangeben, oder verwenden Sie den Befehl **WRKMQMSTS** unter IBM i. Wenn die Anzahl der Ein-und Ausgabeprozesse und die aktuelle Tiefe der Warteschlangenesare alle null sind, ist die Warteschlange leer.

5. Überwachen Sie den Kanal, um sicherzustellen, dass keine unbestäenderen Nachrichten vorhanden sind.

   Um sicherzustellen, dass keine unbestäubbaren Nachrichten auf dem Kanal INVENTORY.TORONTO vorhanden sind, überwachen Sie den Clustersenderkanal INVENTORY.TORONTO auf jedem der anderen WS-Manager. Setzen Sie den Befehl DISPLAY CHSTATUS ab, indem Sie den Parameter INDOUBT von jedem WS-Manager angeben:

   ```
   DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
   ```

   Wenn unbestätigte Nachrichten vorhanden sind, müssen Sie diese beheben, bevor Sie fortfahren. Sie können z. B. versuchen, den Befehl RESOLVE channel auszugeben oder den Kanal zu stoppen und erneut zu starten.

6. Löschen Sie die lokale Warteschlange.

   Wenn Sie sicher sind, dass keine Nachrichten mehr an die Inventaranwendung unter TORONTO zugestellt werden sollen, können Sie die Warteschlange löschen:

   ```
   DELETE QLOCAL(INVENTQ)
   ```

7. Sie können die Inventaranwendung jetzt aus dem System in Toronto entfernen.

   Durch das Entfernen der Anwendung wird die Duplizierung vermieden und Speicherplatz auf dem System eingespart.

## Ergebnisse

Der Cluster, der von dieser Task konfiguriert wird, ist wie der von der vorherigen Task eingerichtet. Der Unterschied ist, dass die INVENTQ -Warteschlange nicht mehr auf WS-Manager TORONTO verfügbar ist.

Wenn Sie die Warteschlange in Schritt 1 außer Betrieb genommen haben, hat der WS-Manager von TORONTO eine Nachricht an die beiden vollständigen WS-Manager-Repositorys gesendet. Sie hat sie über die Änderung des Status benachrichtigt. Die vollständigen WS-Manager-Repository-WS-Manager übergeben diese Informationen an andere Warteschlangenmanager im Cluster, die Aktualisierungen an den Informationen zu INVENTQ angefordert haben.

Wenn ein Warteschlangenmanager eine Nachricht in die INVENTQ -Warteschlange einreiht, zeigt das aktualisierte Teilrepository an, dass die Warteschlange von INVENTQ nur auf dem NEWYORK -Warteschlangenmanager verfügbar ist. Die Nachricht wird an den WS-Manager von NEWYORK gesendet.

## Nächste Schritte

In dieser Task war nur eine Warteschlange zum Entfernen vorhanden, und es wurde nur ein Cluster entfernt, aus dem sie entfernt werden konnte.

Angenommen, es gibt viele Warteschlangen, die sich auf eine Namensliste beziehen, die viele Clusternamen enthält. Der WS-Manager TORONTO kann beispielsweise nicht nur den INVENTQ, sondern auch die PAYROLLQ, SALESQ und PURCHASESQ enthalten. TORONTO stellt diese Warteschlangen in allen entsprechenden Clustern, INVENTORY, PAYROLL, SALES und PURCHASES zur Verfügung. Definieren Sie eine Namensliste der Clusternamen auf dem WS-Manager von TORONTO :

```
DEFINE NAMELIST(TOROLIST)
DESCR('List of clusters TORONTO is in')
NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

Fügen Sie die Namensliste zu jeder Warteschlangendefinition hinzu:

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

Nehmen Sie nun an, dass Sie alle diese Warteschlangen aus dem SALES -Cluster entfernen möchten, da die Operation SALES von der Operation PURCHASES übernommen werden soll. Sie müssen lediglich die TOROLIST -Namensliste ändern, um den Namen des SALES -Clusters aus dem Cluster zu entfernen.

Wenn Sie eine einzelne Warteschlange aus einem der Cluster in der Namensliste entfernen möchten, erstellen Sie eine Namensliste, die die verbleibende Liste der Clusternamen enthält. Ändern Sie anschließend die Warteschlangendefinition so, dass sie die neue Namensliste verwendet. Gehen Sie wie folgt vor, um den PAYROLLQ aus dem INVENTORY -Cluster

1. Erstellen Sie eine Namensliste:

   ```
   DEFINE NAMELIST(TOROSHORTLIST)
   DESCR('List of clusters TORONTO is in other than INVENTORY')
   NAMES(PAYROLL, SALES, PURCHASES)
   ```

2. Ändern Sie die PAYROLLQ -Warteschlangendefinition:

   ```
   ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
   ```

## Warteschlangenmanager aus einem Cluster entfernen: Best Practice

Entfernen Sie einen Warteschlangenmanager aus einem Cluster in Szenarios, in denen der WS-Manager normalerweise mit mindestens einem vollständigen Repository im Cluster kommunizieren kann.

### Vorbereitende Schritte

Diese Methode ist die bewährte Methode für Szenarios, in denen mindestens ein vollständiges Repository verfügbar ist und von dem Warteschlangenmanager, der entfernt wird, kontaktiert werden kann. Diese Methode beinhaltet den geringsten manuellen Eingriff und ermöglicht dem WS-Manager die Aushandlung eines kontrollierten Rückzugs aus dem Cluster. Wenn der Warteschlangenmanager, der entfernt wird, keine Verbindung zu einem vollständigen Repository aufnehmen kann, finden Sie weitere Informationen in „Entfernen eines Warteschlangenmanagers aus einem Cluster: Alternative Methode” auf Seite 392.

### Informationen zu diesem Vorgang

Diese Beispieltask entfernt den WS-Manager LONDON aus dem INVENTORY -Cluster. Der INVENTORY -Cluster wird wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben konfiguriert und wie in „Clusterwarteschlange aus einem WS-Manager entfernen” auf Seite 387 beschrieben geändert.

Der Prozess zum Entfernen eines Warteschlangenmanagers aus einem Cluster ist komplizierter als der Prozess zum Hinzufügen eines Warteschlangenmanagers.

Wenn ein Warteschlangenmanager einem Cluster beitritt, verfügen die vorhandenen Mitglieder des Clusters über keine Kenntnisse des neuen Warteschlangenmanagers und haben daher keine Interaktionen mit dem Cluster. Es müssen neue Sender-und Empfängerkanäle auf dem Verbindungswarteschlangenmanager erstellt werden, damit sie eine Verbindung zu einem vollständigen Repository herstellen können.

Wenn ein Warteschlangenmanager aus einem Cluster entfernt wird, ist es wahrscheinlich, dass Anwendungen, die mit dem Warteschlangenmanager verbunden sind, Objekte verwenden, wie z. B. Warteschlangen, die an anderer Stelle im Cluster enthalten sind. Auch Anwendungen, die mit anderen Warteschlangenmanagern im Cluster verbunden sind, verwenden möglicherweise Objekte, die auf dem Zielwarteschlangenmanager gehostet sind. Als Ergebnis dieser Anwendungen kann der aktuelle WS-Manager zusätzliche Senderkanäle erstellen, um die Kommunikation mit anderen Cluster-Membern als dem vollständigen Repository herzustellen, das für die Teilnahme am Cluster verwendet wurde. Jeder WS-Manager im Cluster verfügt über eine zwischengespeicherte Kopie von Daten, die andere Cluster-Member beschreiben. Dies kann die Entfernung einschließen, die gerade entfernt wird.

## Vorgehensweise

1. Stellen Sie vor dem Entfernen des Warteschlangenmanagers aus dem Cluster sicher, dass der Warteschlangenmanager keine Ressourcen mehr bereitstellt, die vom Cluster benötigt werden:
   - Wenn der Warteschlangenmanager ein vollständiges Repository enthält, führen Sie die Schritte 1 bis 6 von „Vollrepositorys in einen anderen WS-Manager verschieben" auf Seite 352 aus. Wenn die vollständige Repository-Funktionalität des zu entfernenden Warteschlangenmanagers nicht in einen anderen WS-Manager verschoben werden soll, ist es nur erforderlich, die Schritte 5 und 6 auszuführen.
   - Wenn der Warteschlangenmanager Clusterwarteschlangen enthält, führen Sie die Schritte 1 bis 7 von „Clusterwarteschlange aus einem WS-Manager entfernen" auf Seite 387 aus.
   - Wenn der Warteschlangenmanager Cluster-Topics enthält, löschen Sie die Themen (z. B. mit dem Befehl DELETE TOPIC) oder verschieben Sie sie auf andere Hosts, wie im Abschnitt „Clusterthemendefinition in einen anderen WS-Manager verschieben" auf Seite 474 beschrieben.

     **Anmerkung:** Wenn Sie einen Warteschlangenmanager aus einem Cluster entfernen und der WS-Manager weiterhin ein Clusterthema enthält, versucht der Warteschlangenmanager möglicherweise weiterhin, Veröffentlichungen an die Warteschlangenmanager zu übergeben, die im Cluster bleiben, bis das Thema gelöscht wird.
2. Ändern Sie die manuell definierten Clusterempfängerkanäle, um sie aus dem Cluster zu entfernen, auf WS-Manager LONDON:

   ```
   ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
   ```
3. Ändern Sie die manuell definierten Clustersenderkanäle, um sie aus dem Cluster zu entfernen, auf WS-Manager LONDON:

   ```
   ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
   ```

   Die anderen WS-Manager im Cluster lernen, dass dieser Warteschlangenmanager und seine Clusterressourcen nicht mehr Teil des Clusters sind.
4. Überwachen Sie die Clustersendewarteschlange auf dem Warteschlangenmanager LONDON, bis keine Nachrichten vorhanden sind, die darauf warten, in ein vollständiges Repository im Cluster fließen zu können.

   ```
   DISPLAY CHSTATUS(INVENTORY.PARIS) XQMSGSA
   ```

   Wenn Nachrichten in der Übertragungswarteschlange verbleiben, stellen Sie fest, warum sie nicht an die vollständigen Repositorys von PARIS und NEWYORK gesendet werden, bevor Sie fortfahren.

## Ergebnisse

Der WS-Manager LONDON ist nicht mehr Teil des Clusters. Er kann jedoch weiterhin als unabhängiger Warteschlangenmanager eingesetzt werden.

## Nächste Schritte

Das Ergebnis dieser Änderungen kann bestätigt werden, indem der folgende Befehl für die verbleibenden Member des Clusters ausgegeben wird:

```
DISPLAY CLUSQMGR(LONDON)
```

Der Warteschlangenmanager wird so lange angezeigt, bis die automatisch definierten Clustersenderkanäle gestoppt wurden. Sie können warten, bis dies geschehen ist, oder Sie können mit dem folgenden Befehl fortfahren, um aktive Instanzen zu überwachen:

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

Wenn Sie sicher sind, dass diesem WS-Manager keine Nachrichten mehr zugestellt werden, können Sie die Clustersenderkanäle auf LONDON stoppen, indem Sie den folgenden Befehl für die verbleibenden Member des Clusters ausgeben:

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

Nachdem die Änderungen im gesamten Cluster weitergegeben wurden und keine weiteren Nachrichten an diesen Warteschlangenmanager gesendet werden, stoppen Sie den CLUSRCVR -Kanal unter LONDON und löschen Sie ihn nicht mehr:

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
```

Wenn für diesen Kanal eine manuell definierte Übertragungswarteschlange verwendet wurde und das CLCHNAME-Muster keinem anderen vorhandenen oder geplanten Kanal entspricht, können Sie die Übertragungswarteschlange löschen. For example:

```
DELETE QLOCAL(PARIS.CUSTOM.XMITQ)
```

**Anmerkung:** Bei automatisch definierten Übertragungswarteschlangen oder dem gemeinsam genutzten SYSTEM.CLUSTER.TRANSMIT.QUEUE wird verwendet, dieser Schritt ist nicht erforderlich.

Der entfernte WS-Manager kann zu einem späteren Zeitpunkt wieder in den Cluster aufgenommen werden, wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben. Der entfernte Warteschlangenmanager speichert weiterhin die Kenntnisse der verbleibenden Mitglieder des Clusters für bis zu 90 Tage ein. Wenn Sie nicht warten möchten, bis dieser Cache abläuft, kann er zwangsweise entfernt werden, wie in „WS-Manager in den Status vor dem Cluster zurückschreiben” auf Seite 394 beschrieben.

**Zugehörige Tasks**

Warteschlangenmanager aus einem Cluster entfernen (mit IBM MQ Explorer)

**Zugehörige Verweise**

ALTER CHANNEL (Kanaleinstellungen ändern)

DISPLAY CHANNEL (Kanaldefinition anzeigen)

DISPLAY CHSTATUS (Kanalstatus anzeigen)

DISPLAY CLUSQMGR (Kanalinformationen für Clusterwarteschlangenmanager anzeigen)

STOP CHANNEL (Kanal stoppen)

### *Entfernen eines Warteschlangenmanagers aus einem Cluster: Alternative Methode*

Entfernen Sie einen Warteschlangenmanager aus einem Cluster in Szenarios, in denen der WS-Manager aufgrund eines signifikanten System-oder Konfigurationsproblems nicht mit einem vollständigen Repository im Cluster kommunizieren kann.

## Vorbereitende Schritte

Diese alternative Methode zum Entfernen eines Warteschlangenmanagers aus einem Cluster wird manuell gestoppt und löscht alle Clusterkanäle, die den entfernten WS-Manager mit dem Cluster verbinden, und entfernt den Warteschlangenmanager zwangsweise aus dem Cluster. Diese Methode wird in Szenarios verwendet, in denen der Warteschlangenmanager, der entfernt wird, nicht mit einem der vollständigen Repositorys kommunizieren kann. Dies kann (z. B.) sein, weil der Warteschlangenmanager nicht mehr funktioniert hat oder weil zwischen dem WS-Manager und dem Cluster ein verlängerter Kommunikationsfehler aufgetreten ist. Verwenden Sie andernfalls die gängste Methode: „Warteschlangenmanager aus einem Cluster entfernen: Best Practice” auf Seite 389.

## Informationen zu diesem Vorgang

Diese Beispieltask entfernt den WS-Manager `LONDON` aus dem `INVENTORY` -Cluster. Der `INVENTORY` -Cluster wird wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben konfiguriert und wie in „Clusterwarteschlange aus einem WS-Manager entfernen” auf Seite 387 beschrieben geändert.

Der Prozess zum Entfernen eines Warteschlangenmanagers aus einem Cluster ist komplizierter als der Prozess zum Hinzufügen eines Warteschlangenmanagers.

Wenn ein Warteschlangenmanager einem Cluster beitritt, verfügen die vorhandenen Mitglieder des Clusters über keine Kenntnisse des neuen Warteschlangenmanagers und haben daher keine Interaktionen mit dem Cluster. Es müssen neue Sender-und Empfängerkanäle auf dem Verbindungswarteschlangenmanager erstellt werden, damit sie eine Verbindung zu einem vollständigen Repository herstellen können.

Wenn ein Warteschlangenmanager aus einem Cluster entfernt wird, ist es wahrscheinlich, dass Anwendungen, die mit dem Warteschlangenmanager verbunden sind, Objekte verwenden, wie z. B. Warteschlangen, die an anderer Stelle im Cluster enthalten sind. Auch Anwendungen, die mit anderen Warteschlangenmanagern im Cluster verbunden sind, verwenden möglicherweise Objekte, die auf dem Zielwarteschlangenmanager gehostet sind. Als Ergebnis dieser Anwendungen kann der aktuelle WS-Manager zusätzliche Senderkanäle erstellen, um die Kommunikation mit anderen Cluster-Membern als dem vollständigen Repository herzustellen, das für die Teilnahme am Cluster verwendet wurde. Jeder WS-Manager im Cluster verfügt über eine zwischengespeicherte Kopie von Daten, die andere Cluster-Member beschreiben. Dies kann die Entfernung einschließen, die gerade entfernt wird.

Diese Prozedur kann in einem Notfall angemessen sein, wenn es nicht möglich ist, auf den WS-Manager zu warten, um den Cluster ordnungsgemäß zu verlassen.

## Vorgehensweise

1. Stellen Sie vor dem Entfernen des Warteschlangenmanagers aus dem Cluster sicher, dass der Warteschlangenmanager keine Ressourcen mehr bereitstellt, die vom Cluster benötigt werden:
   - Wenn der Warteschlangenmanager ein vollständiges Repository enthält, führen Sie die Schritte 1 bis 6 von „Vollrepositorys in einen anderen WS-Manager verschieben” auf Seite 352 aus. Wenn die vollständige Repository-Funktionalität des zu entfernenden Warteschlangenmanagers nicht in einen anderen WS-Manager verschoben werden soll, ist es nur erforderlich, die Schritte 5 und 6 auszuführen.
   - Wenn der Warteschlangenmanager Clusterwarteschlangen enthält, führen Sie die Schritte 1 bis 7 von „Clusterwarteschlange aus einem WS-Manager entfernen” auf Seite 387 aus.

- Wenn der Warteschlangenmanager Cluster-Topics enthält, löschen Sie die Themen (z. B. mit dem Befehl DELETE TOPIC) oder verschieben Sie sie auf andere Hosts, wie im Abschnitt „Clusterthemendefinition in einen anderen WS-Manager verschieben" auf Seite 474 beschrieben.

  **Anmerkung:** Wenn Sie einen Warteschlangenmanager aus einem Cluster entfernen und der WS-Manager weiterhin ein Clusterthema enthält, versucht der Warteschlangenmanager möglicherweise weiterhin, Veröffentlichungen an die Warteschlangenmanager zu übergeben, die im Cluster bleiben, bis das Thema gelöscht wird.

2. Stoppen Sie alle Kanäle, die für die Kommunikation mit anderen Warteschlangenmanagern im Cluster verwendet werden. Verwenden Sie MODE(FORCE) , um den CLUSRCVR -Kanal auf WS-Manager LONDON zu stoppen. Andernfalls müssen Sie möglicherweise auf den Kanal des Sende-WS-Managers warten, um den Kanal zu stoppen:

```
STOP CHANNEL(INVENTORY.LONDON) MODE(FORCE)
STOP CHANNEL(INVENTORY.TORONTO)
STOP CHANNEL(INVENTORY.PARIS)
STOP CHANNEL(INVENTORY.NEWYORK)
```

3. Überwachen Sie den Kanalstatus auf dem Warteschlangenmanager LONDON, bis die Kanäle gestoppt werden:

```
DISPLAY CHSTATUS(INVENTORY.LONDON)
DISPLAY CHSTATUS(INVENTORY.TORONTO)
DISPLAY CHSTATUS(INVENTORY.PARIS)
DISPLAY CHSTATUS(INVENTORY.NEWYORK)
```

   Nach dem Stoppen der Kanäle werden keine weiteren Anwendungsnachrichten an oder von den anderen WS-Managern in dem Cluster gesendet.

4. Löschen Sie die manuell definierten Clusterkanäle auf dem WS-Manager LONDON:

```
DELETE CHANNEL(INVENTORY.NEWYORK)
DELETE CHANNEL(INVENTORY.TORONTO)
```

5. Die verbleibenden WS-Manager im Cluster behalten weiterhin die Kenntnisse des entfernten Warteschlangenmanagers und können weiterhin Nachrichten an diese Warteschlangenmanager senden. Um die Kenntnisse aus den verbleibenden WS-Managern zu löschen, setzen Sie den entfernten WS-Manager aus dem Cluster in einem der vollständigen Repositorys zurück:

```
RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
```

   Wenn ein weiterer WS-Manager im Cluster vorhanden sein könnte, der denselben Namen wie der entfernte Warteschlangenmanager hat, geben Sie den **QMID** des entfernten Warteschlangenmanagers an.

## Ergebnisse

Der WS-Manager LONDON ist nicht mehr Teil des Clusters. Er kann jedoch weiterhin als unabhängiger Warteschlangenmanager eingesetzt werden.

## Nächste Schritte

Das Ergebnis dieser Änderungen kann bestätigt werden, indem der folgende Befehl für die verbleibenden Member des Clusters ausgegeben wird:

```
DISPLAY CLUSQMGR(LONDON)
```

Der Warteschlangenmanager wird so lange angezeigt, bis die automatisch definierten Clustersenderkanäle gestoppt wurden. Sie können warten, bis dies geschehen ist, oder Sie können mit dem folgenden Befehl fortfahren, um aktive Instanzen zu überwachen:

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

Nachdem die Änderungen im gesamten Cluster weitergegeben wurden und keine weiteren Nachrichten an diesen Warteschlangenmanager gesendet werden, löschen Sie den CLUSRCVR -Kanal unter LONDON:

```
DELETE CHANNEL(INVENTORY.LONDON)
```

Der entfernte WS-Manager kann zu einem späteren Zeitpunkt wieder in den Cluster aufgenommen werden, wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben. Der entfernte Warteschlangenmanager speichert weiterhin die Kenntnisse der verbleibenden Mitglieder des Clusters für bis zu 90 Tage ein. Wenn Sie nicht warten möchten, bis dieser Cache abläuft, kann er zwangsweise entfernt werden, wie in „WS-Manager in den Status vor dem Cluster zurückschreiben” auf Seite 394 beschrieben.

**Zugehörige Verweise**
DELETE CHANNEL (Löschen eines Kanals)
DISPLAY CHANNEL (Kanaldefinition anzeigen)
DISPLAY CHSTATUS (Kanalstatus anzeigen)
DISPLAY CLUSQMGR (Kanalinformationen für Clusterwarteschlangenmanager anzeigen)
STOP CHANNEL (Kanal stoppen)
RESET CLUSTER (Cluster zurücksetzen)

## WS-Manager in den Status vor dem Cluster zurückschreiben

Wenn ein WS-Manager aus einem Cluster entfernt wird, behält er die Kenntnisse der verbleibenden Cluster-Member bei. Dieses Wissen verfällt schließlich und wird automatisch gelöscht. Wenn Sie es jedoch sofort löschen möchten, können Sie die Schritte in diesem Thema verwenden.

### Vorbereitende Schritte

Es wird davon ausgegangen, dass der Warteschlangenmanager aus dem Cluster entfernt wurde und keine Arbeit mehr im Cluster mehr ausführt. Beispielsweise empfangen die Warteschlangen keine Nachrichten mehr aus dem Cluster, und es warten keine Anwendungen auf Nachrichten, die in diese Warteschlangen eintreffen.

### Informationen zu diesem Vorgang

Wenn ein Warteschlangenmanager aus einem Cluster entfernt wird, behält er die Kenntnisse der verbleibenden Cluster-Member für bis zu 90 Tage bei. Dies kann Systemvorteile haben, insbesondere, wenn der WS-Manager schnell wieder in den Cluster aufgenommen wird. Wenn dieses Wissen schließlich abläuft, wird es automatisch gelöscht. Es gibt jedoch Gründe, warum Sie diese Informationen lieber manuell löschen möchten. For example:

- Möglicherweise möchten Sie bestätigen, dass Sie jede Anwendung in diesem WS-Manager gestoppt haben, die zuvor die Clusterressourcen verwendet hat. Solange die Kenntnisse der verbleibenden Cluster-Member nicht mehr vorhanden sind, schreibt jede solche Anwendung weiterhin in eine Übertragungswarteschlange. Nachdem die Clusterkenntnisse gelöscht wurden, generiert das System eine Fehlernachricht, wenn eine solche Anwendung versucht, Clusterressourcen zu verwenden.
- Wenn Sie Statusinformationen für den Warteschlangenmanager anzeigen, können Sie es vorziehen, Informationen über verbleibende Cluster-Member nicht zu verlaufen.

In dieser Task wird der INVENTORY -Cluster als Beispiel verwendet. Der LONDON -Warteschlangenmanager wurde wie in „Warteschlangenmanager aus einem Cluster entfernen: Best Practice” auf Seite 389

beschrieben aus dem INVENTORY -Cluster entfernt. Wenn Sie die Kenntnisse der verbleibenden Member des Clusters löschen möchten, geben Sie die folgenden Befehle im LONDON -Warteschlangenmanager aus.

### Vorgehensweise

1. Entfernen Sie alle Speicher der anderen WS-Manager im Cluster aus diesem Warteschlangenmanager:

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. Überwachen Sie den Warteschlangenmanager, bis alle Clusterressourcen nicht mehr verfügbar sind:

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

**Zugehörige Konzepte**

Cluster

Clusterkomponenten

**Zugehörige Verweise**

Vergleich von Clustering und verteilter Steuerung von Warteschlangen

## Verwalten eines Warteschlangenmanagers

Setzen Sie einen WS-Manager aus einem Cluster aus und setzen Sie sie wieder fort, um die Wartung durchzuführen.

### Informationen zu diesem Vorgang

Von Zeit zu Zeit müssen Sie möglicherweise eine Wartung für einen Warteschlangenmanager ausführen, der Teil eines Clusters ist. Sie müssen beispielsweise möglicherweise Sicherungen der Daten in den zugehörigen Warteschlangen erstellen oder Fixes auf die Software anwenden. Wenn der Warteschlangenmanager alle Warteschlangen enthält, müssen seine Aktivitäten ausgesetzt werden. Wenn die Wartung abgeschlossen ist, können die Aktivitäten wieder aufgenommen werden.

### Vorgehensweise

1. Setzen Sie einen Warteschlangenmanager aus, indem Sie den Befehl SUSPEND QMGR **runmqsc** ausgeben:

```
SUSPEND QMGR CLUSTER(SALES)
```

   Der Befehl SUSPEND **runmqsc** benachrichtigt die Warteschlangenmanager im SALES -Cluster darüber, dass dieser Warteschlangenmanager ausgesetzt wurde.

   Der Befehl SUSPEND QMGR dient nur dazu, andere Warteschlangenmanager zu informieren, um zu vermeiden, dass Nachrichten an diesen WS-Manager gesendet werden, wenn möglich. Dies bedeutet nicht, dass der WS-Manager inaktiviert ist. Einige Nachrichten, die von diesem WS-Manager bearbeitet werden müssen, werden immer noch an sie gesendet, z. B. wenn dieser Warteschlangenmanager der einzige Host einer Clusterwarteschlange ist.

   Solange der WS-Manager ausgesetzt ist, vermeiden die Routinen der Workloadverwaltung, Nachrichten an diese zu senden. Nachrichten, die von diesem WS-Manager bearbeitet werden müssen, enthalten Nachrichten, die vom lokalen Warteschlangenmanager gesendet werden.

   IBM MQ verwendet einen Algorithmus für den Lastausgleich, um festzustellen, welche Ziele geeignet sind, statt den lokalen WS-Manager auszuwählen, wann immer dies möglich ist.

a) Verwenden Sie die Option FORCE im SUSPEND QMGR -Befehl, um die Aussetzung eines Warteschlangenmanagers zu erzwingen:

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

MODE(FORCE) stoppt zwangsweise alle eingehenden Kanäle von anderen WS-Managern im Cluster. Wenn Sie MODE(FORCE) nicht angeben, gilt der Standardwert MODE(QUIESCE) .

2. Führen Sie die erforderlichen Wartungsaufgaben aus.
3. Setzen Sie den Befehl RESUME QMGR **runmqsc** ab, um den Warteschlangenmanager wiederaufzunehmen:

```
RESUME QMGR CLUSTER(SALES)
```

### Ergebnisse

Der Befehl RESUME **runmqsc** benachrichtigt die vollständigen Repositorys darüber, dass der Warteschlangenmanager wieder verfügbar ist. Die vollständigen WS-Manager-Repositorys verbreiten diese Informationen an andere Warteschlangenmanager, die Aktualisierungen an Informationen zu diesem Warteschlangenmanager angefordert haben.

## Verwalten der Clusterübertragungswarteschlange

Stellen Sie sicher, dass die Clusterübertragungswarteschlangen zur Verfügung stehen. Sie sind von wesentlicher Bedeutung für die Leistung von Clustern. z/OS Setzen Sie unter z/OS den Wert für INDXTYPE einer Clusterübertragungswarteschlange auf CORRELID.

### Vorbereitende Schritte

- Stellen Sie sicher, dass die Clusterübertragungswarteschlange nicht voll ist.
- Achten Sie darauf, dass Sie keinen ALTER **runmqsc** -Befehl absetzen, um ihn entweder zu inaktivieren oder versehentlich zu inaktivieren.
- Stellen Sie sicher, dass das Medium, in dem die Clusterübertragungswarteschlange auf z/OS gespeichert ist (z. B. z/OS-Seitengruppen), nicht voll wird.

### Informationen zu diesem Vorgang

z/OS

Die folgende Prozedur gilt nur für z/OS.

### Vorgehensweise

Setzen Sie INDXTYPE auf die Clusterübertragungswarteschlange auf CORRELID .

## Cluster-WS-Manager wird neu freigegeben

Mit dem Befehl REFRESH CLUSTER können Sie automatisch definierte Kanäle und automatisch definierte Clusterobjekte aus dem lokalen Repository entfernen. Es gehen keine Nachrichten verloren.

### Vorbereitende Schritte

Möglicherweise werden Sie von Ihrem IBM Support Center aufgefordert, den Befehl zu verwenden. Verwenden Sie den Befehl nicht ohne sorgfältige Prüfung. Bei großen Clustern kann die Verwendung des Befehls **REFRESH CLUSTER** beispielsweise zu einer Unterbrechung des Clusters führen, während er in Bearbeitung ist, und danach in 27-Tage-Intervallen, wenn die Clusterobjekte Statusaktualisierungen

automatisch an alle interessierten Warteschlangenmanager senden. Siehe Clustering: Best Practices für REFRESH CLUSTER verwenden .

### Informationen zu diesem Vorgang

Ein Warteschlangenmanager kann einen neuen Start in einem Cluster vornehmen. Unter normalen Umständen ist es nicht erforderlich, den Befehl REFRESH CLUSTER zu verwenden.

### Vorgehensweise

Geben Sie den Befehl REFRESH CLUSTER **MQSC** in einem Warteschlangenmanager aus, um automatisch definierte Clusterwarteschlangenmanager und Warteschlangenobjekte aus dem lokalen Repository zu entfernen.

Mit dem Befehl werden nur Objekte entfernt, die sich auf andere Warteschlangenmanager beziehen. Es werden keine Objekte entfernt, die sich auf den lokalen Warteschlangenmanager beziehen. Mit dem Befehl werden auch automatisch definierte Kanäle entfernt. Dadurch werden Kanäle entfernt, die keine Nachrichten in der Clusterübertragungswarteschlange enthalten und nicht an einen vollständigen WS-Manager-Repository angeschlossen sind.

### Ergebnisse

Mit dem Befehl REFRESH CLUSTER kann ein WS-Manager in Bezug auf seinen vollständigen Repository-Inhalt wirksam kalt gestartet werden. IBM MQ stellt sicher, dass keine Daten aus den Warteschlangen verloren gehen.

**Zugehörige Informationen**

Clustering: Best Practices für REFRESH CLUSTER verwenden

## Cluster-WS-Manager wiederherstellen

Verwenden Sie den Befehl REFRESH CLUSTER **runmqsc** , um die Clusterinformationen zu einem Warteschlangenmanager auf den neuesten Stand zu bringen. Führen Sie diese Prozedur aus, nachdem Sie einen Warteschlangenmanager aus einer Zeitpunktsicherung wiederhergestellt haben.

### Vorbereitende Schritte

Sie haben einen Clusterwarteschlangenmanager aus einer zeitpunktbasierten Sicherung wiederhergestellt.

### Informationen zu diesem Vorgang

Um einen Warteschlangenmanager in einem Cluster wiederherzustellen, stellen Sie den Warteschlangenmanager wieder her und aktualisieren Sie anschließend die Clusterinformationen mit dem Befehl REFRESH CLUSTER **runmqsc** .

**Anmerkung:** Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager senden, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

### Vorgehensweise

Geben Sie den Befehl REFRESH CLUSTER auf dem wiederhergestellten Warteschlangenmanager für alle Cluster aus, an denen der Warteschlangenmanager beteiligt ist.

### Nächste Schritte

Es ist nicht erforderlich, den Befehl REFRESH CLUSTER auf einem anderen WS-Manager auszugeben.

**Zugehörige Konzepte**

Clustering: Best Practices für REFRESH CLUSTER verwenden

## Clusterkanäle für Verfügbarkeit konfigurieren

Befolgen Sie die bewährten Konfigurationsverfahren, um Clusterkanäle störungsfrei zu halten, wenn unterbrochenen Netzstopps vorhanden sind.

### Vorbereitende Schritte

Cluster entlasten Sie von der Notwendigkeit, Kanäle zu definieren, aber Sie müssen sie trotzdem verwalten. Dieselbe Kanaltechnologie wird für die Kommunikation zwischen Warteschlangenmanagern in einem Cluster verwendet, wie es in der verteilten Steuerung von Warteschlangen verwendet wird. Um Informationen über Clusterkanäle zu erhalten, müssen Sie mit den folgenden Themen vertraut sein:

- Funktionsweise von Kanälen
- Wie Sie ihren Status finden
- Kanalexits verwenden

### Informationen zu diesem Vorgang

Es kann sein, dass Sie die folgenden Punkte besonders berücksichtigen:

### Vorgehensweise

Beachten Sie bei der Konfiguration von Clusterkanälen die folgenden Punkte:

- Wählen Sie Werte für `HBINT` oder `KAINT` auf Clustersenderkanälen und Clusterempfängerkanälen aus, die das Netz nicht mit vielen Überwachungssignalen belasten oder die Nachrichtenflüsse beibehalten. Ein Intervall von weniger als 10 Sekunden führt zu falschen Fehlern, wenn sich Ihr Netz manchmal verlangsamt und Verzögerungen in dieser Länge einführt.
- Legen Sie den Wert für `BATCHHB` fest, um das Fenster zu verkleinern, um eine Nachricht mit einem Fehler zu verursachen, da dies in einem fehlgeschlagenen Kanal unbestätigt ist. Ein unbestätigter Stapel auf einem fehlgeschlagenen Kanal ist wahrscheinlicher, wenn die Stapelverarbeitung länger zur Füllung angegeben wird. Wenn der Nachrichtenverkehr entlang des Kanals sporadisch mit langen Zeiträumen zwischen den Bursts von Nachrichten ist, ist eine fehlgeschlagene Stapelverarbeitung wahrscheinlicher.
- Es tritt ein Problem auf, wenn die Clustersenderseite eines Kanals fehlschlägt und dann versucht wird, einen Neustart zu starten, bevor das Überwachungssignal oder die "keep alive" den Fehler erkannt hat. Der Kanalsenderneustart wird zurückgewiesen, wenn das Ende des Clusterempfängers des Kanals aktiv geblieben ist. Um den Fehler zu vermeiden, müssen Sie den Clusterempfängerkanal beenden und erneut starten, wenn ein Clustersenderkanal versucht, einen Neustart durchzuführen.

  z/OS **unter IBM MQ for z/OS**

  Beheben Sie das Problem mit dem aktiv bleibenden Clusterempfängerende des Kanals mit Hilfe der Parameter **ADOPTMCA** und **ADOPTCHK** in **ALTER QMGR**.

  Multi **unter Multiplatforms**

  Beheben Sie das Problem mit dem aktiv bleibenden Clusterempfängerende des Kanals mit Hilfe der Attribute **AdoptNewMCA**, **AdoptNewMCATimeout** und **AdoptNewMCACheck** in der Datei `qm.ini` oder in der Windows-Registry.

**Beispiel**

Im Abschnitt „Empfohlene Einstellungen” auf Seite 249 finden Sie Beispiele zur Implementierung dieser Einstellungen unter IBM MQ for z/OS und IBM MQ for Multiplatforms.

## Prüfen, ob asynchrone Befehle für verteilte Netze abgeschlossen sind

Viele Befehle sind asynchron, wenn sie in einem verteilten Netz verwendet werden. Abhängig von dem Befehl und dem Netzstatus, wenn er ausgegeben wird, kann es sehr viel Zeit in Anspruch nehmen, die Verarbeitung abzuschließen. Der WS-Manager gibt keine Nachricht nach Abschluss aus, so dass Sie andere Möglichkeiten zur Überprüfung der Ausführung des Befehls benötigen.

### Informationen zu diesem Vorgang

Fast jede Konfigurationsänderung, die Sie an einem Cluster vornehmen, ist wahrscheinlich asynchron abgeschlossen. Dies liegt an der internen Verwaltung und Aktualisierung von Zyklen, die in Clustern betrieben werden. Bei Publish/Subscribe-Hierarchien ist jede Konfigurationsänderung, die sich auf Subskriptionen auswirkt, wahrscheinlich asynchron abgeschlossen. Dies ist bei dem Namen des Befehls nicht immer klar.

Die folgenden MQSC-Befehle werden möglicherweise asynchron ausgeführt. Jeder dieser Befehle verfügt über ein PCF-Äquivalent, und die meisten sind auch über den IBM MQ Explorer verfügbar. Wenn diese Befehle in einem kleinen Netz ohne Workload ausgeführt werden, werden diese Befehle in der Regel innerhalb weniger Sekunden abgeschlossen. Dies ist jedoch bei größeren und stärkeren Netzen nicht der Fall. Außerdem kann der Befehl **REFRESH CLUSTER** viel länger dauern, insbesondere wenn er auf mehreren Warteschlangenmanagern gleichzeitig ausgegeben wird.

Überprüfen Sie, ob die erwarteten Objekte auf den fernen Warteschlangenmanagern vorhanden sind, um das Vertrauen zu haben, dass diese Befehle beendet sind.

### Prozedur

- ALTER QMGR

  Verwenden Sie für den Befehl ALTER QMGR PARENT DISPLAY PUBSUB TYPE(PARENT) ALL , um den Status der angeforderten übergeordneten Beziehung zu verfolgen.

  Verwenden Sie für die Befehle ALTER QMGR REPOS und ALTER QMGR REPOSNL die Option DISPLAY CLUSQMGR QMTYPE , um die Fertigstellung zu bestätigen.
- DEFINE CHANNEL , ALTER CHANNEL und DELETE CHANNEL

  Verwenden Sie für alle Parameter, die in der Tabelle ALTER CHANNEL parameters aufgeführt sind, den Befehl DISPLAY CLUSQMGR , um zu überwachen, wann Änderungen an den Cluster weitergegeben wurden.
- DEFINE NAMELIST , ALTER NAMELIST und DELETE NAMELIST .

  Wenn Sie ein **NAMELIST** für das Attribut **CLUSNL** eines **QMgr** -Objekts verwenden, kann sich eine Warteschlange oder ein Clusterkanal auf dieses Objekt auswirken. Überwachen Sie das betroffene Objekt nach Bedarf.

  Änderungen an SYSTEM.QPUBSUB.QUEUE.NAMELIST können die Erstellung oder den Abbruch von Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie beeinflussen. Verwenden Sie den Befehl DISPLAY SUB SUBTYPE(PROXY) , um dies zu überwachen.
- DEFINE Queues , ALTER queues und DELETE queues .

  Verwenden Sie für alle Parameter, die in der Tabelle Parameter, die vom Befehl DISPLAY QUEUE zurückgegeben werden können , den Befehl DISPLAY QCLUSTER , um zu überwachen, wann Änderungen an den Cluster weitergegeben wurden.
- DEFINE SUB und DELETE SUB

  Wenn Sie die erste Subskription für eine Themenzeichenfolge definieren, können Sie Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster erstellen. Wenn Sie die letzte Subskription für eine Themenzeichenfolge löschen, können Sie außerdem Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster abbrechen.

  Um zu überprüfen, ob ein Befehl zum Definieren oder Löschen einer Subskription abgeschlossen ist, überprüfen Sie, ob die erwartete Proxy-Subskription auf anderen Warteschlangenmanagern im

verteilten Netz vorhanden ist oder nicht. Wenn Sie *direktes Routing* in einem Cluster verwenden, überprüfen Sie, ob die erwartete Proxy-Subskription in den anderen Teilrepositorys im Cluster vorhanden ist. Wenn Sie *Topic-Host-Routing* in einem Cluster verwenden, überprüfen Sie, ob die erwartete Proxy-Subskription auf den übereinstimmenden Themenhosts vorhanden ist. Verwenden Sie den folgenden MQSC-Befehl:

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

Verwenden Sie dieselbe Prüfung für die folgenden äquivalenten Subskriptions-und Abmelde- MQI-Aufrufe, wenn sie in einem Cluster oder einer Hierarchie ausgegeben werden:

– Subskribieren Sie über MQSUB .
– Beenden Sie die Subskription mithilfe von MQCLOSE mit MQCO_REMOVE_SUB.

- DEFINE TOPIC , ALTER TOPIC und DELETE TOPIC

  Wenn Sie überprüfen möchten, ob ein Befehl zum Definieren, Ändern oder Löschen eines Clusterthemas abgeschlossen ist, zeigen Sie das Thema in den anderen Teilrepositorys im Cluster an (wenn Sie *direktes Routing* verwenden) oder auf den anderen Themenhosts (wenn Sie *Topic-Host-Routing* verwenden).

  Verwenden Sie für alle Parameter, die in der Tabelle Parameter, die vom Befehl DISPLAY TOPIC zurückgegeben werden können , den Befehl `DISPLAY TCLUSTER` , um zu überwachen, wann Änderungen an den Cluster weitergegeben wurden.

  **Anmerkung:**

  – Der Parameter **CLUSTER** kann die Erstellung oder den Abbruch von Proxy-Subskriptionen in einem Publish/Subscribe-Cluster beeinflussen.
  – Die Parameter **PROXYSUB** und **SUBSCOPE** können die Erstellung oder den Abbruch von Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster beeinflussen.
  – Verwenden Sie den Befehl `DISPLAY SUB SUBTYPE(PROXYSUB)` , um dies zu überwachen.

- REFRESH CLUSTER

  Wenn Sie den Befehl **REFRESH CLUSTER** ausführen, fragen Sie die Warteschlangenlänge des Clusterbefehls ab. Warten Sie, bis der Wert null erreicht ist, und bleiben Sie bei null, bevor Sie nach den Objekten suchen.

  1. Verwenden Sie den folgenden MQSC-Befehl, um zu überprüfen, ob die Länge der Clusterbefehlswarteschlange null ist.

     ```
     DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
     ```

  2. Wiederholen Sie die Überprüfung, bis die Warteschlangenlänge null erreicht, und bleibt bei der nachfolgenden Überprüfung auf Null.

  Der Befehl **REFRESH CLUSTER** entfernt Objekte und erstellt sie erneut. In großen Konfigurationen kann die Ausführung sehr lange dauern. Siehe Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster.

- REFRESH QMGR TYPE (PROXYSUB)

  Um zu überprüfen, ob der Befehl **REFRESH QMGR TYPE(PROXYSUB)** beendet wurde, überprüfen Sie, ob die Proxy-Subskriptionen auf anderen Warteschlangenmanagern im verteilten Netz korrigiert wurden. Wenn Sie *direktes Routing* in einem Cluster verwenden, überprüfen Sie, ob die Proxy-Subskriptionen in den anderen Teilrepositorys im Cluster korrigiert wurden. Wenn Sie *Topic-Host-Routing* in einem Cluster verwenden, überprüfen Sie, ob die erwarteten Proxy-Subskriptionen auf den übereinstimmenden Themenhosts korrigiert wurden. Verwenden Sie den folgenden MQSC-Befehl:

  ```
  DISPLAY SUB(*) SUBTYPE(PROXYSUB)
  ```

- Cluster zurücksetzen

  Verwenden Sie `DISPLAY CLUSQMGR`, um zu überprüfen, ob der Befehl **RESET CLUSTER** ausgeführt wurde.
- RESET QMGR TYPE (PUBSUB)

  Verwenden Sie `DISPLAY PUBSUB TYPE(PARENT|CHILD)`, um zu überprüfen, ob der Befehl **RESET QMGR** ausgeführt wurde.

  **Anmerkung:** Der Befehl **RESET QMGR** kann zum Abbruch von Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster führen. Verwenden Sie den Befehl `DISPLAY SUB SUBTYPE(PROXYSUB)` , um dies zu überwachen.
- Sie können auch andere Systemwarteschlangen überwachen, die, wenn Befehle abgeschlossen sind, dazu neigen, eine Warteschlangenlänge von null zu verwenden.

  Sie können z. B. die `SYSTEM.INTER.QMGR.CONTROL` -Warteschlange und die `SYSTEM.INTER.QMGR.FANREQ` -Warteschlange überwachen. Weitere Informationen finden Sie unter Proxy-Subskriptionsdatenverkehr in Clustern überwachen und Erzeuger und Konsumenten in Publish/Subscribe-Netzen abgleichen .

## Nächste Schritte

Wenn diese Prüfungen nicht bestätigen, dass ein asynchroner Befehl beendet wurde, ist möglicherweise ein Fehler aufgetreten. Um zu untersuchen, überprüfen Sie zuerst das Protokoll für den Warteschlangenmanager, auf dem der Befehl abgesetzt wurde, und überprüfen Sie dann (für einen Cluster) die vollständigen Repository-Protokolle des Clusters.

**Zugehörige Verweise**

z/OS Asynchrones Verhalten von Clusterbefehlen unter z/OS

## Nachrichten an und von Clustern weiterleiten

Verwenden Sie Warteschlangenaliasnamen, WS-Manager-Aliasnamen und Definitionen ferner Warteschlangen, um Cluster mit externen Warteschlangenmanagern und anderen Clustern zu verbinden.

Weitere Informationen zum Weiterleiten von Nachrichten an und von Clustern finden Sie in den folgenden Unterabschnitten:

**Zugehörige Konzepte**

Cluster

Komponenten eines Clusters

„WS-Manager-Aliasnamen und -Cluster” auf Seite 415
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

„Warteschlangenaliasnamen und -cluster” auf Seite 419
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

„Aliasnamen für Antwortwarteschlangen und Cluster” auf Seite 419
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

**Zugehörige Tasks**

„WS-Manager-Cluster konfigurieren” auf Seite 312
Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Sie können Cluster-Komponenten definieren und Cluster erstellen und verwalten.

„Neuen Cluster einrichten” auf Seite 327

Führen Sie die folgenden Anweisungen aus, um den Beispielcluster zu konfigurieren. In separaten Anweisungen wird beschrieben, wie der Cluster auf TCP/IP, LU 6.2 und mit einer einzelnen Übertragungswarteschlange oder mehreren Übertragungswarteschlangen eingerichtet wird. Testen Sie den Cluster, indem Sie eine Nachricht von einem WS-Manager an den anderen Warteschlangenmanager senden.

**Zugehörige Verweise**

Vergleich von Clustering und verteilter Steuerung von Warteschlangen

### *Anforderung/Antwort in einem Cluster konfigurieren*

Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

## Vorbereitende Schritte

Abbildung 53 auf Seite 403 zeigt einen WS-Manager mit dem Namen QM3 , der sich außerhalb des Clusters DEMO befindet. Bei QM3 könnte es sich um einen Warteschlangenmanager auf einem IBM MQ-Produkt handeln, das keine Cluster unterstützt. QM3 enthält eine Warteschlange mit dem Namen Q3, die wie folgt definiert ist:

```
DEFINE QLOCAL(Q3)
```

Im Cluster befinden sich zwei WS-Manager, die QM1 und QM2 genannt werden. QM2 enthält eine Clusterwarteschlange mit dem Namen Q2, die wie folgt definiert ist:

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

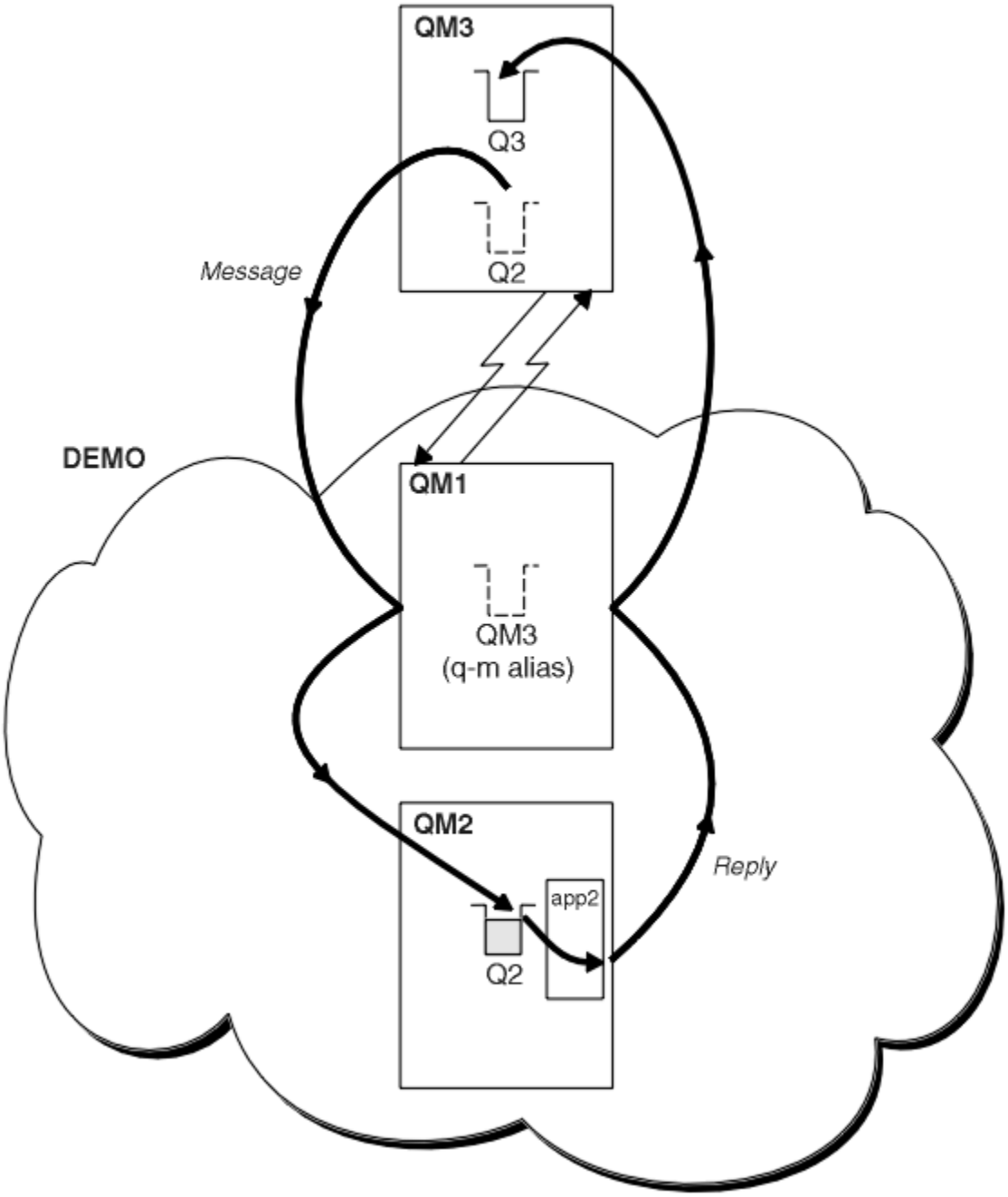



*Abbildung 53. Aus einem WS-Manager außerhalb des Clusters einschalten*

## Informationen zu diesem Vorgang

Befolgen Sie die Anweisungen in der Prozedur, um den Pfad für die Anforderungs-und Antwortnachrichten zu konfigurieren.

## Vorgehensweise

1. Senden Sie die Anforderungsnachricht an den Cluster.

   Berücksichtigen Sie, wie der Warteschlangenmanager, der sich außerhalb des Clusters befindet, eine Nachricht in die Warteschlange Q2 von QM2 eingibt, die sich innerhalb des Clusters befindet. Ein WS-Manager außerhalb des Clusters muss eine QREMOTE -Definition für jede Warteschlange in dem Cluster haben, in den Nachrichten einreiht.

   a) Definieren Sie eine ferne Warteschlange für Q2 auf QM3.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
```

Da QM3 nicht Teil eines Clusters ist, muss die Kommunikation mit verteilten Warteschlangenverfahren kommunizieren. Daher muss es auch über einen Senderkanal und eine Übertragungswarteschlange zu QM1 verfügen. QM1 benötigt einen entsprechenden Empfängerkanal. Die Kanäle und Übertragungswarteschlangen werden in Abbildung 53 auf Seite 403 nicht explizit angezeigt.

Im Beispiel gibt eine Anwendung in QM3 einen Aufruf MQPUT aus, um eine Nachricht in Q2einzureihen. Die Definition QREMOTE  bewirkt, dass die Nachricht an Q2 unter QM2 weitergeleitet wird, wobei der Senderkanal verwendet wird, der Nachrichten aus der QM1 -Übertragungswarteschlange erhält.

2. Empfangen Sie die Antwortnachricht aus dem Cluster.

   Verwenden Sie einen WS-Manager-Aliasnamen, um einen Rückgabepfad für Antworten auf einen Warteschlangenmanager außerhalb des Clusters zu erstellen. Das Gateway ( QM1) wirbt einen WS-Manager-Aliasnamen für den Warteschlangenmanager, der sich außerhalb des Clusters befindet, QM3. Er wirbt QM3 für die Warteschlangenmanager im Cluster, indem er das Clusterattribut einer WS-Manager-Aliasdefinition für QM3 hinzufügt. Eine WS-Manager-Aliasdefinition ist wie eine ferne Warteschlangendefinition, aber mit einem leeren RNAME.

   a) Definieren Sie einen WS-Manager-Aliasnamen für QM3 auf QM1.

   ```
   DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
   ```

   Wir müssen die Auswahl des Namens für die Übertragungswarteschlange berücksichtigen, die verwendet wird, um Antworten von QM1 an QM3 weiterzuleiten. Implizit in der QREMOTE -Definition ist durch die Auslassung des Attributs XMITQ der Name der Übertragungswarteschlange QM3. QM3 ist jedoch derselbe Name, wie wir erwarten, mit dem Aliasnamen des Warteschlangenmanagers für den Rest des Clusters zugänglich zu machen. IBM MQ lässt nicht zu, dass Sie für die Übertragungswarteschlange und den Warteschlangenmanager-Alias denselben Namen vergeben. Eine Lösung besteht darin, eine Übertragungswarteschlange zu erstellen, mit der Nachrichten an QM3 mit einem anderen Namen an den WS-Manager-Aliasnamen weitergeleitet werden können.

   b) Geben Sie den Namen der Übertragungswarteschlange in der QREMOTE -Definition an.

   ```
   DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
   ```

   Der neue Aliasname des WS-Managers koppelt die neue Übertragungswarteschlange mit dem Namen QM3.XMIT mit dem Aliasnamen des QM3 -Warteschlangenmanagers. Es handelt sich um eine einfache und richtige Lösung, die aber nicht völlig zufriedenstellend ist. Sie hat die Namenskonvention für Übertragungswarteschlangen verletzt, die sie mit dem Namen des Zielwarteschlangenmanagers erhalten. Gibt es alternative Lösungen, die die Namenskonvention der Übertragungswarteschlange beibehalten?

   Das Problem tritt auf, weil der Anforderer standardmäßig die Übergabe von QM3 als Antwort-WS-Manager-Namen in der Anforderungsnachricht, die von QM3 gesendet wird, als Antwort anfordert. Der Server unter QM2 verwendet den Namen des QM3 -Antwortwarteschlangenmanagers, um QM3 in den Antworten zu adressgleichen. Die Lösung, die QM1 erforderlich ist, um QM3 als WS-Manager-Aliasnamen zugänglich zu machen, um Antwortnachrichten zurückzugeben, und verhindert, dass QM1 QM3 als Namen der Übertragungswarteschlange verwendet.

   Statt standardmäßig QM3 als Antwort-WS-Managernamen zu verwenden, müssen Anwendungen in QM3 einen WS-Manager-Aliasnamen für Antwortnachrichten an QM1 übergeben. Der Gateway-Warteschlangenmanager QM1 wirbt den Aliasnamen des Warteschlangenmanagers für Antworten auf QM3 und nicht für QM3 selbst, wodurch der Konflikt mit dem Namen der Übertragungswarteschlange vermieden wird.

   c) Definieren Sie einen WS-Manager-Aliasnamen für QM3 auf QM1.

   ```
   DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
   ```

Es sind zwei Änderungen an den Konfigurationsbefehlen erforderlich.

i) Der QREMOTE in QM1 wirbt jetzt für den WS-Manager-Aliasnamen QM3.ALIAS für den Rest des Clusters und koppelt ihn an den Namen des realen Warteschlangenmanagers QM3 an. QM3 ist wiederum der Name der Übertragungswarteschlange, an die Antwortwarteschlangen zurück an QM3 gesendet werden

ii) Die Clientanwendung muss QM3.ALIAS als Namen für den Antwortwarteschlangenmanager bereitstellen, wenn er die Anforderungsnachricht erstellt. Sie können QM3.ALIAS für die Clientanwendung auf eine von zwei Arten zur Verfügung stellen.

- Code QM3.ALIAS im Feld für den Namen des Antwortwarteschlangenmanagers, das von MQPUT in MQMDerstellt wird. Wenn Sie eine dynamische Warteschlange für Antworten verwenden, müssen Sie diese Vorgehensweise auf diese Weise ausführen.
- Verwenden Sie bei der Bereitstellung des Namens der Warteschlange für die Antwortwarteschlange einen Antwortwarteschlangenalias ( Q3.ALIAS) anstelle einer Warteschlange für Antwortwarteschlangen.

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

## Nächste Schritte

**Anmerkung:** Sie können die Verwendung von Aliasnamen für Empfangswarteschlangen für Antworten mit **AMQSREQ0**nicht veranschaulichen. Sie öffnet die Warteschlange für Antwortwarteschlangen unter Verwendung des Warteschlangennamens, der in Parameter 3 angegeben ist, oder die Standardmodellwarteschlange für SYSTEM.SAMPLE.REPLY . Sie müssen das Beispiel ändern, indem Sie einen weiteren Parameter angeben, der den Aliasnamen der Empfangswarteschlange für Antworten enthält, um den Aliasnamen des Antwortwarteschlangenmanagers für MQPUTzu benennen.

**Zugehörige Konzepte**

WS-Manager-Aliasnamen und -Cluster
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

Aliasnamen für Antwortwarteschlangen und Cluster
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

Warteschlangenaliasnamen und -cluster
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

**Zugehörige Tasks**

Request/Antwort von einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

Lastausgleich von außerhalb eines Clusters konfigurieren
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

Nachrichtenpfade zwischen Clustern konfigurieren
Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

„Namen eines Cluster-Ziel-WS-Managers ausblenden” auf Seite 406

Verlegen Sie eine Nachricht an eine Clusterwarteschlange, die in einem beliebigen WS-Manager in einem Cluster definiert ist, ohne den Warteschlangenmanager zu benennen.

*Namen eines Cluster-Ziel-WS-Managers ausblenden*
Verlegen Sie eine Nachricht an eine Clusterwarteschlange, die in einem beliebigen WS-Manager in einem Cluster definiert ist, ohne den Warteschlangenmanager zu benennen.

## Vorbereitende Schritte

- Vermeiden Sie es, die Namen von Warteschlangenmanagern zu enthüllen, die sich innerhalb des Clusters an Warteschlangenmanager befinden, die sich außerhalb des Clusters befinden.
  - Durch das Auflösen von Referenzen auf den Warteschlangenmanager, der als Host für eine Warteschlange im Cluster fungiert, wird die Flexibilität für den Lastausgleich aufgehoben.
  - Außerdem ist es für Sie schwierig, einen Warteschlangenmanager zu ändern, der als Host für eine Warteschlange im Cluster fungiert.
  - Alternativ können Sie RQMNAME durch einen WS-Manager-Aliasnamen ersetzen, der vom Clusteradministrator bereitgestellt wird.
  - „Namen eines Cluster-Ziel-WS-Managers ausblenden" auf Seite 406 beschreibt die Verwendung eines WS-Manager-Aliasnamens, um einen WS-Manager außerhalb eines Clusters von der Verwaltung von Warteschlangenmanagern in einem Cluster zu entkoppeln.
- Der vorgeschlagene Weg zum Namen von Übertragungswarteschlangen besteht jedoch darin, ihnen den Namen des Zielwarteschlangenmanagers zu geben. Der Name der Übertragungswarteschlange zeigt den Namen eines Warteschlangenmanagers im Cluster an. Sie müssen auswählen, welche Regel folgen soll. Sie können die Übertragungswarteschlange entweder mit dem Namen des WS-Managers oder mit dem Clusternamen benennen:

  **Benennen Sie die Übertragungswarteschlange mit dem Namen des Gateway-Warteschlangenmanagers.**
  Die Offenlegung des Namens des Gateway-WS-Managers an WS-Manager außerhalb eines Clusters ist eine sinnvolle Ausnahme von der Regel, die Namen von Clusterwarteschlangenmanagern zu verdecken.

  **Benennen Sie die Übertragungswarteschlange mit dem Namen des Clusters.**
  Wenn Sie die Konvention für die Benennung von Übertragungswarteschlangen mit dem Namen des Zielwarteschlangenmanagers nicht befolgen, verwenden Sie den Clusternamen.

## Informationen zu diesem Vorgang

Ändern Sie die Task „Anforderung/Antwort in einem Cluster konfigurieren" auf Seite 402, um den Namen des Zielwarteschlangenmanagers innerhalb des Clusters zu verdecken.

## Vorgehensweise

Definieren Sie im Beispiel Abbildung 54 auf Seite 407 einen WS-Manager-Aliasnamen auf dem Gateway-WS-Manager QM1 mit dem Namen DEMO:

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

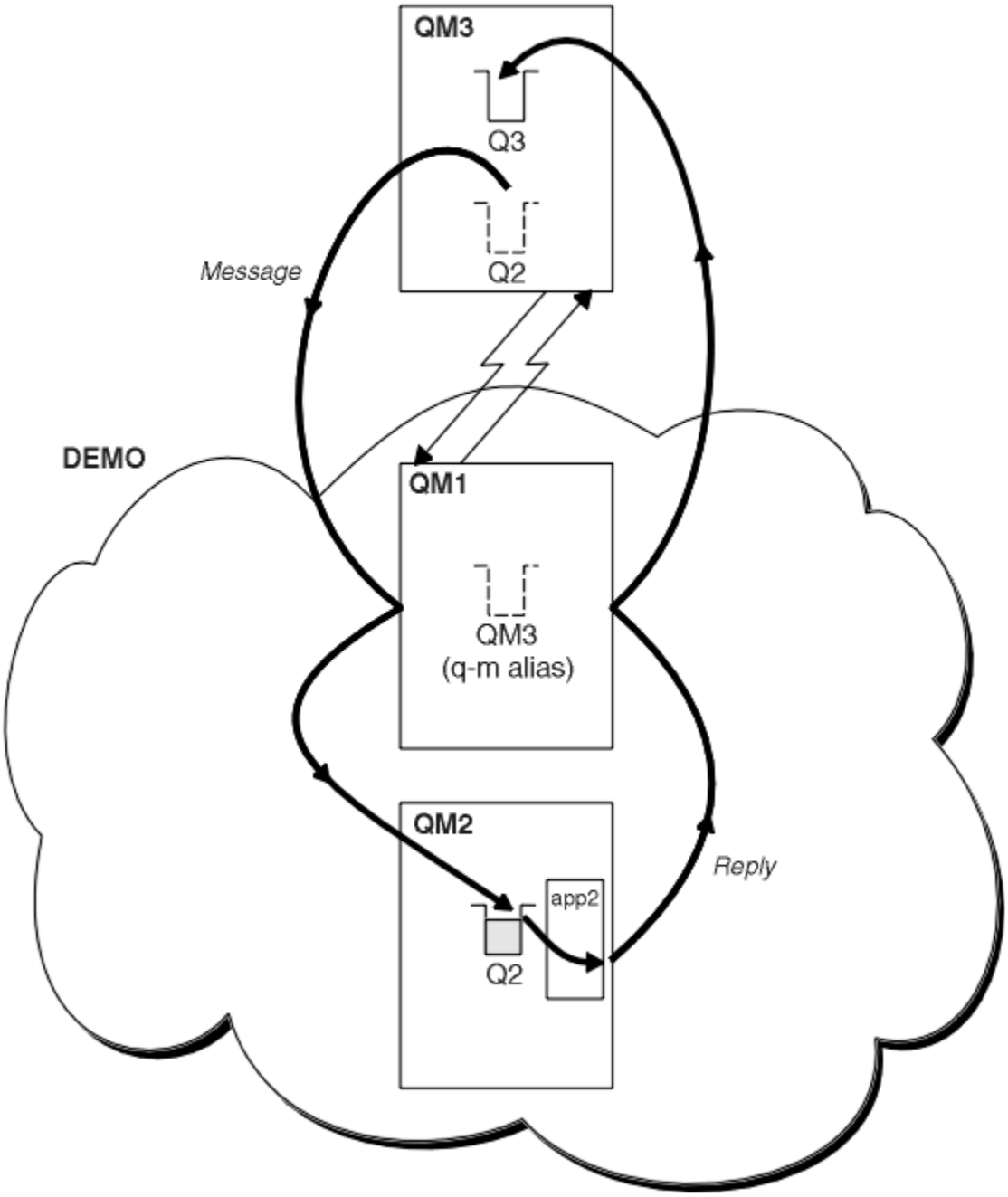



*Abbildung 54. Aus einem WS-Manager außerhalb des Clusters einschalten*

Die QREMOTE -Definition unter QM1 macht den Aliasnamen des Warteschlangenmanagers DEMO dem Gateway-Warteschlangenmanager bekannt. QM3 kann der Warteschlangenmanager außerhalb des Clusters den Warteschlangenmanager-Aliasnamen DEMO verwenden, um Nachrichten an Clusterwarteschlangen unter DEMOzu senden, anstatt einen tatsächlichen Warteschlangenmanagernamen verwenden zu müssen.

Wenn Sie die Konvention für die Verwendung des Clusternamens verwenden, um die Übertragungswarteschlange zu benennen, die eine Verbindung zu einem Cluster herstellen soll, wird die Definition der fernen Warteschlange für Q2 wie folgt:

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO) XMIT(DEMO)
```

## Ergebnisse

Nachrichten, die für Q2 auf DEMO bestimmt sind, werden in die Übertragungswarteschlange von DEMO gestellt. Aus der Übertragungswarteschlange werden sie vom senderkanal an den Gateway-Warteschlangenmanager QM1 übertragen. Der Gateway-Warteschlangenmanager leitet die Nachrichten an jeden Warteschlangenmanager im Cluster weiter, in dem sich die Clusterwarteschlange Q2 befindet.

### *Request/Antwort von einem Cluster konfigurieren*

Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

## Vorbereitende Schritte

Abbildung 55 auf Seite 409 zeigt einen Warteschlangenmanager ( QM2) innerhalb des Clusters DEMO. Er sendet eine Anforderung an eine Warteschlange Q3, die sich auf dem WS-Manager außerhalb des Clusters befindet. Die Antworten werden an Q2 im QM2 innerhalb des Clusters zurückgegeben.

Für die Kommunikation mit dem WS-Manager außerhalb des Clusters agieren mindestens ein Warteschlangenmanager im Cluster als Gateway. Ein Gateway-WS-Manager hat einen Kommunikationspfad zu den Warteschlangenmanagern außerhalb des Clusters. In dem Beispiel ist QM1 das Gateway.

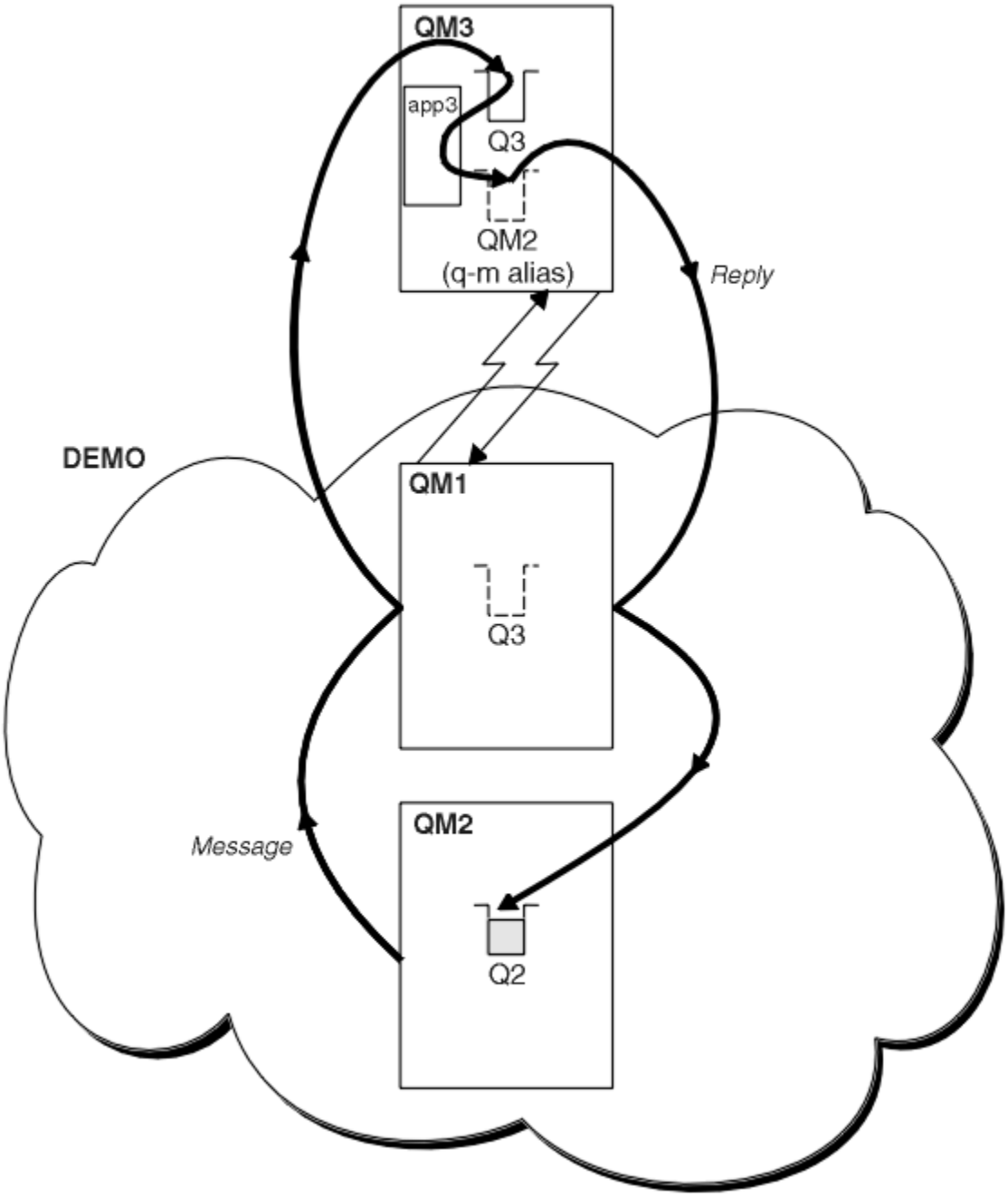



*Abbildung 55. In einen WS-Manager außerhalb des Clusters einschalten*

## Informationen zu diesem Vorgang

Befolgen Sie die Anweisungen zum Festlegen des Pfads für die Anforderungs-und Antwortnachrichten.

## Vorgehensweise

1. Senden Sie die Anforderungsnachricht aus dem Cluster.

   Überlegen Sie, wie der Warteschlangenmanager QM2, der sich im Cluster befindet, eine Nachricht in die Warteschlange Q3 von QM3 einreiht, die sich außerhalb des Clusters befindet.

   a) Erstellen Sie eine QREMOTE -Definition in QM1 , die die ferne Warteschlange Q3 für den Cluster zugänglich macht.

```
DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
```

Sie verfügt außerdem über einen Senderkanal und eine Übertragungswarteschlange zum Warteschlangenmanager, der sich außerhalb des Clusters befindet. QM3 verfügt über einen entsprechenden Empfängerkanal. Die Kanäle werden in Abbildung 55 auf Seite 409 nicht angezeigt.

Eine Anwendung unter QM2 gibt einen Aufruf MQPUT aus, der die Zielwarteschlange und die Warteschlange angibt, an die Antworten gesendet werden sollen. Die Zielwarteschlange ist Q3 und die Warteschlange für Antwortantworten ist Q2.

Die Nachricht wird an QM1 gesendet, die die Definition der fernen Warteschlange verwendet, um den Warteschlangennamen in Q3 auf QM3 aufzulösen.

2. Empfangen Sie die Antwortnachricht vom WS-Manager außerhalb des Clusters.

   Ein WS-Manager außerhalb des Clusters muss für jeden Warteschlangenmanager im Cluster, an den er eine Nachricht sendet, über einen Warteschlangenmanager-Aliasnamen verfügen. Der Aliasname des WS-Managers muss auch den Namen der Übertragungswarteschlange für den Gateway-Warteschlangenmanager angeben. In diesem Beispiel benötigt QM3 eine WS-Manager-Aliasdefinition für QM2:

   a) Erstellen Sie einen WS-Manager-Aliasnamen QM2 unter QM3.

   ```
   DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
   ```

   QM3 benötigt außerdem eine Sende-Channel-Warteschlange und eine Übertragungswarteschlange für QM1 und QM1 benötigt einen entsprechenden Empfängerkanal.

   Die Anwendung **app3**unter QM3 kann dann Antworten an QM2senden, indem sie einen MQPUT -Aufruf ausgibt und den Warteschlangennamen Q2 und den Namen des Warteschlangenmanagers QM2angibt.

## Nächste Schritte

Sie können mehr als eine Route aus einem Cluster definieren.

**Zugehörige Konzepte**

WS-Manager-Aliasnamen und -Cluster
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

Aliasnamen für Antwortwarteschlangen und Cluster
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

Warteschlangenaliasnamen und -cluster
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

**Zugehörige Tasks**

Anforderung/Antwort in einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

Lastausgleich von außerhalb eines Clusters konfigurieren
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

Nachrichtenpfade zwischen Clustern konfigurieren

Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

### *Lastausgleich von außerhalb eines Clusters konfigurieren*

Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

## Vorbereitende Schritte

Konfigurieren Sie das Beispiel, wie in Abbildung 53 auf Seite 403 in „Anforderung/Antwort in einem Cluster konfigurieren” auf Seite 402 dargestellt.

## Informationen zu diesem Vorgang

In diesem Szenario sendet der Warteschlangenmanager außerhalb des Clusters QM3 in Abbildung 56 auf Seite 412Anforderungen an die Warteschlange Q2. Q2 befindet sich auf zwei Warteschlangenmanagern, QM2 und QM4 im Cluster DEMO. Beide WS-Manager werden mit der Standardbindungsoption NOTFIXED konfiguriert, um den Lastausgleich zu verwenden. Die Anforderungen von QM3, des Warteschlangenmanagers außerhalb des Clusters, werden an eine Instanz von Q2 über QM1 gesendet.

QM3 ist nicht Teil eines Clusters und kommuniziert mit verteilten Warteschlangenverfahren. Er muss über einen Senderkanal und eine Übertragungswarteschlange zu QM1 verfügen. QM1 benötigt einen entsprechenden Empfängerkanal. Die Kanäle und Übertragungswarteschlangen werden in Abbildung 56 auf Seite 412 nicht explizit angezeigt.

Die Prozedur erweitert das Beispiel in Abbildung 53 auf Seite 403 in „Anforderung/Antwort in einem Cluster konfigurieren” auf Seite 402.

## Vorgehensweise

1. Erstellen Sie eine QREMOTE -Definition für Q2 auf QM3.

   ```
   DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
   ```

   Erstellen Sie eine QREMOTE -Definition für jede Warteschlange in dem Cluster, in der QM3 Nachrichten einreiht.
2. Erstellen Sie einen WS-Manager-Aliasnamen Q3 in QM1.

   ```
   DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
   ```

   Q3 ist kein echter WS-Manager-Name. Dies ist der Name einer WS-Manager-Aliasdefinition im Cluster, die den Aliasnamen des Warteschlangenmanagers Q3 mit Leerzeichen, ' ', entspricht.
3. Definieren Sie eine lokale Warteschlange mit dem Namen Q2 in den einzelnen QM2 und QM4.

   ```
   DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
   ```
4. QM1, der Gateway-Warteschlangenmanager, enthält keine speziellen Definitionen.

## Ergebnisse

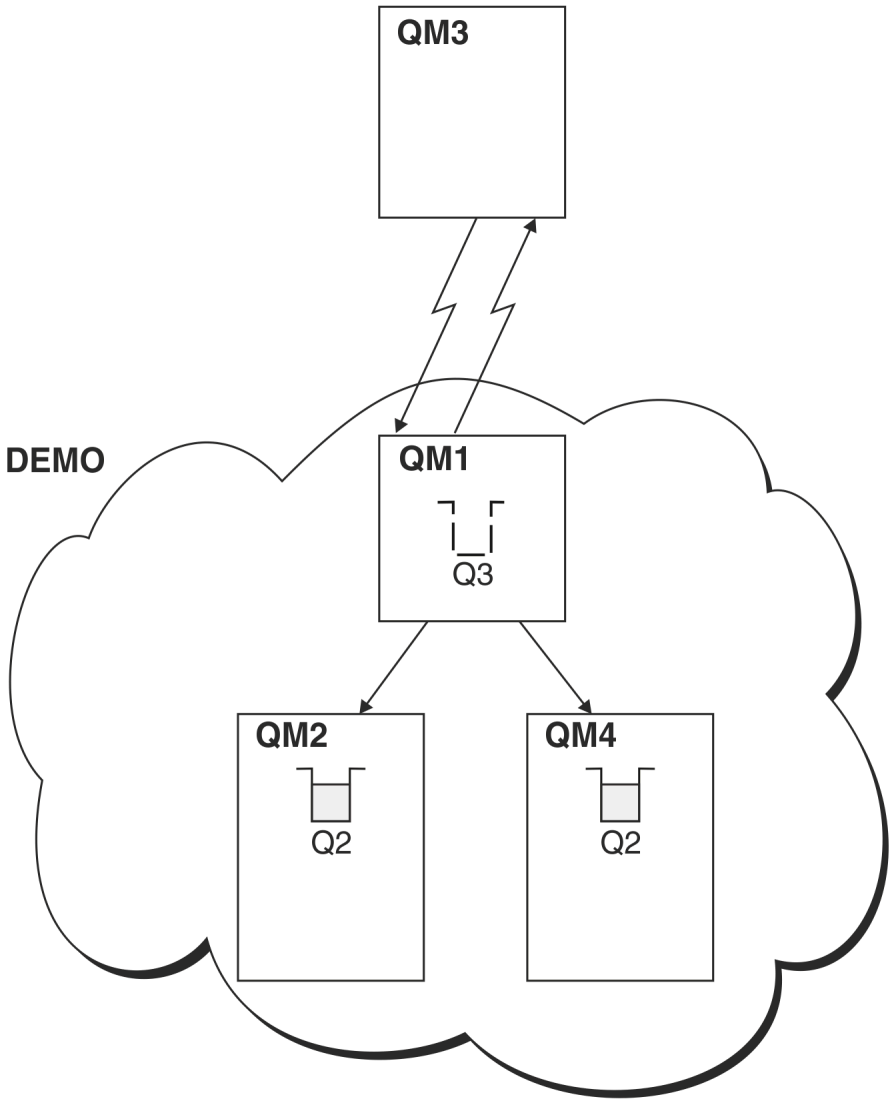



*Abbildung 56. Aus einem WS-Manager außerhalb des Clusters einschalten*

Wenn eine Anwendung unter QM3 einen MQPUT -Aufruf ausgibt, um eine Nachricht in Q2einzureihen, bewirkt die Definition QREMOTE unter QM3 , dass die Nachricht über den Gateway-Warteschlangenmanager QM1weitergeleitet wird. Wenn QM1 die Nachricht empfängt, ist es sich bewusst, dass die Nachricht immer noch für eine Warteschlange mit dem Namen Q2 bestimmt ist und die Namensauflösung ausführt. QM1 überprüft seine lokalen Definitionen und findet keine für Q2. QM1 überprüft dann die Clusterkonfiguration und stellt fest, dass es zwei Instanzen von Q2 im Cluster DEMO kennt. QM1 kann jetzt den Lastausgleich verwenden, um Nachrichten zwischen den Instanzen von Q2 , die sich auf QM2 und QM4 befinden, zu verteilen.

**Zugehörige Konzepte**

WS-Manager-Aliasnamen und -Cluster
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

Aliasnamen für Antwortwarteschlangen und Cluster
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

Warteschlangenaliasnamen und -cluster
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

Namensauflösung

**Zugehörige Tasks**

Anforderung/Antwort in einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

Request/Antwort von einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

Nachrichtenpfade zwischen Clustern konfigurieren
Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

**Zugehörige Verweise**

Auflösung des Warteschlangennamens

## *Nachrichtenpfade zwischen Clustern konfigurieren*

Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

## Informationen zu diesem Vorgang

Anstatt alle Warteschlangenmanager in einem großen Cluster zu gruppieren, können Sie viele kleinere Cluster haben. Jeder Cluster verfügt über einen oder mehrere Warteschlangenmanager, die als Brücke fungieren. Dies hat den Vorteil, dass Sie die Sichtbarkeit von Warteschlangen-und Warteschlangenmanagernamen in den Clustern einschränken können. Siehe Überlappende Cluster (Overlapping clusters) . Verwenden Sie Aliasnamen, um die Namen von Warteschlangen und Warteschlangenmanagern zu ändern, um Namensunverträglichkeiten zu vermeiden oder um die lokalen Namenskonventionen einzuhalten.

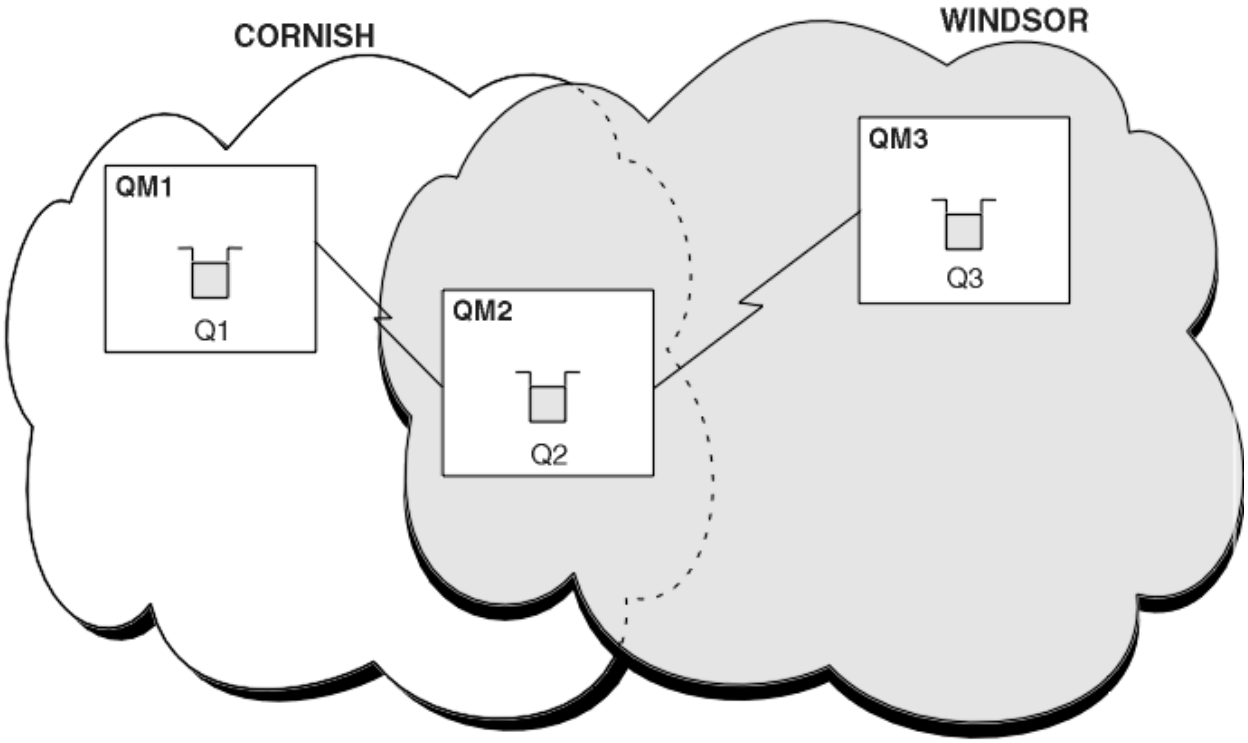



*Abbildung 57. Clusterübergreifende Überbrückung*

Abbildung 57 auf Seite 414 zeigt zwei Cluster mit einer Brücke zwischen ihnen. Es könnte mehr als eine Brücke vorhanden sein.

Konfigurieren Sie die Cluster mit der folgenden Prozedur:

## Vorgehensweise

1. Definieren Sie eine Clusterwarteschlange Q1 auf QM1.

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. Definieren Sie eine Clusterwarteschlange Q3 auf QM3.

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. Erstellen Sie eine Namensliste mit dem Namen CORNISHWINDSOR unter QM2, die die Namen beider Cluster enthält.

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')
NAMES(CORNISH, WINDSOR)
```

4. Definieren Sie eine Clusterwarteschlange, Q2 auf QM2

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

## Nächste Schritte

QM2 ist ein Member der beiden Cluster und ist die Brücke zwischen den beiden Clustern. Für jede Warteschlange, die Sie über die Brücke hinweg sichtbar machen möchten, benötigen Sie eine QALIAS

-Definition auf der Brücke. In Abbildung 57 auf Seite 414 unter QM2 benötigen Sie beispielsweise Folgendes:

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

Mithilfe des Warteschlangenalias kann eine Anwendung, die mit einem Warteschlangenmanager in CORNISH verbunden ist (z. B. QM1), eine Nachricht an Q3 senden. Er bezieht sich auf Q3 als MYQ3. Die Nachricht wird an Q3 unter QM3 weitergeleitet.

Wenn Sie eine Warteschlange öffnen, müssen Sie DEFBIND entweder auf NOTFIXED oder auf QDEF setzen. Wenn DEFBIND als Standardwert ( OPEN) belassen wird, löst der Warteschlangenmanager die Aliasdefinition in den Brückenwarteschlangenmanager auf, in dem sie enthalten ist. Die Brücke leitet die Nachricht nicht weiter.

Für jeden Warteschlangenmanager, den Sie sichtbar machen möchten, benötigen Sie eine WS-Manager-Aliasnamendefinition. Unter QM2 benötigen Sie z. B.:

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

Eine Anwendung, die mit einem beliebigen WS-Manager in WINDSOR verbunden ist, z. B. QM3, kann eine Nachricht in jede Warteschlange unter QM1 stellen, indem sie QM1 explizit im Aufruf MQOPEN benennt.

**Zugehörige Konzepte**

WS-Manager-Aliasnamen und -Cluster
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

Aliasnamen für Antwortwarteschlangen und Cluster
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

Warteschlangenaliasnamen und -cluster
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

**Zugehörige Tasks**

Anforderung/Antwort in einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

Request/Antwort von einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

Lastausgleich von außerhalb eines Clusters konfigurieren
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

## *WS-Manager-Aliasnamen und -Cluster*

Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

WS-Manager-Aliasnamen, die unter Verwendung einer Definition einer fernen Warteschlange mit einem leeren RNAME erstellt werden, haben fünf Verwendungen:

**Namen des WS-Managers beim Senden von Nachrichten neu zuordnen**

Ein Warteschlangenmanager-Aliasname kann verwendet werden, um den in einem MQOPEN -Aufruf angegebenen Warteschlangenmanagernamen einem anderen Warteschlangenmanager neu zuzuordnen. Es kann sich um einen Cluster-WS-Manager handeln. Beispielsweise kann ein Warteschlangenmanager die Definition des WS-Manager-Aliasnamens haben:

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

YORK kann als Aliasname für den WS-Manager CLUSQM verwendet werden. Wenn eine Anwendung auf dem Warteschlangenmanager, die diese Definition vorgenommen hat, eine Nachricht an den Warteschlangenmanager YORK stellt, löst der lokale WS-Manager den Namen in CLUSQM auf. Wenn der lokale WS-Manager nicht CLUSQM genannt wird, wird die Nachricht in die Clusterübertragungswarteschlange gestellt, die in CLUSQM verschoben werden soll. Außerdem ändert er den Übertragungsheader, um CLUSQM anstelle von YORK zu verwenden.

**Anmerkung:** Die Definition gilt nur für den Warteschlangenmanager, der die Definition des Warteschlangenmanagers vornimmt. Um den Aliasnamen für den gesamten Cluster zugänglich zu machen, müssen Sie das Attribut CLUSTER zur Definition der fernen Warteschlange hinzufügen. Anschließend werden Nachrichten von anderen Warteschlangenmanagern, die für YORK bestimmt waren, an CLUSQM gesendet.

**Ändern oder Angeben der Übertragungswarteschlange beim Senden von Nachrichten**

Das Aliasing kann verwendet werden, um einen Cluster in ein Nicht-Clustersystem zu verknüpfen. Beispielsweise können Warteschlangenmanager im Cluster ITALY mit dem Warteschlangenmanager PALERMO kommunizieren, der sich außerhalb des Clusters befindet. Für die Kommunikation muss einer der WS-Manager im Cluster als Gateway fungieren. Geben Sie im Gateway-WS-Manager den folgenden Befehl aus:

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

Der Befehl ist eine Aliasdefinition des Warteschlangenmanagers. Sie definiert und macht ROME als Warteschlangenmanager bekannt, über den Nachrichten von jedem WS-Manager im Cluster ITALY mit mehreren Hops zu ihrem Ziel in PALERMO gelangen können. Nachrichten, die in eine Warteschlange gestellt werden, die mit dem Namen des Warteschlangenmanagers, der auf ROME gesetzt ist, geöffnet wurde, werden mit der WS-Manager-Aliasdefinition an den Warteschlangenmanager-Warteschlangenmanager gesendet. Dort werden die Nachrichten in die Übertragungswarteschlange X gestellt und von Nicht-Clusterkanälen in den Warteschlangenmanager PALERMO verschoben.

Die Auswahl des Namens ROME in diesem Beispiel ist nicht signifikant. Die Werte für QREMOTE und RQMNAME können beide identisch sein.

**Bestimmung des Ziels beim Empfangen von Nachrichten**

Wenn ein WS-Manager eine Nachricht empfängt, extrahiert er den Namen der Zielwarteschlange und des Warteschlangenmanagers aus dem Übertragungsheader. Sie sucht nach einer WS-Manager-Aliasnamendefinition mit demselben Namen wie der Warteschlangenmanager im Übertragungsheader. Wenn er eine findet, ersetzt er den Warteschlangenmanagernamen im Übertragungsheader durch den RQMNAME aus der WS-Manager-Aliasdefinition.

Es gibt zwei Gründe für die Verwendung eines WS-Manager-Aliasnamens auf diese Weise:

- Nachrichten an einen anderen WS-Manager zu leiten
- Um den Namen des Warteschlangenmanagers zu ändern, der mit dem lokalen WS-Manager identisch sein soll

**WS-Manager-Aliasnamen in einem Gateway-WS-Manager verwenden, um Nachrichten zwischen Warteschlangenmanagern in verschiedenen Clustern weiterzuleiten.**

Eine Anwendung kann mithilfe eines WS-Manager-Aliasnamens eine Nachricht an eine Warteschlange in einem anderen Cluster senden. Die Warteschlange muss keine Clusterwarteschlange sein. Die War-

teschlange ist in einem Cluster definiert. Die Anwendung ist mit einem WS-Manager in einem anderen Cluster verbunden. Ein Gateway-WS-Manager verbindet die beiden Cluster. Wenn die Warteschlange nicht als Cluster-Cluster definiert ist, muss die Anwendung die Warteschlange unter Verwendung des Warteschlangennamens und des Aliasnamens eines Cluster-WS-Managers öffnen, damit die korrekte Weiterleitung erfolgt. Ein Beispiel für eine Konfiguration finden Sie in „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager" auf Seite 364, von der aus der in Abbildung 1 dargestellte Antwortnachrichtenfluss ausgeführt wird.

Das Diagramm zeigt den Pfad, den die Antwortnachricht zurück zu einer temporären dynamischen Warteschlange mit dem Namen RQ nimmt. Die Serveranwendung, die mit QM3 verbunden ist, öffnet die Antwortwarteschlange unter Verwendung des Warteschlangenmanagernamens QM2. Der Warteschlangenmanagername QM2 ist unter QM1 als Aliasname eines Clusterwarteschlangenmanagers definiert. QM3 leitet die Antwortnachricht an QM1 weiter. QM1 leitet die Nachricht an QM2 weiter.

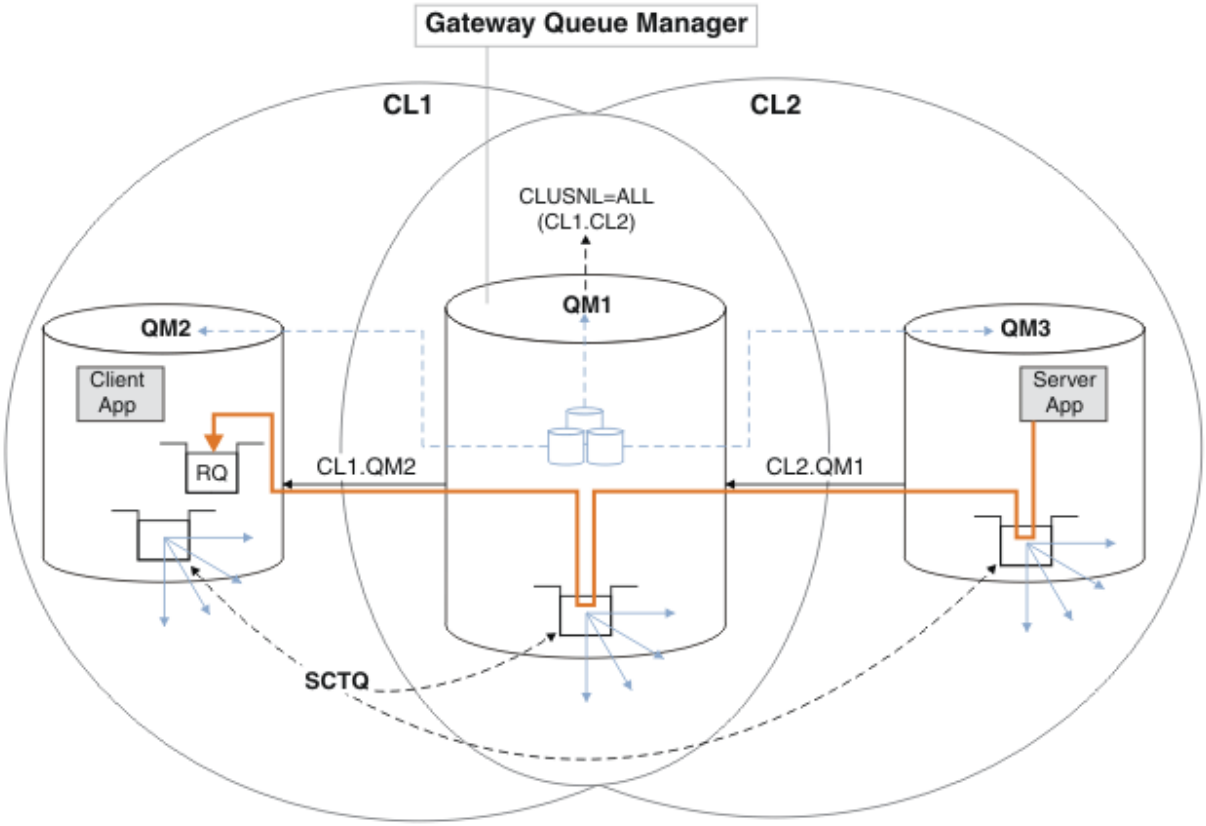



*Abbildung 58. Verwenden eines Warteschlangenmanager-Aliasnamens, um die Antwortnachricht an einen anderen Cluster zurückzugeben*

Die Art und Weise, wie das Routing funktioniert, ist wie folgt. Jeder Warteschlangenmanager in jedem Cluster verfügt über eine Warteschlangenmanager-Aliasnamensdefinition unter QM1. Die Aliasnamen werden in allen Clustern gruppiert. Die grauen gestrichelten Pfeile von jedem der Aliasnamen zu einem Warteschlangenmanager zeigen, dass jeder Warteschlangenmanager-Aliasname in mindestens einem der Cluster in einen echten Warteschlangenmanager aufgelöst wird. In diesem Fall wird der Aliasname QM2 sowohl in Cluster CL1 als auch in CL2 zusammengefasst und in den realen Warteschlangenmanager QM2 in CL1 aufgelöst. Die Serveranwendung erstellt die Antwortnachricht unter Verwendung des Namens der Empfangswarteschlange für Antworten RQ und des Namens des Antwortwarteschlangenmanagers QM2. Die Nachricht wird an QM1 weitergeleitet, da die Warteschlangenmanager-Aliasdefinition QM2 in QM1 im Cluster CL2 definiert ist und der Warteschlangenmanager QM2 nicht im Cluster CL2 ist. Da die Nachricht nicht an den Zielwarteschlangenmanager gesendet werden kann, wird sie an den Warteschlangenmanager gesendet, der die Aliasdefinition enthält.

QM1 stellt die Nachricht in die Clusterübertragungswarteschlange unter QM1 zur Übertragung an QM2. QM1 leitet die Nachricht an QM2 weiter, weil die Warteschlangenmanager-Aliasdefinition unter QM1

für QM2 QM2 als realen Zielwarteschlangenmanager definiert. Die Definition ist nicht kreisförmig, da die Aliasdefinitionen nur auf reale Definitionen verweisen können. Der Aliasname kann nicht auf sich selbst verweisen. Die reale Definition wird von QM1 aufgelöst, da sich sowohl QM1 als auch QM2 in demselben Cluster befinden: CL1. QM1 ermittelt die Verbindungsinformationen für QM2 aus dem Repository für CL1 und leitet die Nachricht an QM2 weiter. Damit die Nachricht von QM1 weitergeleitet wird, muss die Serveranwendung die Antwortwarteschlange mit der Option DEFBIND auf MQBND_BIND_NOT_FIXED geöffnet haben. Wenn die Serveranwendung die Antwortwarteschlange mit der Option MQBND_BIND_ON_OPEN geöffnet hat, wird die Nachricht nicht weitergeleitet und in eine Warteschlange für nicht zustellbare Nachrichten eingereiht.

**Die Verwendung eines Warteschlangenmanagers als Gateway in den Cluster für die Lastverteilung von Nachrichten von außerhalb des Clusters aus.**

Sie definieren eine Warteschlange mit dem Namen EDINBURGH in mehr als einem Warteschlangenmanager im Cluster. Sie möchten, dass der Clustering-Mechanismus die Auslastung für Nachrichten, die von außerhalb des Clusters in diese Warteschlange einreisen, ausgleichen soll.

Ein Warteschlangenmanager von außerhalb des Clusters benötigt eine Übertragungswarteschlange und einen Senderkanal zu einem WS-Manager im Cluster. Diese Warteschlange wird als Gateway-WS-Manager bezeichnet. Um den standardmäßigen Lastausgleichsmechanismus nutzen zu können, muss eine der folgenden Regeln gelten:

- Der Gateway-WS-Manager darf keine Instanz der EDINBURGH -Warteschlange enthalten.
- Der Gateway-WS-Manager gibt CLWLUSEQ(ANY) auf ALTER QMGR an.

Ein Beispiel für einen Lastausgleich von außerhalb eines Clusters finden Sie in „Lastausgleich von außerhalb eines Clusters konfigurieren” auf Seite 411.

**Zugehörige Konzepte**

Aliasnamen für Antwortwarteschlangen und Cluster
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

Warteschlangenaliasnamen und -cluster
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

**Zugehörige Tasks**

Anforderung/Antwort in einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

Request/Antwort von einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

Lastausgleich von außerhalb eines Clusters konfigurieren
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

Nachrichtenpfade zwischen Clustern konfigurieren
Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

## *Aliasnamen für Antwortwarteschlangen und Cluster*

Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

For example:

- Eine Anwendung im Warteschlangenmanager VENICE sendet mit dem Aufruf MQPUT eine Nachricht an den Warteschlangenmanager PISA . Die Anwendung stellt die folgenden Antworten auf die Antwortwarteschlange im Nachrichtendeskriptor bereit:

```
ReplyToQ='QUEUE'
ReplyToQMgr=''
```

- Damit die Antworten, die an QUEUE gesendet werden, unter PISA auf OTHERQ empfangen werden können, erstellen Sie eine Definition der fernen Warteschlange in VENICE , die als Aliasname für die Antwortwarteschlange verwendet wird. Der Aliasname ist nur auf dem System wirksam, auf dem es erstellt wurde.

```
DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
```

RQMNAME und QREMOTE können dieselben Namen angeben, auch wenn RQMNAME selbst ein Cluster-WS-Manager ist.

**Zugehörige Konzepte**

WS-Manager-Aliasnamen und -Cluster
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

Warteschlangenaliasnamen und -cluster
Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

**Zugehörige Tasks**

Anforderung/Antwort in einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

Request/Antwort von einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

Lastausgleich von außerhalb eines Clusters konfigurieren
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

Nachrichtenpfade zwischen Clustern konfigurieren
Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

## *Warteschlangenaliasnamen und -cluster*

Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffssteuerungen zu übernehmen.

Eine QALIAS -Definition wird verwendet, um einen Aliasnamen zu erstellen, über den eine Warteschlange bekannt sein soll. Sie können einen Aliasnamen aus einer Reihe von Gründen erstellen:

- Sie möchten mit der Verwendung einer anderen Warteschlange beginnen, aber Sie möchten Ihre Anwendungen nicht ändern.
- Sie möchten nicht, dass Anwendungen den tatsächlichen Namen der Warteschlange, in die sie Nachrichten einreihen, kennen.
- Es kann eine Namenskonvention vorhanden sein, die sich von der Namenskonvention unterscheidet, in der die Warteschlange definiert ist.
- Ihre Anwendungen sind möglicherweise nicht berechtigt, auf die Warteschlange durch ihren tatsächlichen Namen zuzugreifen, sondern nur durch ihren Aliasnamen.

Erstellen Sie mit dem Befehl DEFINE QALIAS eine QALIAS -Definition auf einem Warteschlangenmanager. Führen Sie z. B. den folgenden Befehl aus:

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

Der Befehl wirbt für eine Warteschlange mit dem Namen PUBLIC für die Warteschlangenmanager im Cluster C. PUBLIC ist ein Aliasname, der in die Warteschlange namens LOCALaufgelöst wird. Nachrichten, die an PUBLIC gesendet werden, werden an die Warteschlange LOCAL weitergeleitet.

Sie können auch eine Warteschlangenaliasdefinition verwenden, um einen Warteschlangennamen in eine Clusterwarteschlange aufzulösen. Führen Sie z. B. den folgenden Befehl aus:

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

Mit dem Befehl kann ein Warteschlangenmanager den Namen PRIVATE verwenden, um auf eine Warteschlange zuzugreifen, die an anderer Stelle im Cluster durch den Namen PUBLIC angezeigt wird. Da diese Definition das Attribut CLUSTER nicht enthält, gilt sie nur für den Warteschlangenmanager, der sie macht.

**Zugehörige Konzepte**

WS-Manager-Aliasnamen und -Cluster
Verwenden Sie WS-Manager-Aliasnamen, um den Namen von Warteschlangenmanagern zu verdecken, wenn Nachrichten an einen Cluster gesendet oder aus einem Cluster gesendet werden, und die Nachrichten, die an einen Cluster gesendet werden, in Lastausgleichsnachrichten gesendet werden

Aliasnamen für Antwortwarteschlangen und Cluster
Eine Aliasdefinition für die Warteschlange für Antwortwarteschlangen wird verwendet, um alternative Namen für Antwortinformationen anzugeben. Definitionen von Warteschlangen für Antwortwarteschlangen können mit Clustern verwendet werden, die in einer verteilten Warteschlangenumgebung identisch sind.

**Zugehörige Tasks**

Anforderung/Antwort in einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem WS-Manager außerhalb eines Clusters. Verdecken Sie die inneren Details des Clusters, indem Sie einen Gateway-WS-Manager als Kommunikationspfad zu und vom Cluster verwenden.

Request/Antwort von einem Cluster konfigurieren
Konfigurieren Sie einen Anforderungs-/Antwortnachrichtenpfad von einem Cluster zu einem WS-Manager außerhalb des Clusters. Verdecken Sie die Details dazu, wie ein Warteschlangenmanager innerhalb des Clusters über einen Gateway-Warteschlangenmanager außerhalb des Clusters kommuniziert.

Lastausgleich von außerhalb eines Clusters konfigurieren
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

Nachrichtenpfade zwischen Clustern konfigurieren
Verbinden Sie Cluster unter Verwendung eines Gateway-Warteschlangenmanagers. Stellen Sie Warteschlangen oder Warteschlangenmanager für alle Cluster sichtbar, indem Sie Clusterwarteschlangen-oder Cluster-WS-Manager-Aliasnamen auf dem Gateway-Warteschlangenmanager definieren.

## Cluster für Workload-Management verwenden

Indem Sie mehrere Instanzen einer Warteschlange auf verschiedenen Warteschlangenmanagern in einem Cluster definieren, können Sie die Arbeit der Wartung der Warteschlange auf mehrere Server verteilen. Es gibt mehrere Faktoren, die verhindern können, dass Nachrichten in einem anderen Warteschlangenmanager in den Fall eines Fehlers erneut gestellt werden.

Neben der Konfiguration von Clustern zur Reduzierung der Systemverwaltung können Sie Cluster erstellen, in denen mehr als ein Warteschlangenmanager eine Instanz derselben Warteschlange hostet.

Sie können Ihren Cluster so organisieren, dass die Warteschlangenmanager in ihm klone einander sind. Jeder Warteschlangenmanager ist in der Lage, dieselben Anwendungen auszuführen und lokale Definitionen derselben Warteschlangen zu verwenden. z/OS In einem parallelen Sysplex von z/OS können die geklonten Anwendungen beispielsweise auf Daten in einer gemeinsam genutzten Db2- oder VSAM-Datenbank (Virtual Storage Access Method) zugreifen. Sie können die Workload zwischen Ihren Warteschlangenmanagern verteilen, indem Sie mehrere Instanzen einer Anwendung verwenden. Jede Instanz der Anwendung empfängt Nachrichten und wird unabhängig von den anderen ausgeführt.

Die Vorteile der Verwendung von Clustern auf diese Weise sind wie folgt:

- Erhöhte Verfügbarkeit Ihrer Warteschlangen und Anwendungen.
- Schneller Durchsatz von Nachrichten.
- Mehr gleichmäßige Verteilung der Auslastung in Ihrem Netzwerk.

Jeder Warteschlangenmanager, der eine Instanz einer bestimmten Warteschlange hostet, kann Nachrichten verarbeiten, die für diese Warteschlange bestimmt sind, und Anwendungen benennen keinen Warteschlangenmanager, wenn sie Nachrichten senden. Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Geeignete Ziele werden basierend auf der Verfügbarkeit des Warteschlangenmanagers und der Warteschlange sowie auf einer Reihe von workloadspezifischen Attributen ausgewählt, die Warteschlangenmanagern, Warteschlangen und Kanälen zugeordnet sind. Siehe Lastausgleich in Clustern .

z/OS In IBM MQ for z/OS können Warteschlangenmanager, die sich in Gruppen mit gemeinsamer Warteschlange befinden, Clusterwarteschlangen als gemeinsam genutzte Warteschlange hosten. Gemeinsam genutzte Clusterwarteschlangen sind für alle Warteschlangen in der gleichen Gruppe mit gemeinsamer Warteschlange verfügbar. Beispiel: In Ein Cluster mit mehreren Instanzen derselben Warteschlange kann es sich bei beiden oder beiden der Warteschlangenmanager QM2 und QM4 um einen gemeinsam genutzten Warteschlangenmanager handeln. Jeder verfügt über eine Definition für die Warteschlange Q3. Jeder Warteschlangenmanager, der sich in der gleichen Gruppe mit gemeinsamer Warteschlange wie QM4 befindet, kann eine Nachricht lesen, die in die gemeinsam genutzte Warteschlange Q3 eingereiht wurde. Jede Gruppe mit gemeinsamer Warteschlange kann bis zu 32 Warteschlangenmanager mit Zugriff auf die gleichen Daten enthalten. Durch die gemeinsame Nutzung von Warteschlangen wird der Durchsatz Ihrer Nachrichten erheblich gesteigert.

Weitere Informationen zu Clusterkonfigurationen für das Workload-Management finden Sie in den folgenden Unterabschnitten:

**Zugehörige Konzepte**

Vergleich von Clustering und verteilter Steuerung von Warteschlangen

Verteilte Warteschlangen und Cluster

Komponenten eines Clusters

Clusterkanäle

Was passiert, wenn eine Clusterwarteschlange für MQPUT inaktiviert ist

Der Lastausgleichssatz auf einem Clustersenderkanal funktioniert nicht.

„Nachrichten an und von Clustern weiterleiten” auf Seite 401

Verwenden Sie Warteschlangenaliasnamen, WS-Manager-Aliasnamen und Definitionen ferner Warteschlangen, um Cluster mit externen Warteschlangenmanagern und anderen Clustern zu verbinden.

**Zugehörige Tasks**

Exits für Clusterauslastung schreiben und kompilieren

„WS-Manager-Cluster konfigurieren” auf Seite 312
Cluster bieten einen Mechanismus für die Verbindung von Warteschlangenmanagern in einer Weise, die sowohl die Erstkonfiguration als auch die laufende Verwaltung vereinfacht. Sie können Cluster-Komponenten definieren und Cluster erstellen und verwalten.

„Neuen Cluster einrichten” auf Seite 327
Führen Sie die folgenden Anweisungen aus, um den Beispielcluster zu konfigurieren. In separaten Anweisungen wird beschrieben, wie der Cluster auf TCP/IP, LU 6.2 und mit einer einzelnen Übertragungswarteschlange oder mehreren Übertragungswarteschlangen eingerichtet wird. Testen Sie den Cluster, indem Sie eine Nachricht von einem WS-Manager an den anderen Warteschlangenmanager senden.

„Lastausgleich von außerhalb eines Clusters konfigurieren” auf Seite 411
Konfigurieren Sie einen Nachrichtenpfad von einem WS-Manager außerhalb eines Clusters in eine beliebige Kopie einer Clusterwarteschlange. Das Ergebnis ist eine Auslastungsabgleichsanforderungen von außerhalb des Clusters an die einzelnen Instanzen einer Clusterwarteschlange.

**Zugehörige Verweise**
Das Beispielprogramm 'Clusterwarteschlangenüberwachung' (AMQSCLM)

## *Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange*

In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Abbildung 59 auf Seite 423 zeigt einen Cluster, in dem mehr als eine Definition für die Warteschlange `Q3` vorhanden ist. Wenn eine Anwendung von `QM1` eine Nachricht in `Q3` einreiht, weiß sie nicht unbedingt, welche Instanz von `Q3` ihre Nachricht verarbeiten wird. Wenn eine Anwendung unter `QM2` oder `QM4`, ausgeführt wird, wenn lokale Instanzen von `Q3` vorhanden sind, wird die lokale Instanz von `Q3` standardmäßig geöffnet. Durch die Festlegung des Warteschlangenattributs `CLWLUSEQ` kann die lokale Instanz der Warteschlange wie eine ferne Instanz der Warteschlange behandelt werden.

Die Option `MQOPEN` `DefBind` steuert, ob der Zielwarteschlangenmanager ausgewählt wird, wenn der Aufruf `MQOPEN` ausgegeben wird oder wenn die Nachricht aus der Übertragungswarteschlange übertragen wird.

Wenn Sie `DefBind` auf `MQBND_BIND_NOT_FIXED` setzen, kann die Nachricht an eine Instanz der Warteschlange gesendet werden, die verfügbar ist, wenn die Nachricht übertragen wird. Dadurch werden die folgenden Probleme vermieden:

- Die Zielwarteschlange ist nicht verfügbar, wenn die Nachricht auf dem Zielwarteschlangenmanager ankommt.
- Der Status der Warteschlange wurde geändert.
- Die Nachricht wurde mit einem Aliasnamen der Clusterwarteschlange verwendet, und es ist keine Instanz der Zielwarteschlange auf dem Warteschlangenmanager vorhanden, auf dem die Instanz des Aliasnamens der Clusterwarteschlange definiert ist.

Wenn diese Probleme bei der Übertragungszeit erkannt werden, wird eine andere verfügbare Instanz der Zielwarteschlange gesucht, und die Nachricht wird erneut weitergeleitet. Wenn keine Instanzen der Warteschlange verfügbar sind, wird die Nachricht in die Warteschlange für dead-letter gestellt.

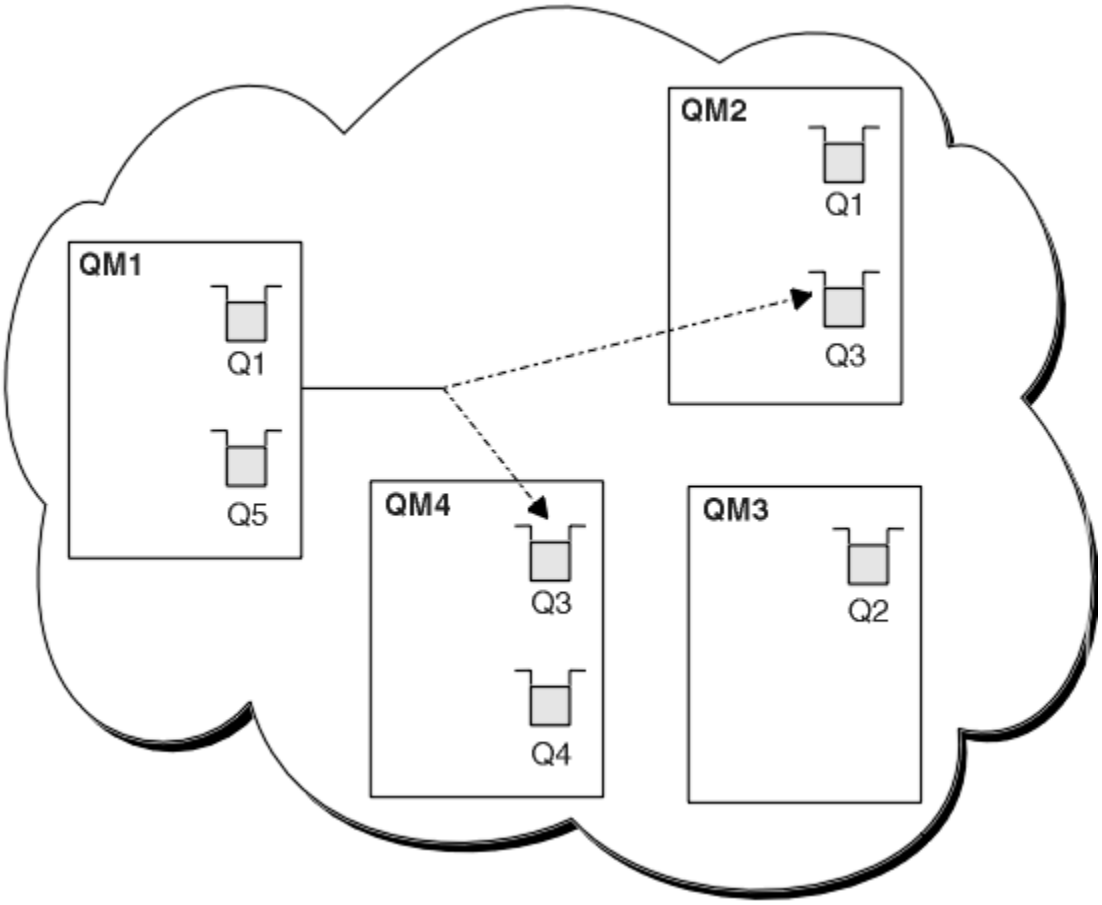



*Abbildung 59. Ein Cluster mit mehreren Instanzen derselben Warteschlange*

Ein Faktor, der verhindern kann, dass Nachrichten weitergeleitet werden, ist, wenn Nachrichten einem festen Warteschlangenmanager oder Kanal mit `MQBND_BIND_ON_OPEN` zugeordnet wurden. Nachrichten, die an `MQOPEN` gebunden sind, werden nie einem anderen Kanal zugeordnet. Beachten Sie auch, dass die Nachrichtenzuordnung nur dann stattfindet, wenn ein Clusterkanal tatsächlich fehlschlägt. Die Neuzuordnung tritt nicht auf, wenn der Kanal bereits ausgefallen ist.

Das System versucht, eine Nachricht erneut zu senden, wenn der Zielwarteschlangenmanager nicht mehr in Betrieb ist. In diesem Bereich wirkt sich dies nicht auf die Integrität der Nachricht aus, indem die Gefahr besteht, dass die Nachricht verloren geht oder ein Duplikat erstellt wird. Wenn ein WS-Manager fehlschlägt und eine Nachricht im Zweifel bleibt, wird diese Nachricht nicht weitergeleitet.

z/OS Unter IBM MQ for z/OS wird der Kanal erst vollständig gestoppt, wenn der Nachrichtenneuzuordnungsprozess abgeschlossen ist. Wenn Sie den Kanal mit dem Modus `FORCE` oder `TERMINATE` stoppen, wird der Prozess unterbrochen. Wenn Sie dies dann tun, wurden möglicherweise einige `BIND_NOT_FIXED` -Nachrichten möglicherweise bereits einem anderen Kanal zugeordnet, oder die Nachrichten sind möglicherweise nicht in der Reihenfolge.

**Anmerkung:** z/OS

1. Bevor Sie einen Cluster einrichten, der mehrere Instanzen derselben Warteschlange enthält, müssen Sie sicherstellen, dass Ihre Nachrichten nicht über Abhängigkeiten von einander verfügen. Beispiel: Sie müssen in einer bestimmten Sequenz oder vom selben Warteschlangenmanager verarbeitet werden.
2. Erstellen Sie die Definitionen für verschiedene Instanzen derselben Warteschlange identisch. Andernfalls erhalten Sie unterschiedliche Ergebnisse von verschiedenen `MQINQ` -Aufrufen.

**Zugehörige Konzepte**

Anwendungsprogrammierung und Cluster

Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Zwei Netze in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält
Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

## *Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält*

Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie im Abschnitt Neuen WS-Manager zu einem Cluster hinzufügen beschrieben konfiguriert. Sie enthält drei Warteschlangenmanager: LONDON und NEWYORK beide enthalten vollständige Repositorys, PARIS enthält ein Teilrepository. Die Bestandsanwendung wird auf dem System in New York ausgeführt und ist mit dem NEWYORK -Warteschlangenmanager verbunden. Die Anwendung wird durch den Eingang von Nachrichten in der INVENTQ -Warteschlange gesteuert.
- Wir möchten eine Instanz von INVENTQ hinzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungssystems in Paris und New York bereitzustellen.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager hinzuzufügen, der eine lokale Warteschlange enthält.

### Vorgehensweise

1. Ändern Sie den PARIS -Warteschlangenmanager.

Für die Anwendung in Paris, um die INVENTQ in Paris und die in New York zu verwenden, müssen wir den Warteschlangenmanager informieren. Geben Sie unter PARIS den folgenden Befehl ein:

```
ALTER QMGR CLWLUSEQ(ANY)
```

2. Überprüfen Sie die Bestandsanwendung auf Nachrichtenaffinitäten.

   Bevor Sie fortfahren, stellen Sie sicher, dass die Inventaranwendung keine Abhängigkeiten von der Reihenfolge der Verarbeitung von Nachrichten hat. Weitere Informationen hierzu finden Sie im Abschnitt Nachrichtenaffinitäten bearbeiten .
3. Installieren Sie die Inventaranwendung auf dem System in Paris.
4. Definieren Sie die Clusterwarteschlange INVENTQ.

   Die INVENTQ -Warteschlange, die bereits vom NEWYORK -Warteschlangenmanager gehostet wird, wird auch von PARIS gehostet. Definieren Sie sie auf dem PARIS -Warteschlangenmanager wie folgt:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

   Nachdem Sie alle Definitionen erstellt haben, starten Sie nun in IBM MQ for z/OS den Kanalinitiator, sofern dies noch nicht geschehen ist. Starten Sie auf allen Plattformen ein Listenerprogramm auf dem Warteschlangenmanager PARIS. Der Listener ist für eingehende Netzanforderungen empfangsbereit und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In Abbildung 60 auf Seite 425 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

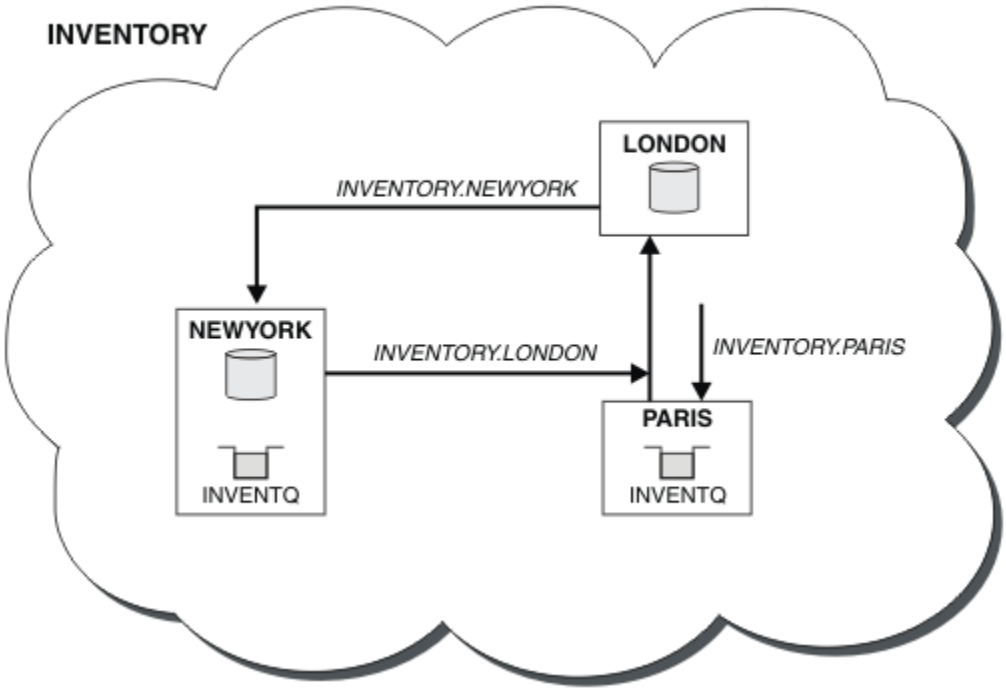



*Abbildung 60. Der INVENTORY -Cluster mit drei Warteschlangenmanagern*

Die Änderungen an diesem Cluster wurden ohne Änderung der Warteschlangenmanager NEWYORK oder LONDON vorgenommen. Die vollständigen Repositorys in diesen Warteschlangenmanagern werden automatisch mit den Informationen aktualisiert, die sie benötigen, um Nachrichten an INVENTQ in PARIS senden zu können.

## Nächste Schritte

Die INVENTQ -Warteschlange und die Inventaranwendung werden jetzt auf zwei Warteschlangenmanagern im Cluster gehostet. Dies erhöht die Verfügbarkeit, beschleunigt den Durchsatz von Nachrichten und ermöglicht die Verteilung der Auslastung zwischen den beiden Warteschlangenmanagern. Nachrichten, die von einem der WS-Manager LONDON, NEWYORK, PARIS an INVENTQ gesendet werden, werden abwechselnd an PARIS oder NEWYORK weitergeleitet, so dass die Auslastung ausgeglichen ist.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Anwendungsprogrammierung und Cluster
Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Zwei Netze in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält
Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

## *Zwei Netze in einem Cluster verwenden*

Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

## Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie im Abschnitt "WS-Manager zu einem Cluster hinzufügen" beschrieben konfiguriert. Es enthält drei Warteschlangenmanager; LONDON und NEWYORK enthalten beide vollständige Repositorys, PARIS enthält ein Teilrepository. Die Bestandsanwendung wird auf dem System

in New York ausgeführt und ist mit dem NEWYORK -Warteschlangenmanager verbunden. Die Anwendung wird durch den Eingang von Nachrichten in der INVENTQ -Warteschlange gesteuert.

- Ein neues Geschäft wird in TOKYO hinzugefügt, wo es zwei verschiedene Netzwerke gibt. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um zwei Netze in einem Cluster zu verwenden.

## Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository TOKYO auf das erste Element verweist.

   Jeder Warteschlangenmanager in einem Cluster muss sich auf einen oder einen der vollständigen Repositorys beziehen, um Informationen zum Cluster zu erfassen. Es baut sein eigenes Teilrepository auf. Es ist nicht besonders wichtig, welches Repository Sie auswählen. In diesem Beispiel wird NEWYORK ausgewählt. Sobald der neue WS-Manager dem Cluster beigetreten ist, kommuniziert er mit beiden Repositorys.
2. Definieren Sie die CLUSRCVR -Kanäle.

   Jeder Warteschlangenmanager in einem Cluster muss einen Clusterempfänger definieren, auf dem er Nachrichten empfangen kann. Dieser WS-Manager muss in jedem Netz kommunizieren können.

   ```
   DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver channel using network B for TOKYO')
   ```

   ```
   DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver channel using network A for TOKYO')
   ```
3. Definieren Sie einen CLUSSDR -Kanal auf dem Warteschlangenmanager TOKYO..

   Jeder Warteschlangenmanager in einem Cluster muss einen Clustersenderkanal definieren, auf dem er Nachrichten an sein erstes vollständiges Repository senden kann. In diesem Fall haben wir NEWYORK ausgewählt, daher benötigt TOKYO die folgende Definition:

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender channel from TOKYO to repository at NEWYORK')
   ```

   Nachdem Sie alle Definitionen erstellt haben, starten Sie nun unter IBM MQ for z/OS den Kanalinitiator, sofern dies noch nicht geschehen ist. Starten Sie auf allen Plattformen ein Listenerprogramm auf dem Warteschlangenmanager PARIS. Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In Abbildung 61 auf Seite 428 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

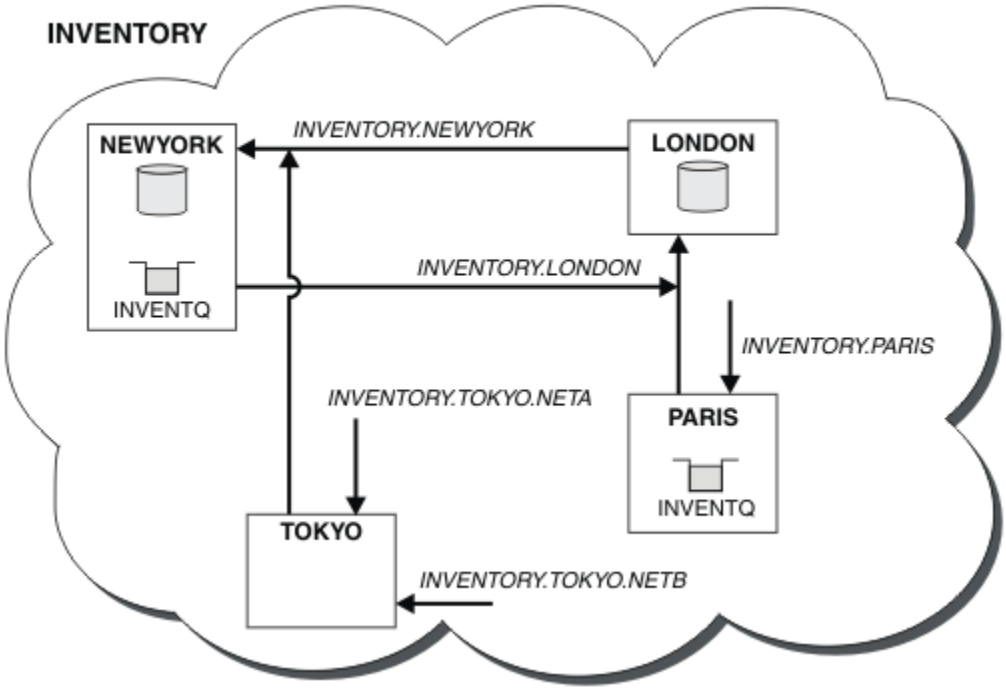



*Abbildung 61. Der INVENTORY -Cluster mit vier Warteschlangenmanagern*

Wenn Sie nur drei Definitionen vornehmen, haben wir den WS-Manager T0KY0 dem Cluster mit zwei verschiedenen Netzrouten hinzugefügt.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Anwendungsprogrammierung und Cluster
Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält

Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

„WS-Manager zu einem Cluster hinzufügen" auf Seite 339
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung der einzigen Clusterübertragungswarteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE übertragen.

### *Primäres und sekundäres Netz in einem Cluster verwenden*

Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

## Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY-Cluster wurde wie in „Zwei Netze in einem Cluster verwenden" auf Seite 426 beschrieben eingerichtet. Sie enthält vier WS-Manager; LONDON und NEWYORK enthalten vollständige Repositorys; PARIS und TOKYO enthalten Teilrepositorys. Die Bestandsanwendung wird auf dem System in New York ausgeführt und ist mit dem Warteschlangenmanager NEWYORK verbunden. Der WS-Manager von TOKYO verfügt über zwei verschiedene Netze, auf denen er kommunizieren kann.
- Sie möchten eines der Netze das primäre Netz und ein anderes der Netze das Ausweichnetz bilden. Sie planen, das Ausweichnetz zu verwenden, wenn ein Problem mit dem primären Netz besteht.

## Informationen zu diesem Vorgang

Verwenden Sie das Attribut NETPRTY , um ein primäres und ein sekundäres Netz in einem Cluster zu konfigurieren.

## Vorgehensweise

Ändern Sie die vorhandenen CLUSRCVR -Kanäle in TOKYO.

Verwenden Sie die folgenden Befehle, um anzugeben, dass das Netz ein Kanal der primäre Kanal und der Kanal des Netzes B der sekundäre Kanal ist:

a) `ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')`

b) `ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1) DESCR('Backup cluster-receiver channel for TOKYO')`

## Nächste Schritte

Wenn Sie den Kanal mit unterschiedlichen Netzprioritäten konfigurieren, haben Sie jetzt den Cluster definiert, dass Sie über ein primäres Netz und ein sekundäres Netz verfügen. Die Warteschlangenmanager im Cluster, die diese Kanäle verwenden, verwenden automatisch das primäre Netz, wenn es verfügbar ist. Die WS-Manager-Funktionsübernahme (Failover) für die Verwendung des sekundären Netzes, wenn das primäre Netz nicht verfügbar ist.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Anwendungsprogrammierung und Cluster
Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Zwei Netze in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält
Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

## *Warteschlange hinzufügen, die als Sicherung dienen soll*

Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY-Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen" auf Seite 339 beschrieben eingerichtet. Sie enthält drei Warteschlangenmanager: LONDON und NEWYORK beide enthalten vollständige Repositorys, PARIS enthält ein Teilrepository. Die Bestandsanwendung wird auf dem System in New York ausgeführt und ist mit dem NEWYORK -Warteschlangenmanager verbunden. Die Anwendung wird durch den Eingang von Nachrichten in der INVENTQ -Warteschlange gesteuert.
- Es wird ein neues Geschäft in Chicago eingerichtet, um eine Sicherung für das Inventarsystem bereitzustellen, das jetzt in New York läuft. Das Chicago-System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um eine Warteschlange hinzuzufügen, die als Sicherung dienen soll.

### Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository CHICAGO auf das erste Element verweist.

   Jeder Warteschlangenmanager in einem Cluster muss sich auf einen oder einen der vollständigen Repositorys beziehen, um Informationen zum Cluster zu erfassen. Es baut sein eigenes Teilrepository

auf. Es ist nicht besonders wichtig, welches Repository Sie für einen bestimmten Warteschlangenmanager auswählen. In diesem Beispiel wird NEWYORK ausgewählt. Sobald der neue WS-Manager dem Cluster beigetreten ist, kommuniziert er mit beiden Repositorys.

2. Definieren Sie den Kanal CLUSRCVR .

   Jeder Warteschlangenmanager in einem Cluster muss einen Clusterempfänger definieren, auf dem er Nachrichten empfangen kann. Definieren Sie unter CHICAGO Folgendes:

   ```
   DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNA-
   ME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver channel
   for CHICAGO')
   ```

3. Definieren Sie einen CLUSSDR -Kanal auf WS-Manager CHICAGO.

   Jeder Warteschlangenmanager in einem Cluster muss einen Clustersenderkanal definieren, auf dem er Nachrichten an sein erstes vollständiges Repository senden kann. In diesem Fall haben wir NEWYORK ausgewählt, daher benötigt CHICAGO die folgende Definition:

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME(NEW-
   YORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender channel from CHI-
   CAGO to repository at NEWYORK')
   ```

4. Ändern Sie die vorhandene Clusterwarteschlange INVENTQ.

   Das INVENTQ , das bereits vom NEWYORK -Warteschlangenmanager gehostet wird, ist die Hauptinstanz der Warteschlange.

   ```
   ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
   ```

5. Überprüfen Sie die Bestandsanwendung auf Nachrichtenaffinitäten.

   Bevor Sie fortfahren, stellen Sie sicher, dass die Inventaranwendung keine Abhängigkeiten von der Reihenfolge der Verarbeitung von Nachrichten hat.

6. Installieren Sie die Bestandsanwendung auf dem System in CHICAGO.
7. Definieren Sie die Sicherungs-Cluster-Warteschlange INVENTQ

   Der INVENTQ , der bereits vom NEWYORK -Warteschlangenmanager gehostet wird, wird auch als Sicherung von CHICAGO gehostet. Definieren Sie sie auf dem CHICAGO -Warteschlangenmanager wie folgt:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
   ```

   Nachdem Sie alle Definitionen erstellt haben, starten Sie nun unter IBM MQ for z/OS den Kanalinitiator, sofern dies noch nicht geschehen ist. Starten Sie auf allen Plattformen ein Listenerprogramm auf dem Warteschlangenmanager CHICAGO. Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In Abbildung 62 auf Seite 432 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

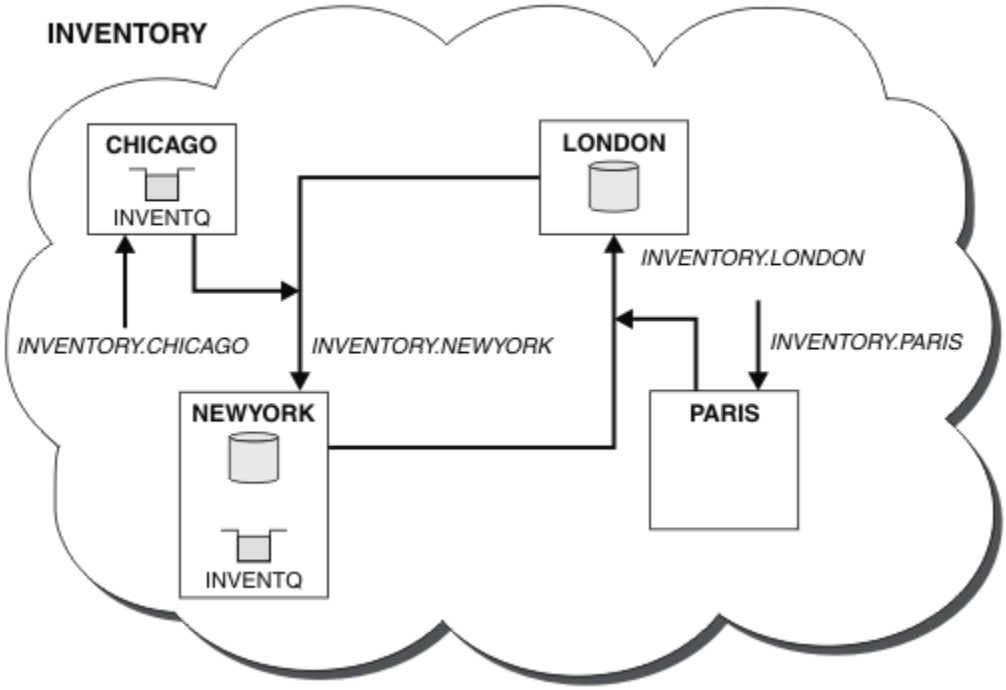



*Abbildung 62. Cluster BESTANDSFÜHRUNG mit vier Warteschlangenmanagern*

Die INVENTQ -Warteschlange und die Inventaranwendung werden jetzt auf zwei Warteschlangenmanagern im Cluster gehostet. Der WS-Manager von CHICAGO ist eine Sicherung. Nachrichten, die an INVENTQ gesendet werden, werden an NEWYORK weitergeleitet, es sei denn, sie sind nicht verfügbar, wenn sie statt CHICAGO gesendet werden.

**Anmerkung:**

Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihre Statusänderung mehrmals, wobei einige der Zustände weniger bevorzugt für den Algorithmus für die Clusterauslastung sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherung) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.

Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherungsziel gesendet werden, verwenden Sie CLWLPRTY nicht. Ziehen Sie die Verwendung separater Warteschlangen in Betracht, oder CLWLRANK mit einem manuellen Umschalten von der primären auf die Sicherung.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Anwendungsprogrammierung und Cluster
Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Zwei Netze in einem Cluster verwenden

Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält
Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

## *Einschränkung der Anzahl verwender Kanäle*

Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Es soll eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert werden. Um die Anzahl der Kanäle, die für eine niedrige Zahl verwendet werden, zu halten, ist die Anzahl der aktiven Kanäle, die jeder Server ausführt, eingeschränkt. Die Anwendung wird durch den Eingang von Nachrichten in der PRICEQ -Warteschlange gesteuert.
- Vier Server-WS-Manager hosten die Anwendung "Preisprüfung". Zwei Abfragenwarteschlangenmanager senden Nachrichten an den PRICEQ , um einen Preis abzufragen. Zwei weitere WS-Manager werden als vollständige Repositorys konfiguriert.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um die Anzahl der verwendeten Kanäle zu beschränken.

### Vorgehensweise

1. Wählen Sie zwei vollständige Repositorys aus.

   Wählen Sie zwei Warteschlangenmanager aus, die die vollständigen Repositorys für Ihren Preiscluster-Cluster sein sollen. Sie werden als REPOS1 und REPOS2 bezeichnet.

   Geben Sie den folgenden Befehl ein:

   ```
   ALTER QMGR REPOS(PRICECHECK)
   ```
2. Definieren Sie einen CLUSRCVR -Kanal auf jedem Warteschlangenmanager.

   Definieren Sie in jedem WS-Manager im Cluster einen Clusterempfängerkanal und einen Clustersenderkanal. Es spielt keine Rolle, die zuerst definiert wird.

   ```
   DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNA-
   ME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
   ```
3. Definieren Sie auf jedem WS-Manager einen CLUSSDR -Kanal.

   Erstellen Sie in jedem WS-Manager eine CLUSSDR -Definition, um diesen Warteschlangenmanager mit einem oder einem der vollständigen WS-Manager-Repositorys zu verbinden.

```
DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNA-
ME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to reposi-
tory queue manager')
```

4. Installieren Sie die Anwendung "Preisprüfung".
5. Definieren Sie die PRICEQ -Warteschlange auf allen WS-Managern des Servers.

   Geben Sie den folgenden Befehl für jede ein:

   ```
   DEFINE QLOCAL(PRICEQ) CLUSTER(PRICECHECK)
   ```
6. Anzahl der Kanäle, die von Abfragen verwendet werden, strikt

   Auf den WS-Managern der Abfrage beschränken wir die Anzahl der verwendeten aktiven Kanäle, indem Sie die folgenden Befehle für jede der folgenden Befehle ausgeben:

   ```
   ALTER QMGR CLWLMRUC(2)
   ```
7. Wenn Sie dies noch nicht getan haben, starten Sie den Kanalinitiator unter IBM MQ for z/OS. Starten Sie auf allen Plattformen ein Empfangsprogramm.

   Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In Abbildung 63 auf Seite 434 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

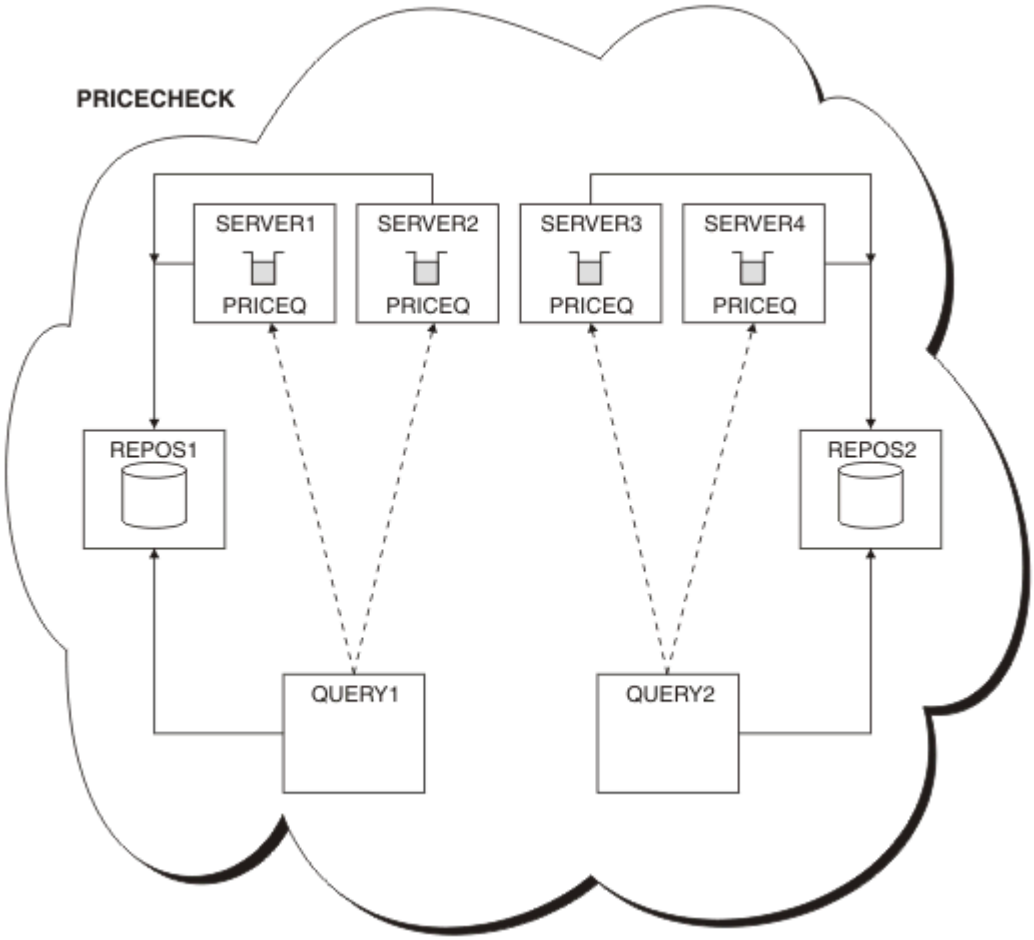



*Abbildung 63. Der PRICECHECK -Cluster mit vier Server-WS-Managern, zwei Repositorys und zwei Abfragewarteschlangenmanagern*

Obwohl es vier Instanzen der PRICEQ -Warteschlange im PRICECHECK -Cluster gibt, verwendet jeder Abfragerwarteschlangenmanager nur zwei von zwei davon. Der WS-Manager von QUERY1 verfügt beispielsweise nur über aktive Kanäle zu den WS-Managern SERVER1 und SERVER2 . Wenn SERVER1 nicht mehr verfügbar ist, beginnt der QUERY1 -Warteschlangenmanager anschließend mit der Verwendung eines anderen Warteschlangenmanagers, z. B. SERVER3.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Anwendungsprogrammierung und Cluster
Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Zwei Netze in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält
Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

## *Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält*

Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

### Vorbereitende Schritte

**Anmerkung:** Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben eingerichtet. Es enthält drei WS-Manager: LONDON und NEWYORK enthalten vollständige Repositorys, PARIS enthält ein Teilrepository und stellt Nachrichten aus INVENTQ. Die Bestandsanwendung wird auf dem System in New York ausgeführt, das mit dem NEWYORK -Warteschlangenmanager

verbunden ist. Die Anwendung wird durch den Eingang von Nachrichten in der INVENTQ -Warteschlange gesteuert.

- Es wird ein neues Geschäft in Los Angeles eingerichtet. Um zusätzliche Kapazitäten zu schaffen, wollen Sie das Inventarsystem in Los Angeles und New York betreiben. Der neue WS-Manager kann doppelt so viele Nachrichten verarbeiten wie New York.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen leistungsfähigeren Warteschlangenmanager hinzuzufügen, der als Host für eine Warteschlange dient.

## Vorgehensweise

1. Entscheiden Sie, welches vollständige Repository LOSANGELES auf das erste Element verweist.
2. Jeder Warteschlangenmanager in einem Cluster muss sich auf einen oder einen der vollständigen Repositorys beziehen, um Informationen zum Cluster zu erfassen. Es baut sein eigenes Teilrepository auf. Es ist nicht besonders wichtig, welches Repository Sie auswählen. In diesem Beispiel wird NEWYORK ausgewählt. Sobald der neue WS-Manager dem Cluster beigetreten ist, kommuniziert er mit beiden Repositorys.

   ```
   DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
   CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
   ```

3. Definieren Sie den Kanal CLUSRCVR auf WS-Manager LOSANGELES.

   Jeder WS-Manager in einem Cluster muss einen Clusterempfängerkanal definieren, auf dem er Nachrichten empfangen kann. Definieren Sie unter LOSANGELES Folgendes:

   ```
   DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
   DESCR('Cluster-receiver channel for queue manager LOSANGELES')
   CLWLWGHT(2)
   ```

   Der Clusterempfängerkanal wirbt für die Verfügbarkeit des Warteschlangenmanagers, um Nachrichten von anderen Warteschlangenmanagern im Cluster INVENTORY zu empfangen. Wenn Sie CLWLWGHT auf zwei setzen, wird sichergestellt, dass der WS-Manager von Los Angeles doppelt so viele Bestandsnachrichten wie New York erhält (wenn der Kanal für NEWYORK auf einen Wert gesetzt ist).
4. Ändern Sie den Kanal CLUSRCVR auf dem Warteschlangenmanager NEWYORK.

   Stellen Sie sicher, dass der WS-Manager von Los Angeles doppelt so viele Inventarnachrichten wie New York erhält. Ändern Sie die Definition des Clusterempfängerkanals.

   ```
   ALTER CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) CLWLWGHT(1)
   ```

5. Überprüfen Sie die Bestandsanwendung auf Nachrichtenaffinitäten.

   Bevor Sie fortfahren, stellen Sie sicher, dass die Inventaranwendung keine Abhängigkeiten von der Reihenfolge der Verarbeitung von Nachrichten hat.
6. Installieren Sie die Inventaranwendung auf dem System in Los Angeles.
7. Definieren Sie die Clusterwarteschlange INVENTQ.

   Die INVENTQ -Warteschlange, die bereits vom NEWYORK -Warteschlangenmanager gehostet wird, befindet sich ebenfalls in LOSANGELES. Definieren Sie sie auf dem LOSANGELES -Warteschlangenmanager wie folgt:

   ```
   DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
   ```

   Nachdem Sie alle Definitionen erstellt haben, starten Sie nun unter IBM MQ for z/OS den Kanalinitiator, sofern dies noch nicht geschehen ist. Starten Sie auf allen Plattformen ein Listenerprogramm auf

dem Warteschlangenmanager LOSANGELES. Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In „Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält” auf Seite 435 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

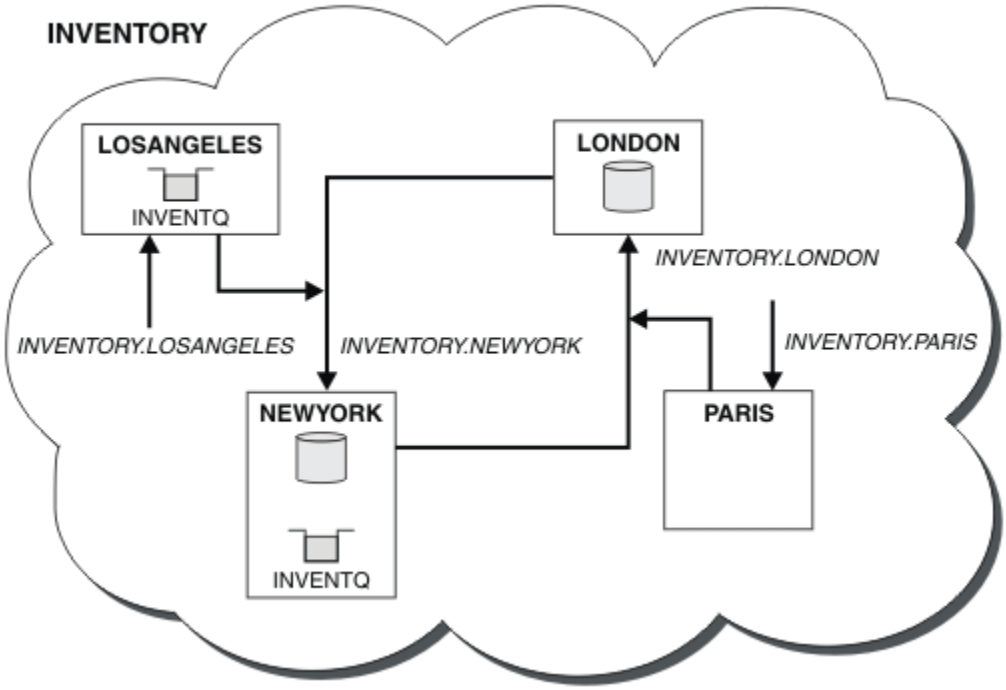



*Abbildung 64. Der INVENTORY -Cluster mit vier Warteschlangenmanagern*

Diese Änderung am Cluster wurde ausgeführt, ohne dass die Warteschlangenmanager LONDON und PARIS geändert werden müssen. Die Repositorys in diesen Warteschlangenmanagern werden automatisch mit den Informationen aktualisiert, die sie benötigen, um Nachrichten an INVENTQ in LOSANGELES senden zu können.

## Nächste Schritte

Die INVENTQ -Warteschlange und die Inventaranwendung werden auf zwei Warteschlangenmanagern im Cluster gehostet. Die Konfiguration erhöht die Verfügbarkeit, beschleunigt den Durchsatz von Nachrichten und ermöglicht die Verteilung der Auslastung zwischen den beiden Warteschlangenmanagern. Nachrichten, die entweder von LOSANGELES oder NEWYORK an INVENTQ gestellt werden, werden, wenn möglich, von der Instanz auf dem lokalen WS-Manager bearbeitet. Nachrichten, die von LONDON oder PARIS gestellt werden, werden an LOSANGELES oder NEWYORK weitergeleitet, wobei doppelt so viele Nachrichten an LOSANGELES gesendet werden.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

Anwendungsprogrammierung und Cluster

Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von INVENTQ hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Zwei Netze in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in TOKYO hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

### *Anwendungsprogrammierung und Cluster*

Sie müssen keine Programmieränderungen vornehmen, um die Vorteile mehrerer Instanzen derselben Warteschlange nutzen zu können. Einige Programme funktionieren jedoch nicht korrekt, es sei denn, eine Folge von Nachrichten wird an dieselbe Instanz einer Warteschlange gesendet.

Anwendungen können eine Warteschlange mit dem Aufruf MQOPEN öffnen. Anwendungen verwenden den Aufruf MQPUT , um Nachrichten in eine geöffnete Warteschlange einzureihen. Anwendungen können eine einzelne Nachricht mit dem Aufruf MQPUT1 in eine Warteschlange stellen, die noch nicht geöffnet ist.

Wenn Sie Cluster konfigurieren, die über mehrere Instanzen derselben Warteschlange verfügen, gibt es keine spezifischen Hinweise zur Anwendungsprogrammierung. Wenn Sie jedoch die Aspekte des Workloadmanagements im Clustering nutzen möchten, müssen Sie möglicherweise Ihre Anwendungen ändern. Wenn Sie ein Netz einrichten, in dem mehrere Definitionen derselben Warteschlange vorhanden sind, überprüfen Sie Ihre Anwendungen auf Nachrichtenaffinitäten.

Angenommen, Sie haben zwei Anwendungen, die sich auf eine Reihe von Nachrichten stützen, die zwischen ihnen in Form von Fragen und Antworten fließen. Sie möchten wahrscheinlich, dass Antworten auf denselben WS-Manager zurückgehen, der eine Frage gesendet hat. Es ist wichtig, dass die Workload-Management-Routine die Nachrichten nicht an einen WS-Manager sendet, der eine Kopie der Antwortwarteschlange enthält.

Es können Anwendungen vorhanden sein, für die Nachrichten in der Reihenfolge verarbeitet werden müssen (z. B. eine Datenbankanwendungsanwendung, die Stapel von Nachrichten sendet, die in der Sequenz abgerufen werden müssen). Die Verwendung von segmentierten Nachrichten kann auch zu einem Affinitätsproblem führen.

## Lokale oder ferne Version der Zielwarteschlange öffnen

Achten Sie darauf, wie der Warteschlangenmanager auswählt, ob eine lokale oder eine ferne Version der Zielwarteschlange verwendet wird.

1. Der Warteschlangenmanager öffnet die lokale Version der Zielwarteschlange, um Nachrichten zu lesen, oder um die Attribute der Warteschlange festzulegen.

2. Der Warteschlangenmanager öffnet eine beliebige Instanz der Zielwarteschlange, in die Nachrichten geschrieben werden sollen, wenn mindestens eine der folgenden Bedingungen zutrifft:
   - Es ist keine lokale Version der Zielwarteschlange vorhanden.
   - Der WS-Manager gibt `CLWLUSEQ(ANY)` auf `ALTER QMGR` an.
   - Die Warteschlange auf dem WS-Manager gibt `CLWLUSEQ(ANY)` an.

**Zugehörige Konzepte**

Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange
In diesem Beispiel für einen Cluster mit mehr als einer Instanz einer Warteschlange werden Nachrichten an verschiedene Instanzen der Warteschlange weitergeleitet. Sie können eine Nachricht zu einer bestimmten Instanz der Warteschlange erzwingen, und Sie können auswählen, dass eine Nachrichtenfolge an einen der Warteschlangenmanager gesendet werden soll.

**Zugehörige Tasks**

Hinzufügen eines Warteschlangenmanagers, der eine lokale Warteschlange enthält
Führen Sie die folgenden Anweisungen aus, um eine Instanz von `INVENTQ` hinzuzufügen, um zusätzliche Kapazität für die Ausführung des Inventaranwendungs-Systems in Paris und New York bereitzustellen.

Zwei Netze in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um einen neuen Speicher in `TOKYO` hinzuzufügen, in dem sich zwei verschiedene Netze befinden. Beide müssen für die Kommunikation mit dem WS-Manager in Tokio verfügbar sein.

Primäres und sekundäres Netz in einem Cluster verwenden
Führen Sie die folgenden Anweisungen aus, um ein Netz zum primären Netz zu machen, und ein anderes Netz das Ausweichnetz zu erstellen. Verwenden Sie das Ausweichnetz, wenn ein Problem mit dem primären Netz besteht.

Warteschlange hinzufügen, die als Sicherung dienen soll
Folgen Sie diesen Anweisungen, um eine Sicherung in Chicago für das Inventarsystem bereitzustellen, das jetzt in New York ausgeführt wird. Das Chicagoer System wird nur verwendet, wenn es ein Problem mit dem New Yorker System gibt.

Einschränkung der Anzahl verwender Kanäle
Befolgen Sie diese Anweisungen, um die Anzahl der aktiven Kanäle zu beschränken, die jeder Server ausführt, wenn eine Preisprüfung auf verschiedenen Warteschlangenmanagern installiert ist.

Hinzufügen eines leistungsfähigeren Warteschlangenmanagers, der eine Warteschlange enthält
Befolgen Sie diese Anweisungen, um zusätzliche Kapazität bereitzustellen, indem Sie das Bestandssystem in Los Angeles sowie New York ausführen, wo Los Angeles die doppelte Anzahl von Nachrichten als New York verarbeiten kann.

*Nachrichtenaffinitäten bearbeiten*
Nachrichtenaffinitäten sind selten Teil eines guten Programmierdesigns. Sie müssen Nachrichtenaffinitäten entfernen, um das Clustering vollständig zu verwenden. Wenn Sie keine Nachrichtenaffinitäten entfernen können, können Sie erzwingen, dass zugehörige Nachrichten mit demselben Kanal und mit demselben Warteschlangenmanager zugestellt werden.

Wenn Sie Anwendungen mit Nachrichtenaffinitäten haben, entfernen Sie die Affinitäten, bevor Sie mit der Verwendung von Clustern beginnen.

Durch das Entfernen von Nachrichtenaffinitäten wird die Verfügbarkeit von Anwendungen verbessert. Eine Anwendung sendet einen Stapel von Nachrichten, die Nachrichtenaffinitäten an einen Warteschlangenmanager senden. Der Warteschlangenmanager schlägt fehl, nachdem er nur einen Teil der Stapelverarbeitung empfangen hat. Der sendende Warteschlangenmanager muss warten, bis er die unvollständige Nachrichtenstapelverarbeitung wiederhergestellt und verarbeitet hat, bevor er weitere Nachrichten senden kann.

Durch das Entfernen von Nachrichtenaffinitäten wird auch die Skalierbarkeit von Anwendungen verbessert. Ein Stapel von Nachrichten mit Affinitäten kann Ressourcen auf dem Zielwarteschlangenmanager sperren, während er auf nachfolgende Nachrichten wartet. Diese Ressourcen bleiben möglicherweise lange Zeit gesperrt, so dass andere Anwendungen ihre Arbeit nicht ausführen können.

Darüber hinaus verhindern Nachrichtenaffinitäten, dass die Cluster-Workload-Management-Routinen die beste Auswahl an WS-Managern treffen.

Wenn Sie Affinitäten entfernen möchten, sollten Sie die folgenden Möglichkeiten berücksichtigen:

- Statusinformationen in den Nachrichten eintragen
- Verwaltung von Statusinformation in einem für alle Warteschlangenmanager zugänglichen nicht flüchtigen Speicher, z. B. in einer Db2-Datenbank
- Replizieren von schreibgeschützten Daten, so dass auf mehrere WS-Manager zugegriffen werden kann

Wenn es nicht sinnvoll ist, Ihre Anwendungen so zu ändern, dass Nachrichtenaffinitäten entfernt werden, gibt es eine Reihe möglicher Lösungen für das Problem.

### Ein bestimmtes Ziel im MQOPEN -Aufruf benennen

Geben Sie den Namen der fernen Warteschlange und den Namen des Warteschlangenmanagers in jedem MQOPEN -Aufruf an. Alle Nachrichten, die unter Verwendung dieser Objektkennung in die Warteschlange eingereiht werden, gehen an denselben Warteschlangenmanager, bei dem es sich möglicherweise um den lokalen Warteschlangenmanager handelt.

Die Angabe des Namens der fernen Warteschlange und des Warteschlangenmanagernamens in jedem MQOPEN -Aufruf hat Nachteile:

- Es wird kein Lastausgleich durchgeführt. Sie profitieren nicht von den Vorteilen des Lastausgleichs in der Clusterauslastung.
- Wenn der Zielwarteschlangenmanager fern ist und mehr als ein Kanal zu ihm vorhanden ist, können die Nachrichten unterschiedliche Routen verwenden, und die Nachrichtenfolge wird immer noch nicht beibehalten.
- Wenn Ihr Warteschlangenmanager über eine Definition für eine Übertragungswarteschlange mit demselben Namen wie der Zielwarteschlangenmanager verfügt, werden Nachrichten in diese Übertragungswarteschlange und nicht in die Clusterübertragungswarteschlange übertragen.

### Geben Sie den Namen des Warteschlangenmanagers im Feld für den Antwortwarteschlangenmanager zurück.

Ermöglichen Sie dem WS-Manager, der die erste Nachricht in einem Stapel empfängt, seinen Namen in seiner Antwort zurück. Dies erfolgt über das Feld ReplyToQMgr des Nachrichtendeskriptors. Der WS-Manager am sendenden Ende kann dann den Namen des Antwortwarteschlangenmanagers extrahieren und in allen nachfolgenden Nachrichten angeben.

Die Verwendung der ReplyToQMgr -Informationen aus der Antwort hat Nachteile:

- Der anfordernde Warteschlangenmanager muss auf eine Antwort auf seine erste Nachricht warten.
- Sie müssen zusätzlichen Code schreiben, um die Informationen zu ReplyToQMgr zu suchen und zu verwenden, bevor Sie nachfolgende Nachrichten senden.
- Wenn mehr als eine Route zum WS-Manager vorhanden ist, wird die Reihenfolge der Nachrichten möglicherweise nicht beibehalten.

### Legen Sie die Option MQOO_BIND_ON_OPEN im Aufruf MQOPEN fest.

Erzwingen Sie mit der Option MQOO_BIND_ON_OPEN für den Aufruf von MQOPEN, dass alle Nachrichten an dasselbe Ziel zugestellt werden sollen. Bei Verwendung von Nachrichtengruppen mit Clustern muss entweder MQOO_BIND_ON_OPEN oder MQOO_BIND_ON_GROUP angegeben werden, um sicherzustellen, dass alle Nachrichten in der Gruppe an demselben Ziel verarbeitet werden.

Wenn Sie eine Warteschlange öffnen und MQOO_BIND_ON_OPEN angeben, werden alle Nachrichten, die an diese Warteschlange gesendet werden, gezwungen, an dieselbe Instanz der Warteschlange zu senden. MQOO_BIND_ON_OPEN bindet alle Nachrichten an denselben Warteschlangenmanager und auch an denselben Leitweg. Wenn es beispielsweise eine IP-Route und eine NetBIOS-Route zum selben Ziel gibt, wird

eine dieser Routen ausgewählt, wenn die Warteschlange geöffnet wird, und diese Auswahl wird für alle Nachrichten berücksichtigt, die mit der Objektkennung in dieselbe Warteschlange gestellt werden.

Durch Angabe von MQOO_BIND_ON_OPEN erzwingen Sie, dass alle Nachrichten an dasselbe Ziel weitergeleitet werden. Daher werden Anwendungen mit Nachrichtenaffinitäten nicht unterbrochen. Wenn die Zieladresse nicht verfügbar ist, bleiben die Nachrichten in der Übertragungswarteschlange, bis sie wieder verfügbar wird.

MQOO_BIND_ON_OPEN gilt auch, wenn der Name des Warteschlangenmanagers beim Öffnen einer Warteschlange im Objektdeskriptor angegeben wird. Es kann mehr als eine Route zum angegebenen WS-Manager geben. Es kann z. B. mehrere Netzpfade geben, oder ein anderer WS-Manager hat möglicherweise einen Aliasnamen definiert. Wenn Sie MQOO_BIND_ON_OPEN angeben, wird eine Route ausgewählt, wenn die Warteschlange geöffnet wird.

**Anmerkung:** Dies ist das empfohlene Verfahren. Es funktioniert jedoch nicht in einer Multi-Hop-Konfiguration, in der ein WS-Manager einen Aliasnamen für eine Clusterwarteschlange ausweist. Es hilft auch nicht in Situationen, in denen Anwendungen unterschiedliche Warteschlangen in demselben Warteschlangenmanager für verschiedene Nachrichtengruppen verwenden.

Eine Alternative zur Angabe von MQOO_BIND_ON_OPEN im Aufruf MQOPEN ist die Änderung Ihrer Warteschlangendefinitionen. Geben Sie in Ihren Warteschlangendefinitionen DEFBIND(OPEN) an und lassen Sie die Option DefBind im Aufruf MQOPEN standardmäßig auf MQOO_BIND_AS_Q_DEFzu.

## Legen Sie die Option MQOO_BIND_ON_GROUP im Aufruf MQOPEN fest.

Erzwingen Sie mithilfe der Option MQOO_BIND_ON_GROUP für den Aufruf von MQOPEN, dass alle Nachrichten in einer Gruppe an dasselbe Ziel zugestellt werden sollen. Bei Verwendung von Nachrichtengruppen mit Clustern muss entweder MQOO_BIND_ON_OPEN oder MQOO_BIND_ON_GROUP angegeben werden, um sicherzustellen, dass alle Nachrichten in der Gruppe an demselben Ziel verarbeitet werden.

Wenn Sie eine Warteschlange öffnen und MQOO_BIND_ON_GROUP angeben, erzwingen Sie alle Nachrichten in einer Gruppe, die an diese Warteschlange gesendet werden, um an dieselbe Instanz der Warteschlange gesendet zu werden. MQOO_BIND_ON_GROUP bindet alle Nachrichten in einer Gruppe an denselben Warteschlangenmanager und auch an denselben Leitweg. Wenn es beispielsweise eine IP-Route und eine NetBIOS-Route zum selben Ziel gibt, wird eine dieser Routen ausgewählt, wenn die Warteschlange geöffnet wird und diese Auswahl für alle Nachrichten in einer Gruppe berücksichtigt wird, die mit Hilfe der Objektkennung in dieselbe Warteschlange gestellt wird.

Wenn Sie MQOO_BIND_ON_GROUP angeben, erzwingen Sie alle Nachrichten in einer Gruppe, die an dasselbe Ziel weitergeleitet werden sollen. Daher werden Anwendungen mit Nachrichtenaffinitäten nicht unterbrochen. Wenn die Zieladresse nicht verfügbar ist, bleiben die Nachrichten in der Übertragungswarteschlange, bis sie wieder verfügbar wird.

MQOO_BIND_ON_GROUP gilt auch, wenn der Name des Warteschlangenmanagers beim Öffnen einer Warteschlange im Objektdeskriptor angegeben wird. Es kann mehr als eine Route zum angegebenen WS-Manager geben. Es kann z. B. mehrere Netzpfade geben, oder ein anderer WS-Manager hat möglicherweise einen Aliasnamen definiert. Wenn Sie MQOO_BIND_ON_GROUP angeben, wird eine Route ausgewählt, wenn die Warteschlange geöffnet wird.

Damit MQOO_BIND_ON_GROUP wirksam ist, müssen Sie die PUT-Option MQPMO_LOGICAL_ORDER in MQPUT einfügen. Sie können **GroupId** im MQMD der Nachricht auf MQGI_NONE setzen und Sie müssen folgende Nachrichtenflags in das MQMD-Feld **MsgFlags** der Nachrichten einfügen:

- Letzte Nachricht in der Gruppe: MQMF_LAST_MSG_IN_GROUP
- Alle anderen Nachrichten in der Gruppe: MQMF_MSG_IN_GROUP

Wenn MQOO_BIND_ON_GROUP angegeben wird, die Nachrichten jedoch nicht gruppiert sind, entspricht das Verhalten dem Wert MQOO_BIND_NOT_FIXED .

**Anmerkung:** Dies ist das empfohlene Verfahren, mit dem sichergestellt wird, dass Nachrichten in einer Gruppe an dasselbe Ziel gesendet werden. Sie funktioniert jedoch nicht in einer Multi-Hop-Konfiguration, in der ein Warteschlangenmanager einen Aliasnamen für eine Clusterwarteschlange bewirbt.

Eine Alternative zur Angabe von MQOO_BIND_ON_GROUP im Aufruf MQOPEN ist die Änderung Ihrer Warteschlangendefinitionen. Geben Sie in Ihren Warteschlangendefinitionen DEFBIND(GROUP) an und lassen Sie die Option DefBind im Aufruf MQOPEN standardmäßig auf MQOO_BIND_AS_Q_DEFzu.

### Angepasstes Exitprogramm für Clusterauslastung schreiben

Anstatt Ihre Anwendungen zu ändern, können Sie das Problem der Nachrichtenaffinitäten umgehen, indem Sie ein Exitprogramm für Clusterauslastung schreiben. Das Schreiben eines Exitprogramms zur Clusterauslastung ist nicht einfach und wird nicht empfohlen. Das Programm muss so gestaltet sein, dass es die Affinität erkennt, indem es den Inhalt von Nachrichten überprüft. Nachdem die Affinität erkannt wurde, muss das Programm das Dienstprogramm für die Workload-Management erzwingen, um alle zugehörigen Nachrichten an denselben Warteschlangenmanager weiterzuleiten.

## Multi Uniform-Cluster konfigurieren

Mit Uniform-Clustern können Anwendungen im Hinblick auf Umfang und Verfügbarkeit entwickelt werden und sie können eine Verbindung zu jedem Warteschlangenmanager in diesem Uniform-Cluster herstellen.

### Vorbereitende Schritte

Informationen zu einer Einführung in das Clustering finden Sie unter Cluster. Eine Einführung in Uniform-Cluster finden Sie unter „Informationen zu Uniform-Clustern” auf Seite 442.

### Informationen zu diesem Vorgang

Uniform-Cluster nutzen das IBM MQ-Clustering für die Kommunikation zwischen Warteschlangenmanagern und die Lastverteilung zwischen Warteschlangen. Sie unterscheiden sich von typischen IBM MQ-Clustern allerdings folgendermaßen:

- Uniform-Cluster enthalten normalerweise eine kleinere Anzahl von Warteschlangenmanagern im Cluster. Sie sollten keinen Uniform-Cluster mit mehr als 10 Warteschlangenmanagern erstellen.
- Jedes Mitglied des Clusters hat eine nahezu identische Konfiguration.
- Der Cluster wird in der Regel von einer einzelnen Anwendung oder einer Gruppe zusammengehöriger Anwendungen verwendet.
- Die Anzahl der Anwendungsinstanzen, die eine Verbindung zum Cluster herstellen, sollte größer oder gleich der Anzahl der Warteschlangenmanager sein.

Mit der automatischen Konfiguration und der Unterstützung für automatisches Clustering können Sie sich sowohl die Erstellung eines Uniform-Clusters als auch die Synchronisierung der Mitgliederkonfigurationen dieses Clusters vereinfachen.

### Prozedur

- Informationen zu Uniform-Clustern
- Uniform-Cluster erstellen
- Uniform-Cluster erstellen
- Warteschlangenmanager aus einem Uniform-Cluster aussetzen

## Multi Informationen zu Uniform-Clustern

Eine Uniform-Clusterimplementierung hat das Ziel, dass Anwendungen im Hinblick auf Umfang und Verfügbarkeit entwickelt werden und eine Verbindung zu jedem Warteschlangenmanager im Uniform-Cluster herstellen können. Dadurch sind Anwendungen nicht von einem bestimmten Warteschlangenmanager abhängig, was zu einer verbesserten Verfügbarkeit und einem besseren Lastausgleich im Messaging-Verkehr führt. z/OS Unter IBM MQ for z/OS sind keine Uniform-Cluster verfügbar. Viele Funktionen eines Uniform-Clusters werden dort von Gruppen mit gemeinsamer Warteschlange übernommen.

Bei Uniform-Clustern handelt es sich um ein bestimmtes Muster eines IBM MQ-Clusters, in dem eine kleine Gruppe von Warteschlangenmanagern mit hoher Verfügbarkeit und einem horizontalen Maßstab bereitgestellt wird. Diese Warteschlangenmanager werden fast identisch konfiguriert, so dass eine Anwendung mit ihnen als einzelne Gruppe interagieren kann. Die Nutzung jedes Warteschlangenmanagers im Cluster kann somit einfacher sichergestellt werden, da die gleichmäßige Verteilung von Anwendungsinstanzen auf alle Warteschlangenmanager automatisch vorgenommen wird.

Durch Uniform-Cluster werden einige manuelle Schritte überflüssig, die ein Administrator zum Erstellen und Verwalten einer Gruppe von unabhängigen und miteinander verbundenen Warteschlangenmanagern ausführen muss. Dabei wird ein Teil der Clientverbindungslogik vom Client auf den Warteschlangenmanager verschoben, wodurch Clients mithilfe der Informationen über die Stufe der Anwendungsaktivität entscheiden, zu welchen Warteschlangenmanagern sie eine Verbindung herstellen.

Mit der automatischen Konfiguration und der Unterstützung für automatisches Clustering können Sie sich sowohl die anfängliche Einrichtung eines Uniform-Clusters als auch die Synchronisierung der Mitgliederkonfigurationen dieses Clusters vereinfachen. Wenn Sie sich dieser automatischen Funktionen bedienen, beschreibt eine Konfigurationsdatei den Cluster und eine weitere Konfigurationsdatei beschreibt die MQSC-Konfiguration für alle Warteschlangenmanager des Uniform-Clusters. Bei jedem Neustart eines Warteschlangenmanagers wird die Konfiguration erneut angewendet und der Cluster wird automatisch gebildet. Weitere Informationen zu dieser Funktion finden Sie im Abschnitt „Uniform-Cluster erstellen” auf Seite 458.

Damit Sie alle Vorteile eines Uniform-Clusters nutzen können, sollte jede Anwendung auch in mehrere übereinstimmende Instanzen skaliert werden. Es wird empfohlen, dabei mindestens so viele Instanzen zu verwenden, wie Warteschlangenmanager vorhanden sind, idealerweise sogar sehr viel mehr.

Ein IBM MQ-Cluster mit einer beliebigen Größe stellt viele Funktionen bereit:

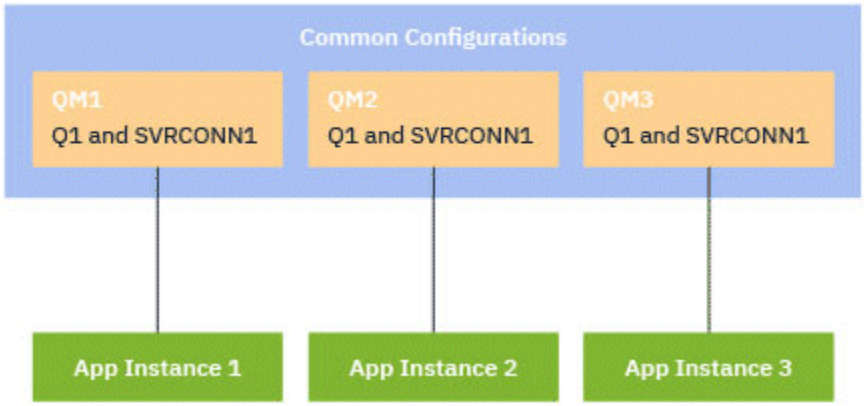
- Ein Verzeichnis aller Clusterressourcen, die von einem Mitglied in einem Cluster erkennt werden können
- Automatisches Erstellen von Kanälen und Herstellen einer Verbindung
- Horizontale Skalierung in mehreren übereinstimmenden Warteschlangen mithilfe der Lastverteilung für Nachrichten
- Dynamische Nachrichtenweiterleitung auf Basis der Verfügbarkeit

Uniform-Cluster nutzen das IBM MQ-Clustering für die Kommunikation zwischen Warteschlangenmanagern und die Lastverteilung zwischen Warteschlangen. Sie unterscheiden sich von typischen IBM MQ-Clustern allerdings folgendermaßen:

- Uniform-Cluster enthalten normalerweise eine kleinere Anzahl von Warteschlangenmanagern im Cluster. Sie sollten keinen Uniform-Cluster mit mehr als 10 Warteschlangenmanagern erstellen.
- Jedes Mitglied des Clusters hat eine nahezu identische Konfiguration.
- Der Cluster wird in der Regel von einer einzelnen Anwendung oder einer Gruppe zusammengehöriger Anwendungen verwendet.
- Die Anzahl der Anwendungsinstanzen, die eine Verbindung zum Cluster herstellen, sollte größer oder gleich der Anzahl der Warteschlangenmanager sein.

In der Struktur eines Uniform-Clusters bieten alle Warteschlangenmanager im Cluster dieselben Messaging-Services an. Beispielsweise können Sie alle Cluster-Member so konfigurieren, dass dieselben lokalen Warteschlangen definiert sind, und Clientanwendungen ermöglichen, eine Verbindung zu einem beliebigen Member des Clusters herzustellen. Sie können auch die gleichen Serververbindungskanäle definiert haben und möglicherweise die gleichen Berechtigungssätze, Kanalauthentifizierungsregeln usw. Bei den Mitgliedern des Clusters kann es aber trotzdem Unterschiede bei den Objekten und der Konfiguration geben. Beispielsweise können einige Anwendungen temporäre dynamische Warteschlangen erstellen, während sie mit einem Warteschlangenmanager verbunden sind. Außerdem können einige Konfigurationsaktualisierungen, wie z. B. neue oder aktualisierte Zertifikate, während eines bestimmten Zeitraums auf die Mitglieder ausgelagert werden. Wie bei regulären IBM MQ-Clustern erfordern zwei Warteschlangenmanager eine zusätzliche Konfiguration, um sie als vollständige Repository-Warteschlangenmanager zu erstellen.

Im folgenden Diagramm wird gezeigt, dass die Warteschlangenmanager ähnliche Konfigurationen aufweisen. Sie definieren die gleiche Warteschlange mit der Bezeichnung Q1 und den gleichen Serververbindungskanal SVRCONN1.




Hinweis: Wenn mehrere Warteschlangenmanager mit identischen Namen für den Serververbindungskanal mit einer einzelnen Clientdefinitionstabelle (CCDT) arbeiten sollen, müssen das aktualisierte CCDT-Format verwenden, das in IBM MQ 9.1.2 eingeführt wurde. Siehe „CCDT im JSON-Format konfigurieren" auf Seite 47.

## Anwendungsnamen und Anwendungsinstanzen

Ein Anwendungsname wird als Attribut APPLTAG des **DISPLAY CONN(*) TYPE CONN** -Befehls angezeigt. Seit IBM MQ 9.1.2 wird der Anwendungsname auf andere Weise festgelegt.

Bei einer Instanz einer Anwendung handelt es sich um eine Gruppe eng zusammengehöriger Verbindungen, die eine *Ausführungseinheit* für diese Anwendung bereitstellen. In der Regel handelt es sich um einen einzelnen Betriebssystemprozess, der eine Reihe von Threads und zugehörigen IBM MQ-Verbindungen aufweisen kann.

Weitere Informationen zu Anwendungsnamen und Anwendungsinstanzen finden Sie unter Anwendungsentwicklungskonzepte.

## Wiederverbindungsfähige Clients

Wiederverbindungsfähige Clients können verschoben werden, um eine gleichmäßige Lastverteilung zu erreichen, wohingegen ein nicht wiederverbindungsfähiger Client definitionsgemäß nicht mit einem anderen Warteschlangenmanager neu verbunden werden kann. Es gibt jedoch weiterhin Fälle, in denen man einen nicht wiederverbindungsfähigen Client in einen Uniform-Cluster einbinden möchte; beispielsweise wenn der Client eine Art persistenten Status erzeugt und wenn mit einem anderen Verfahren sichergestellt wird, dass in jedem der Warteschlangenmanager Instanzen der Anwendung ausgeführt werden.

## Lokal gebundene Anwendungen

In Uniform-Clustern wird erwartet, dass IBM MQ-Anwendungen anstelle von lokal gebundenen Anwendungen eine Verbindung als Clientanwendung herstellen. Lokal gebundene Anwendungen werden nicht daran gehindert, eine Verbindung zu Mitgliedern des Uniform-Clusters herzustellen, aber Uniform-Cluster erreichen keine ausgewogene Lastverteilung bei lokal gebundenen Anwendungen, da sie keine Verbindung zu einem anderen Cluster-Member herstellen können.

### Zugehörige Tasks

Anwendungsname in unterstützten Programmiersprachen angeben

## Multi *Automatischer Ausgleich von Anwendungen*

Die Verteilung und Verfügbarkeit von Anwendungen durch den automatischen Anwendungsausgleich wird erheblich verbessert, indem ein IBM MQ-Uniform-Cluster aktiviert wird, mit dem die Anwendungsvertei-

lung im gesamten Cluster genau verwaltet wird, anstatt auf eine beliebige Festlegung oder manuelle Fixierung von Anwendungen auf bestimmte Warteschlangenmanager angewiesen zu sein.

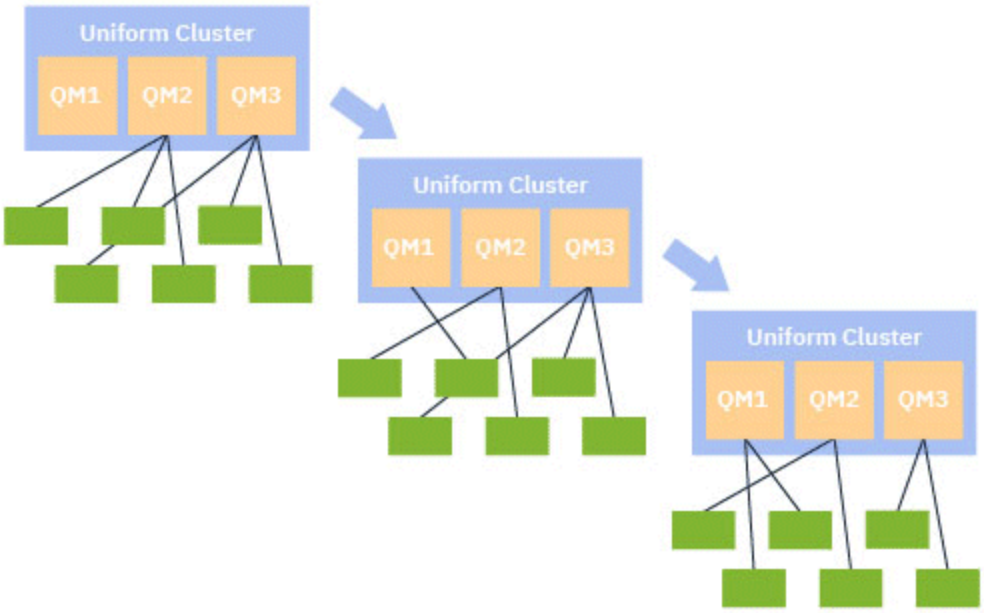
Ab IBM MQ 9.2.0 wird der automatische Ausgleich in einer Gruppe von Clusterwarteschlangenmanagern für Anwendungen unterstützt, die in C, JMS, IBM MQ .NET, XMS .NET geschrieben wurden.

Wenn mindestens so viele Instanzen der gleichen Anwendung wie Warteschlangenmanager vorhanden sind, stellt das Uniform-Cluster fortwährend sicher, dass jeder Warteschlangenmanager mit mindestens einer Instanz der Anwendung verbunden ist.

Anwendungen können eine spezifische Affinität zu einem Warteschlangenmanager verhindern und verwenden stattdessen eine Definitionstabelle für den Clientkanal (CCDT). um die Konnektivität zu der Gruppe der Warteschlangenmanager im Uniform-Cluster zuverlässig zu randomisieren. Anwendungen können dies aus den folgenden Gründen tun:

- Wenn ausreichend konsumierende Anwendungsinstanzen vorhanden sind, gibt es immer eine Instanz der Nachrichten zur Anwendungsverarbeitung.
- Wenn Sie einen Warteschlangenmanager stoppen, werden alle verbundenen Anwendungsinstanzen auf die verbleibenden Warteschlangenmanager im Cluster gleichmäßig verteilt.
- Beim Starten eines Warteschlangenmanagers werden die mit anderen Warteschlangenmanagern im Cluster verbundenen Anwendungsinstanzen automatisch erneut ausgeglichen, damit der neu gestartete Warteschlangenmanager auch integriert ist.

Dadurch stellt der Uniform-Cluster fortlaufend sicher, dass Anwendungen optimal verteilt sind und die Nachrichtenverarbeitung auch im Falle geplanter oder ungeplanter Betriebsunterbrechungen maximiert ist.




Um den automatischen Lastausgleich zu erreichen, teilen sich die Warteschlangenmanager im Uniform-Cluster in regelmäßigen Abständen Informationen untereinander. Sie tun dies, indem sie Metadaten zu Systemthemen unter der reservierten $SYS/MQ-Verzweigung der Themenstruktur veröffentlichen. Jeder Warteschlangenmanager im Uniform-Cluster subskribiert Nachrichten, die von anderen Warteschlangenmanagern veröffentlicht werden, und erstellt ein Bild vom Status der Anwendungen im Uniform-Cluster.

Die Warteschlangenmanager überwachen die Verteilung von Clientanwendungen im gesamten Cluster. Wenn die Anzahl der Anwendungen, die mit einem bestimmten Warteschlangenmanager verbunden sind, so niedrig ist, dass der Cluster als nicht ausgeglichen festgelegt wird, veröffentlicht der Warteschlangenmanager eine Anforderung an ein Systemthema an einen der anderen Warteschlangenmanager im Cluster.

Wenn die Nachricht empfangen wird, fordert der Zielwarteschlangenmanager von einem der zugehörigen Clientanwendungen die Weiterleitung an den anfordernden Warteschlangenmanager an. Die Clientanwendung empfängt die Anforderung zur Weiterleitung, trennt die Verbindung und stellt eine Verbindung zum anfordernden Warteschlangenmanager her. Dieses automatische Verfahren für einen Lastausgleich ist für die Anwendung transparent. Weitere Informationen finden Sie unter „Funktionsweise des automatischen Ausgleichs” auf Seite 446.

Durch die regelmäßige Verteilung von Metadaten auf die verbundenen Anwendungen kann der Uniform-Cluster erreichen, dass das Verhältnis von Clientanwendungen zu Warteschlangenmanagern mit der Zeit ausgeglichen ist. Um zu verhindern, dass Ereignisse zur Weiterverteilung schnell hintereinander auftreten, wird im Algorithmus für den automatischen Ausgleich die Rate begrenzt, mit der Weiterleitungsanforderungen ausgeführt werden.

Sie können den aktuellen Status von Anwendungen in Warteschlangen in einem Cluster und Anwendungsinstanzen überwachen. Weitere Informationen finden Sie im Abschnitt Verteilung der Anwendungslast überwachen. Die Lösungen im Abschnitt Fehlerbehebung für Anwendungsausgleich schließlich helfen Ihnen bei Problemen im Zusammenhang mit der Verteilung der Anwendungslast.

Die erneute Lastverteilung ist nur für Anwendungen mit einer langen Verbindungszeit nützlich. Wenn Sie Clientanwendungen mit kurzen Verbindungszeiten haben, z. B. Clientanwendungen zum regelmäßigen Herstellen und Trennen einer Verbindung zu verschiedenen Warteschlangenmanagern, sollten Sie diese als nicht wiederverbindungsfähig konfigurieren. Dadurch werden sie aus der Gruppe von Anwendungen entfernt, die die Warteschlangenmanager verteilen möchten.

**Zugehörige Konzepte**

„Verwendung der automatischen Verbindungswiederholung durch den automatischen Abgleich” auf Seite 448
Ab IBM MQ 9.2.0 verwendet die automatische Verteilung im Uniform-Cluster die Erweiterungen für die vorhandene Funktion zur automatischen Verbindungswiederholung von IBM MQ.

## Multi *Funktionsweise des automatischen Ausgleichs*

Im Uniform-Cluster werden Clientverbindungen auf Basis des Anwendungsnamens gruppiert. Anwendungen, die eine Verbindung zu einem beliebigen Mitglied des Uniform-Clusters mithilfe des gleichen Anwendungsnamens herstellen, werden von allen anderen Anwendungen mit dem gleichen Anwendungsnamen als äquivalent betrachtet.

Durch die automatische Verteilung der Anwendungslast wird eine gleichmäßige Verteilung von Anwendungsinstanzen auf die Mitglieder des Clusters sichergestellt. Weitere Informationen finden Sie im Abschnitt „Anwendungsnamen und Anwendungsinstanzen” auf Seite 444. Verwenden Sie den Befehl DISPLAY APSTATUS , um den Status einer oder mehrerer Anwendungen und Anwendungsinstanzen anzuzeigen, die mit einem Warteschlangenmanager oder einem einheitlichen Cluster verbunden sind.

Sie können beispielsweise festlegen, dass alle Instanzen einer Versicherungsanfrageanwendung den Anwendungsnamen "INSURANCE.REQUESTS". Zugehörige Verbindungen von dieser Anwendung werden entsprechend automatisch in Instanzen gruppiert, wobei die gesamte Lastverteilung je Instanz durchgeführt wird.

Wenn neue Instanzen der Anwendung eine Verbindung zu einem Mitglied des Uniform-Clusters herstellen, bewertet der Algorithmus für die automatische Verteilung der Anwendungslast, welche Warteschlangenmanager über die wenigstens Instanzen von INSURANCE.REQUESTS verfügen, und leitet einige Verbindungen an diese Warteschlangenmanager weiter.

Der automatische Ausgleich ist nur in den folgenden Fällen aktiviert:

- Der Wert SHARECNV des Kanals ist größer als null.
- Eine der folgenden Bedingungen ist erfüllt:
  - Die Clientanwendung gibt MQCNO_RECONNECT an
  - Die Datei `mqclient.ini` gibt **`Defrecon`**=YES an.

**Anmerkung:** Anwendungen mit einer Affinität mit dem Warteschlangenmanager - beispielsweise aufgrund einer permanenten Subskription oder einer dynamischen Empfangswarteschlange für Antworten

- können nicht sicher neu verteilt werden und sollten die Option MQCNO_RECONNECT_QMGR oder gar keine Option zur Verbindungswiederholung verwenden.

Bei der Weiterleitung eines Clients an einen alternativen Warteschlangenmanager verwendet dieser wie üblich die Definitionstabellen für den lokalen Clientkanal (CCDTs), um die Verbindungsinformationen für das neue Ziel zu suchen. Daher ist es für einen reibungslosen und effizienten Betrieb der automatischen Lastverteilung wichtig, dass Clients eine CCDT verwenden, die einen Eintrag für jedes Mitglied des Uniform-Clusters sowie für jede Warteschlangenmananagergruppe, die zum Ausgleich der einleitenden Verbindungen verwendet wird, enthält.

Die Verwendung einer CCDT im JSON-Format vereinfacht dies, da mehrere Verbindungen mit dem gleichen Serververbindungsnamen möglich sind. Weitere Informationen finden Sie unter „CCDT im JSON-Format konfigurieren” auf Seite 47.

**Zugehörige Konzepte**

„Verwendung der automatischen Verbindungswiederholung durch den automatischen Abgleich” auf Seite 448

Ab IBM MQ 9.2.0 verwendet die automatische Verteilung im Uniform-Cluster die Erweiterungen für die vorhandene Funktion zur automatischen Verbindungswiederholung von IBM MQ.

## ALW *Automatischer Ausgleich von JMS-Anwendungen*

Wenn Jakarta Messaging 3.0 -oder Java Message Service 2.0 -Anwendungen automatisch ausgeglichen werden, werden die zugrunde liegenden Gruppen von IBM MQ -Verbindungen, die von JMS -Anwendungen erstellt werden, zusammen verschoben.

V 9.3.0 Ab IBM MQ 9.3.0ist die Eigenschaft **dynamicallyBalanced** verfügbar, wenn ActivationSpecskonfiguriert wird. Diese Eigenschaft gibt an, ob eine MDB im Rahmen des Anwendungsausgleichs in einem einheitlichen Cluster angefordert werden kann, Nachrichten von einem anderen Warteschlangenmanager zu empfangen. Weitere Informationen finden Sie im Abschnitt Ressourcenadapter für eingehende Kommunikation konfigurieren.

Für die Handhabung von JMS-Verbindungen haben Uniform-Cluster das Konzept einer *Anwendungsinstanz*. Für JMS ist eine *Anwendungsinstanz* als eine JMS-Verbindung und alle ihre zugehörigen JMS-Sitzungen definiert.

Es ist ein eindeutiger Verbindungstag der Clientverbindung zugeordnet, die der JMS-Verbindung entspricht. Der gleiche Tag wird dann für die Clientverbindungen übernommen, die JMS-Sitzungen entsprechen, die von dieser JMS-Verbindung erstellt werden.

Hier ein Beispiel, bei dem ein Paar von Clientanwendungen JMS-Anwendungen an einem Uniform-Cluster mit einem einzigen aktiven Warteschlangenmanager (Warteschlangenmanager 1) ausführt:

- Client 1 erstellt eine Verbindungsfactory, bei der er den Anwendungsnamen "App1" festlegt, sowie eine JMS-Verbindung und drei JMS-Sitzungen. Client 1 erstellt vier Clientverbindungen in Warteschlangenmanager 1, von denen jede den gleichen Verbindungstag aufweist. Dies wird als einzelne Instanz von "App1" behandelt.
- Client 2 erstellt eine Verbindungsfactory, bei der er den Anwendungsnamen "App1" festlegt, sowie eine JMS-Verbindung und zwei JMS-Sitzungen. Client 2 erstellt drei Clientverbindungen, von denen jede den gleichen Verbindungstag aufweist (einen anderen als derjenige, der Client 1 zugewiesen wurde). Dies wird als einzelne, separate Instanz von "App1" behandelt.
- Der Warteschlangenmanager erkennt somit zwei Instanzen von "App1".

Bei Durchführung des automatischen Ausgleichs werden Anwendungsinstanzen verschoben. Ein Warteschlangenmanager wählt eine Anwendungsinstanz (eine Gruppe von Clientverbindungen mit demselben Verbindungstag) aus und fordert an, dass die Instanz auf einen anderen Warteschlangenmanager verschoben wird. Der Clientcode empfängt die Anforderung und stellt sicher, dass alle zugehörigen Verbindungen (die einer JMS-Verbindung und den zugehörigen JMS-Sitzungen entsprechen) in den neuen Warteschlangenmanager wechseln.

Nehmen wir als Beispiel die zuvor erwähnten Anwendungsinstanzen und gehen davon aus, dass ein neuer Warteschlangenmanager (Warteschlangenmanager 2) im Uniform-Cluster initialisiert wird.

Warteschlangenmanager 2 hat keine Arbeit, aber Warteschlangenmanager 1 hat zwei Instanzen von "App1", d. h., Warteschlangenmanager 2 fordert an, dass Warteschlangenmanager 1 eine Instanz von "App1" an Warteschlangenmanager 2 überträgt.

Warteschlangenmanager 1 wählt eine Instanz von "App1" aus, die verschoben werden soll. Im Beispiel wird davon ausgegangen, dass die Instanz, die von Client 1 erstellt wurde, ausgewählt wird.

- Warteschlangenmanager 1 sendet eine Anforderung an Client 1, seine Instanz von "App1" zu QM2 zu verschieben.
- Der Client schließt seine vier vorhandenen Clientverbindungen zu Warteschlangenmanager 1 und erstellt vier neue Verbindungen zu WS-Manager 2.
- Die JMS-Verbindung und deren JMS-Sitzungen sollten, abgesehen von einer kurzen Verarbeitungspause, normalerweise nicht gestört werden.

**Anmerkung:**

Eine Anwendung erhält möglicherweise eine JMS-Ausnahmebedingung, wenn bestimmte Operationen während des Verschiebens einer Anwendungsinstanz ausgeführt werden.

Die JMS-Ausnahmebedingung verfügt über eine verlinkte IBM MQ-Ausnahmebedingung, aus der der Ursachencode abgerufen werden kann, um die Ursache des Fehlers zu ermitteln.

Die erwarteten Ursachencodes lauten wie folgt:

**MQRC_CALL_INTERRUPTED**
Dies tritt auf, wenn z. B. eine Nachricht, die persistent ist (der Standard in JMS), außerhalb eines Synchronisationspunkts eingereiht wird, die Operation jedoch durch eine Verbindungswiederholung unterbrochen wird.

**MQRC_BACKED_OUT**
Dies tritt auf, wenn beispielsweise ein Versuch, eine Nachricht innerhalb eines Synchronisationspunktes einzureihen, durch eine Verbindungswiederholung unterbrochen wird.

**Zugehörige Konzepte**

„Funktionsweise des automatischen Ausgleichs” auf Seite 446
Im Uniform-Cluster werden Clientverbindungen auf Basis des Anwendungsnamens gruppiert. Anwendungen, die eine Verbindung zu einem beliebigen Mitglied des Uniform-Clusters mithilfe des gleichen Anwendungsnamens herstellen, werden von allen anderen Anwendungen mit dem gleichen Anwendungsnamen als äquivalent betrachtet.

„Verwendung der automatischen Verbindungswiederholung durch den automatischen Abgleich” auf Seite 448
Ab IBM MQ 9.2.0 verwendet die automatische Verteilung im Uniform-Cluster die Erweiterungen für die vorhandene Funktion zur automatischen Verbindungswiederholung von IBM MQ.

## Multi *Verwendung der automatischen Verbindungswiederholung durch den automatischen Abgleich*

Ab IBM MQ 9.2.0 verwendet die automatische Verteilung im Uniform-Cluster die Erweiterungen für die vorhandene Funktion zur automatischen Verbindungswiederholung von IBM MQ.

In den IBM MQ-Versionen vor IBM MQ 9.2.0 stellt die Funktion zur automatischen Verbindungswiederholung automatisch eine Verbindung zu einer Standby-Instanz eines Warteschlangenmanagers oder zu einem anderen Warteschlangenmanager her, je nach den bereitgestellten Verbindungsdetails, in der Regel eine Verbindungsnamensliste oder eine Clientkanaldefinitionstabelle (CCDT).

Der IBM MQ-Client führt in einigen Fällen eine unbeaufsichtigte Verbindungswiederholung aus, ohne dass die Anwendung davon Kenntnis hat. Die Entscheidung darüber, mit welchem Warteschlangenmanager die Verbindung wiederhergestellt werden soll, liegt allein an der Reihenfolge der Verbindungsnamen in einer Verbindungsnamensliste oder der Lastausgleichskonfiguration in der CCDT.

Ab IBM MQ 9.2.0 kann eine Anforderung zur Verbindungswiederholung an einen Client gesendet werden, die einen Hinweis enthält, mit welchem Warteschlangenmanager der Client die Verbindung wiederherstellen soll. In vielen Szenarios zur Verbindungswiederholung (z. B. bei einem Warteschlangenmanager-

fehler) oder wenn der Administrator den Befehl **endmqm -r** ausgibt ist kein Warteschlangenmanagername im Hinweis enthalten, und die automatische Wiederholung der Verbindung wird so ausgeführt, wie das aktuell der Fall ist.

Wenn Sie allerdings einen Uniform-Cluster konfiguriert haben, sendet der automatische Anwendungsausgleich regelmäßig Anforderungen zur Verbindungswiederholung an Clients, um einen ausgeglichenen Cluster zu erhalten. In diesen Fällen gibt der Uniform-Cluster im Hinweis für die Verbindungswiederholung einen Warteschlangenmanagernamen an, um sicherzustellen, dass Clientverbindungen zu den Warteschlangenmanagern mit den wenigsten Verbindungen verschoben werden.

Für die Funktion des automatischen Lastausgleichs sind folgende Punkte wichtig:

- IBM MQ-Anwendungen verwenden CCDTs, um Verbindungsinformationen abzurufen.
- CCDTs enthalten einen Eintrag für jeden Warteschlangenmanager im Uniform-Cluster

Ist dies nicht der Fall, kann der Cluster Anwendungen nicht automatisch auf alle Mitglieder des Clusters verteilen.

Wenn eine Anwendung eine Version des IBM MQ-Clients vor IBM MQ 9.2.0 verwendet und für die Unterstützung der automatischen Clientverbindungswiederholung konfiguriert ist, wird der Anwendung möglicherweise eine Anforderung vom Uniform-Cluster gesendet, um die entsprechenden Schritte zur Verbindungswiederholung auszuführen.

Der Client wird nicht aufgefordert, eine erneute Verbindung zu einem bestimmten Warteschlangenmanager herzustellen, sondern führt die gleiche Reihenfolge der Wiederverbindungslogik wie für andere Wiederverbindungsereignisse aus. Es ist möglich, eine gleichmäßige Verteilung von Clientanwendungen vor IBM MQ 9.2.0 im Uniform-Cluster zu erreichen, indem sichergestellt wird, dass Clients für die Verwendung von CCDTs konfiguriert sind, die gleichmäßig gewichtete Einträge für jedes Mitglied des Clusters enthalten.

Anwendungen können mehrere Versuche zur Verbindungswiederholung vornehmen, bis eine Verbindung zu einem Warteschlangenmanager hergestellt wird, für den die zusätzliche Instanz erforderlich ist. Dies ist allerdings eine weniger effiziente Möglichkeit, eine gleichmäßige Verteilung von Anwendungen im gesamten Cluster zu erreichen. Der automatische Ausgleich an in diesen Umgebungen mehr Zeit in Anspruch nehmen.

## Für IBM MQ-Clients wird die automatisch Clientverbindungswiederholung nicht unterstützt

Wenn eine Anwendung eine Version des IBM MQ-Clients verwendet, in der die automatische Clientverbindungswiederholung nicht unterstützt wird, kann die Anwendung einen Fehlerrückgabecode von einem MQI-Aufruf empfangen.

Wenn Ihre Anwendung nicht für die Verarbeitung von Fehlern und der manuellen Ausführung von Verbindungswiederholungen entwickelt wurde, muss der automatische Ausgleich für diese Anwendungen möglicherweise inaktiviert werden.

**Anmerkung:** Der automatische Ausgleich ist für jede Anwendung aktiviert, die als wiederverbindungsfähig ermittelt wurde, d. h., für die MQCNO_RECONNECT in den effektiven Verbindungsoptionen vorhanden ist.

**Zugehörige Tasks**

„Neuen Uniform-Cluster erstellen” auf Seite 459
Hier finden Sie Informationen, wie ein neuer Uniform-Cluster erstellt wird.

## Multi V 9.3.0 *Beeinflussung der Anwendungsumverteilung in einheitlichen Clustern*

Mit einem automatischen Anwendungsausgleich (ein Feature von einheitlichen Clustern) kann eine Anwendungsverbindung aufgefordert werden, an einem beliebigen Punkt in seinem Lebenszyklus zu einem alternativen Warteschlangenmanager zu wechseln.

## Einführung

Ab IBM MQ 9.3.0versucht der Lastausgleichsalgorithmus automatisch, den Status von Anwendungen zu berücksichtigen, um die Unterbrechung des Anwendungsablaufs zu minimieren. Dies kann auf bestimmte Anwendungen oder Anwendungsinstanzen abgestimmt werden, indem IBM MQ zusätzliche Informationen zum Typ der Anwendung oder zum Muster der IBM MQ-Aktivität erteilt wird, die von dieser Anwendung ausgeführt werden.

In der Regel ist es wahrscheinlich, dass die Person, die eine Clientanwendung entwickelt oder implementiert, am besten geeignet ist, dieses Muster zu verstehen und diese Informationen dem Warteschlangenmanager zu liefern (siehe Flexible und skalierbare Clientanwendungen implementieren), es kann aber auch oder zusätzlich von einem Administrator optimiert werden.

Beachten Sie, dass, wenn der Warteschlangenmanager keine gleichmäßige Verteilung von Anwendungen innerhalb eines angemessenen Zeitraums erreichen kann, Anwendungsverbindungen möglicherweise immer noch mit anderen Warteschlangenmanagern abgeglichen werden, ohne auf eine passende Zeit in ihrem IBM MQ-Flow warten zu müssen.

Dies kann auch auf die Anforderungen abgestimmt werden. Wenn es wichtiger ist, schnell eine gleichmäßige Verteilung von Anwendungen zu erreichen, können Sie das Produkt so konfigurieren, dass es weniger Zeit abwartet, um eine geeignete Zeit für die Neuverteilung einer Anwendung zu finden. Wenn es wichtiger ist, eine Unterbrechung von Anwendungen zu verhindern, ist es alternativ möglich, das Produkt so zu konfigurieren, dass es immer auf eine passende Zeit wartet, um die Anwendung zu verschieben.

Weitere Übersichtsinformationen finden Sie unter Flexible und skalierbare Clientanwendungen implementieren .

Informationen zu den .NET-Anwendungen finden Sie unter „Anwendungsneuverteilung in .NET beeinflussen” auf Seite 453.

Weitere Informationen zu .XMS.NET-Anwendungen finden Sie unter Eigenschaften von ConnectionFactory .

V 9.3.4 Weitere Informationen zu JMS-Anwendungen finden Sie unter „Anwendungsneuverteilung in IBM MQ classes for JMS beeinflussen” auf Seite 454 .

## Standardverhalten des Anwendungsausgleichs

Standardmäßig wird die Transaktion/Einheit des Arbeitsstatus einer Anwendungsinteraktion mit einem Warteschlangenmanager für alle Anwendungen berücksichtigt.

Bei lokalen Transaktionen vermeidet der automatische Anwendungsausgleich die Ausgabe von Neuabgleichsanforderungen an Anwendungen, die derzeit an einer Transaktion beteiligt sind. Dies beseitigt zwar nicht die Möglichkeit einer Anwendung, die einen gesicherten Rückkehrcode empfängt, da das Erreichen des konfigurierten Zeitlimits für eine Neuverteilung oder eine echte Betriebsunterbrechung dennoch einen solchen Rückkehrcode verursachen könnte, bedeutet aber, dass Anwendungen normalerweise nicht aufgefordert werden, die Verbindung wiederherzustellen, während sie sich mitten in einer Transaktion befinden.

Bei Anwendungen, die eine neue Transaktion fast unmittelbar nach dem Abschluss der vorherigen Transaktion beginnen, kann es zu einer Verzögerung für den ersten Aufruf in der neuen Transaktion kommen, während der Neuausgleich abgeschlossen ist. Dadurch wird sichergestellt, dass ein automatischer Anwendungsausgleich noch möglich ist, um eine gleichmäßige Verteilung von Anwendungen zwischen den Warteschlangenmanagern in einem einheitlichen Cluster zu erreichen.

Wenn Sie Anwendungen haben, die länger aktive Transaktionen verwenden, sollten Sie in Betracht ziehen, den Wert des Zeitlimits für die Neuverteilung zu erhöhen oder diese Einschränkung vollständig zu inaktivieren. Im Abschnitt „Ausgleichsverhalten konfigurieren” auf Seite 451 finden Sie Links zur Steuerung dieses in MQI und .NET oder zum Entwerfen von Clientanwendungen für Fehlertoleranz und Skalierbarkeit für die entsprechende Codeversion.

## Anforderung-Antwort-Ausgleich

Wenn der Anwendungstyp als **Request-Reply**angegeben ist, wird für jede PUT-Operation, die die Anwendungsinstanz ausführt, eine GET-Antwort erwartet. Wenn es sich bei der Anwendungsinstanz um mehrere Threads handelt oder um Anforderungen und Antworten in Batches, können mehrere Anforderungen und Antworten zu einem bestimmten Zeitpunkt im Flug ausgeführt werden.

Die Anwendung gilt erst dann als zum Verschieben auswählbar, wenn die Anzahl der gesendeten Anforderungen gleich der Anzahl der empfangenen Antworten ist oder der Wert für 'backstop' des Zeitlimits überschritten wird.

Eine Ausnahme hiervon ist, wenn der Nachrichtenablauf für eine Anforderungsnachricht konfiguriert ist. Es wird davon ausgegangen, dass Antworten innerhalb des Ablaufintervalls der Anforderungsnachricht empfangen werden sollen und wenn alle Anforderungsnachrichten abgelaufen sind, wartet der Ausgleichsalgorithmus nicht mehr auf zusätzliche Antworten, bevor die Instanz in Betracht gezogen wird, die für den Umzug auswählbar ist.

Wenn mehrere Anforderungen ausstehen, wird nur der späteste Ablauf unter den gesendeten Anforderungsnachrichten berücksichtigt. Wenn aussagefähige Ablaufwerte verwendet werden, sollten Sie den **Timeout** -Ausgleichsparameter für die Anwendung so konfigurieren, dass er mindestens so hoch ist wie jeder Ablauf von gesendeten Nachrichten, um zu vermeiden, dass das erwartete Ablauffenster für Anforderungen/Antworten verkürzt wird.

Das vorhergehende Muster ist nur für Anwendungen geeignet, die erwarten, dass es Zeiträume gibt, in denen keine ausstehenden Anforderungen vorhanden sind. Komplexe Multithread-Anwendungen, die beispielsweise immer Nachrichten senden und empfangen, werden unter diesem Muster möglicherweise nie für eine Neuverteilung in Frage kommen.

**Anmerkungen:**

- Es wird kein Versuch unternommen, bestimmte Anforderungen und Antworten zu korrelieren, so dass die Anwendung möglicherweise noch wartet, bis die letzte Anforderung abläuft, bevor sie für ein Gleichgewicht in Frage kommt, wenn eine frühere Antwort innerhalb eines Batch von Flugnachrichten abläuft.
- Insbesondere ist Vorsicht geboten, wenn aus ähnlichen Gründen eine unbegrenzte Ablaufzeit und ablaufende Nachrichten kombiniert werden.

  Wenn Anforderungsnachrichten mit einem begrenzten Ablauf ausstehen und neue Nachrichten mit einer unbegrenzten Ablaufzeit gesendet werden, wird das unbegrenzte Ablaufzeitlimit *nicht* vom Ausgleichsalgorithmus berücksichtigt, der weiterhin die aktuelle späteste Ablaufzeit berücksichtigt.

  Andernfalls können frühere Antworten, die abgelaufen sind, verhindern, dass der Antrag jemals in Anspruch nehmen kann. Entsprechend wird die Wartezeit auf den längsten (begrenzten) Ablauf reduziert, wenn Antworten mit unbegrenzten Ablaufzeitlimits ausstehen, aber anschließend ablaufende Anforderungen gesendet werden.

  Im Allgemeinen sollten Sie vermeiden, dass eine einzelne Anwendungsinstanz sowohl auslaufende als auch nicht ablaufende Anforderungsnachrichten in einer ausgewogenen Anwendung sendet, da für einen Entwickler oder Administrator die Berechtigung zum Neuausgleich schwieriger wird, um eine genaue Spur zu finden oder zu definieren.
- Nur die von der sendenden Anwendung angegebene Ablaufzeit (z. B. in der MQI den Wert MQMD.**Expiry**) wird bei der Bestimmung berücksichtigt, wie lange auf Antworten gewartet werden soll. Nachfolgende Änderungen an diesem Wert, z. B. die Verwendung von CAPEXPRY, wirken sich nicht auf die Wartezeit aus.

## Ausgleichsverhalten konfigurieren

Um genau zu beeinflussen, wann IBM MQ Anwendungen neu ausgeglichen, können bestimmte Clientanwendungsumgebungen Informationen zur Verbindungszeit für das verwendete Messaging-Muster bereitstellen.

Diese Informationen werden in einer neuen Struktur bereitgestellt, die als *Ausgleichsoptionen* bezeichnet wird.

Informationen zu MQI finden Sie unter „Ausgleichsverhalten mit der MQI konfigurieren” auf Seite 452.

Informationen zum .NET-Client-Äquivalent dieser Struktur finden Sie in „Anwendungsneuverteilung in .NET beeinflussen” auf Seite 453.

V 9.3.4 Weitere Informationen zur JMS-Methode zum Festlegen dieser Optionen finden Sie im Artikel „Anwendungsneuverteilung in IBM MQ classes for JMS beeinflussen” auf Seite 454 .

Andere Clientumgebungen unterstützen derzeit keine Unterstützung für die Bereitstellung dieser Struktur zur Verbindungszeit.

Multi V 9.3.0 *Ausgleichsverhalten mit der MQI konfigurieren*

Um genau zu beeinflussen, wann IBM MQ Anwendungen neu ausgeglichen, können bestimmte Clientanwendungsumgebungen Informationen zur Verbindungszeit für das verwendete Messaging-Muster bereitstellen.

In der MQI wird die Struktur der Ausgleichsoptionen als MQBNO bezeichnet.

Wenn in Ihrem Programm keine *Ausgleichsoptionen* bereitgestellt werden, leiten unterstützende Clients diese Informationen in der Anwendungszeilengruppe oder in Zeilengruppe 'applicationDefaults' in der `client.ini` -Datei ab, die neben der Clientanwendung implementiert ist.

**Anmerkung:** Diese Zeilengruppen sind identisch, mit der Ausnahme, dass die `Application` -Version ein Feld **`Name`** enthält, um anzugeben, für welche Anwendung diese Optionen gelten.

Wenn eine Form der Zeilengruppe angegeben wird, müssen alle Felder vorhanden sein, mit Ausnahme von **`BalanceOptions`** , das als none angenommen wird, wenn nicht explizit festgelegt.

Die bevorzugte Reihenfolge für die Bereitstellung von Optionen ist:

1. Eine MQBNO-Struktur wird von der Anwendung in CONNX bereitgestellt und vollständig verwendet.
2. Oder die übereinstimmende benannte Zeilengruppe `Application` , falls vorhanden, wird ausschließlich zum Generieren einer Zeilengruppe verwendet.
3. Oder die Zeilengruppe `ApplicationDefaults` , falls vorhanden, wird nur verwendet, um eine zu generieren.
4. Oder es werden keine MQBNO-Flüsse für diese Verbindung verwendet.

Sie können drei Schlüsselteile von Informationen aus der MQBNO-Struktur oder der `client.ini`-Datei angeben:

1. Die **`ApplicationType`** oder das Muster der Anwendung

   Dieses Feld gibt IBM MQ das allgemeine Muster der IBM MQktivität an, an der diese Anwendung beteiligt ist.

   Es werden drei Arten von Anwendungen unterstützt:

   **Einfach**
   Über die in „Standardverhalten des Anwendungsausgleichs” auf Seite 450 beschriebenen Standardwerte hinaus sollten keine spezifischen Regeln angewendet werden.

   **Anforderung-Antwort**
   Nach jedem MQPUT-Aufruf wird für eine Antwortnachricht ein übereinstimmender MQGET-Aufruf erwartet. Weitere Informationen finden Sie unter „Anforderung-Antwort-Ausgleich” auf Seite 451.

   **verwalteter Client**
   Der Neuausgleich von Anforderungen wird immer sofort an den Client gesendet, wobei der Neuausgleich an einem Punkt, den er für angebracht hält, zum Beispiel der JEE-Ressourcenadapter auf diese Weise registriert würde.
2. Die **`Timeout`** , nach der die Neuverteilung die Anwendungsaktivität unterbrechen kann
3. Bestimmte **`BalanceOptions`**

## Beispiele, wenn Ihre Anwendung neu ausgeglichen werden kann

### Beispiel 1

Sie haben eine Anwendung geschrieben, die Nachrichten unter Synchronisationspunkt setzt und den Stapel von Nachrichten festschreibt, indem Sie einen MQCMIT-Aufruf ausgeben. Wenn der MQCMIT-Aufruf abgeschlossen ist, beginnt die Anwendung, Nachrichten unter einen neuen Synchronisationspunkt zu setzen.

**Empfohlene Konfiguration von IBM MQ**
Standardoptionen ausreichend

**Ergebnis**
Eine Anwendungsinstanz wird versetzt, nachdem ein MQCMIT-Aufruf erfolgreich ausgeführt wurde (oder fehlschlägt), sobald die konfigurierte Anzahl von Transaktionen erfüllt wurde.

Wenn ein Nachrichtenstapel mehr als 10 Sekunden überschreitet, kann es standardmäßig rückgängig gemacht werden, wenn ein Neuausgleich angefordert wurde. Wenn Sie erwarten, dass Transaktionen diesen Grenzwert regelmäßig überschreiten und dies zulässig sein muss, können Sie **Timeout** entsprechend erweitern.

### Beispiel 2

Sie haben eine Anwendung geschrieben, die eine Nachricht in eine Clusterwarteschlangeninstanz einreiht, und eine andere Anwendung antwortet auf eine lokale temporäre dynamische Warteschlange mit einer Nachricht, nachdem sie die Anforderung verarbeitet hat. Wenn die Anforderung destruktiv aus der lokalen Warteschlange gelesen wurde, setzt die Anwendung ihre nächste Anforderungsnachricht.

**Empfohlene Konfiguration von IBM MQ**
Setzen Sie Typ auf MQBNO_BALTYPE_REQREP

**Ergebnis**
Eine Anwendungsinstanz wird verschoben, wenn eine Anwendung einen MQGET-Aufruf abschließt, an dem die Anwendungsinstanz in einen anderen Warteschlangenmanager versetzt wird. Alle nachfolgenden MQPUT-Aufrufe werden auf dem neuen Warteschlangenmanager ausgeführt.

MQBNO
ApplicationType

Multi V 9.3.0 *Anwendungsneuverteilung in .NET beeinflussen*

Ab IBM MQ 9.3.0 stehen Ihnen zusätzliche Konstanten zur Verfügung, um die Eigenschaften für die Ausgleichsoption mithilfe einer Hashtabelle aus der Anwendung festzulegen, wenn Sie die `MQQueueManager`-Klasse für die Verbindung zum Warteschlangenmanager verwenden.

Die folgenden Konstanten werden verwendet, um die Anwendungsverteilung in .NET zu beeinflussen:

**Anwendungsart neu ausgleichen**
Der Typ der Ausgleichsaktion, dargestellt durch die Konstante **MQC.BALANCING_APPLICATION_TYPE_PROPERTY**

- Mit dieser Eigenschaft müssen Sie das Feld **ApplicationType** der MQBNO-Struktur festlegen.

  Sie müssen Werte vom Typ "Integer" festlegen und die möglichen Werte lauten:

  **MQC.BALANCING_APPLICATION_TYPE_SIMPLE**
  Einfacher Ausgleich; zusätzlich zu den in „Beeinflussung der Anwendungsumverteilung in einheitlichen Clustern" auf Seite 449 beschriebenen Regeln werden keine spezifischen Regeln angewendet. Dies ist der Standardwert.

  **MQC.BALANCING_APPLICATION_TYPE_REQUEST_REPLY**
  Anforderung-Antwortausgleich; nach jedem **MQPUT**-Aufruf wird ein entsprechender **MQGET**-Aufruf für eine Antwortnachricht erwartet. Der Lastausgleich wird verzögert, bis eine solche Nachricht empfangen wird oder die Anforderungsnachricht **EXPIRY** überschritten wurde.

Wenn die Verbindungswiederholung von der Anwendung aktiviert wird und diese Eigenschaft nicht festgelegt ist, wird **MQC.BALANCING_APPLICATION_TYPE_SIMPLE** verwendet.

**Optionen für Neuausgleich**
Die von der ausgebenden Anwendung festgelegten Ausgleichsoptionen; dargestellt durch die Konstante **MQC.BALANCING_OPTIONS_PROPERTY**

- Mit dieser Eigenschaft müssen Sie das Feld **BalanceOptions** der MQBNO-Struktur festlegen.

  Sie müssen Werte vom Typ "Integer" festlegen und die möglichen Werte lauten:

  **MQC.BALANCING_OPTIONS_NONE**
  Es sind keine Optionen eingestellt. Dies ist der Standardwert.

  **MQC.BALANCING_OPTIONS_IGNORE_TRANSACTIONS**
  Wenn diese Option aktiviert ist, können Anwendungen auch während einer laufenden Transaktion neu abgeglichen werden.

  Wenn die Verbindungswiederholung von der Anwendung aktiviert wird und diese Eigenschaft nicht festgelegt ist, wird **MQC.BALANCING_OPTIONS_NONE** verwendet.

**Zeitlimit für Neuausgleich**
Zeitlimit, nach dem die Neuverteilung die Anwendungsaktivität unterbrechen kann; dargestellt durch die Konstante **MQC.BALANCING_TIMEOUT_PROPERTY**

- Sie müssen diese Eigenschaft verwenden, um das Feld Zeitlimit der MQBNO-Struktur festzulegen.

  Sie müssen Werte vom Typ "Integer" festlegen und die möglichen Werte lauten:

  **MQC.BALANCING_TIMEOUT_AS_DEFAULT**
  Der festgelegte Standardwert für die Zeitlimitüberschreitung. Dies ist der Standardwert.

  **MQC.BALANCING_TIMEOUT_IMMEDIATE**
  Sofortige Zeitlimitüberschreitung

  **MQC.BALANCING_TIMEOUT_NEVER**
  Es tritt keine Zeitlimitüberschreitung auf

  **Anmerkung:** Sie müssen einen Wert nur aus den definierten Werten oder einem Wert von 0 bis 999999999 Sekunden angeben.

Flexible und skalierbare Clientanwendungen implementieren
MQBNO

V 9.3.4 Multi *Anwendungsneuverteilung in IBM MQ classes for JMS beeinflussen*
Ab IBM MQ 9.3.4stehen Ihnen zusätzliche Konstanten zur Verfügung, um die Eigenschaften für die Ausgleichsoption in einem **ConnectionFactory**festzulegen. Diese Konstanten sind nur gültig, wenn **WMQ_PROVIDER_VERSION** auf 7gesetzt ist. `Request_reply` -Anwendungen in einem einheitlichen Cluster müssen die Möglichkeit fehlender Antworten zulassen.

- „Die verfügbaren Konstanten” auf Seite 454.
- „Das Potenzial für verlorene Nachrichten beim Lastausgleich von REQUEST_REPLY -Anwendungen” auf Seite 455.

## Die verfügbaren Konstanten

Die folgenden Konstanten werden verwendet, um die Anwendungsverteilung in IBM MQ classes for JMS zu beeinflussen:

**Anwendungsart neu ausgleichen**
Der Typ der Ausgleichsaktion, dargestellt durch die Konstante **WMQConstants.WMQ_BALANCING_APPLICATION_TYPE**

- Sie müssen diese Eigenschaft verwenden, um das Feld **ApplicationType** der Struktur MQBNO festzulegen.

  Sie müssen Werte vom Typ "Ganzzahl" festlegen. Mögliche Werte:

**WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_SIMPLE(Standard)**
Einfacher Ausgleich; zusätzlich zu den in „Beeinflussung der Anwendungsumverteilung in einheitlichen Clustern" auf Seite 449 beschriebenen Regeln werden keine spezifischen Regeln angewendet.

**WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_REQUEST_REPLY**
Anforderung-Antwortausgleich; nach jedem **MQPUT**-Aufruf wird ein entsprechender **MQGET**-Aufruf für eine Antwortnachricht erwartet. Der Lastausgleich wird verzögert, bis eine solche Nachricht empfangen wird oder die Anforderungsnachricht **EXPIRY** überschritten wurde.

Wenn die Verbindungswiederholung von der Anwendung aktiviert wird und diese Eigenschaft nicht festgelegt ist, wird **WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_SIMPLE** verwendet.

**Optionen für Neuausgleich**
Die von der ausgebenden Anwendung festgelegten Ausgleichsoptionen; dargestellt durch die Konstante **WMQConstants.WMQ_BALANCING_OPTIONS**

- Sie müssen diese Eigenschaft verwenden, um das Feld **BalanceOptions** der Struktur MQBNO festzulegen.

  Sie müssen Werte vom Typ "Ganzzahl" festlegen. Mögliche Werte:

  **WMQConstants.WMQ_BALANCING_OPTIONS_NONE(Standard)**
  Es sind keine Optionen eingestellt.

  **WMQConstants.WMQ_BALANCING_OPTIONS_IGNORE_TRANSACTIONS**
  Wenn diese Option aktiviert ist, können Anwendungen auch während einer laufenden Transaktion neu abgeglichen werden.

  Wenn die Verbindungswiederholung von der Anwendung aktiviert wird und diese Eigenschaft nicht festgelegt ist, wird **WMQConstants.WMQ_BALANCING_OPTIONS_NONE** verwendet.

**Zeitlimit für Neuausgleich**
Das Zeitlimit, nach dem die Neuverteilung die Anwendungsaktivität unterbrechen kann; dargestellt durch die Konstante **WMQConstants.WMQ_BALANCING_TIMEOUT**

- Sie müssen diese Eigenschaft verwenden, um das Feld **Timeout** der Struktur MQBNO festzulegen.

  Sie müssen Werte vom Typ "Ganzzahl" festlegen. Mögliche Werte:

  **WMQConstants.WMQ_BALANCING_TIMEOUT_AS_DEFAULT(Standard)**
  Der festgelegte Standardwert für die Zeitlimitüberschreitung. Der Standardwert ist 10 Sekunden.

  **WMQConstants.WMQ_BALANCING_TIMEOUT_IMMEDIATE**
  Das Zeitlimit wird sofort überschritten.

  **WMQConstants.WMQ_BALANCING_TIMEOUT_NEVER**
  Es tritt keine Zeitlimitüberschreitung auf.

  **Wert zwischen 1 und 999999999**
  Gibt einen Wert in Sekunden an.

  **Anmerkung:** Sie müssen einen Wert nur aus den definierten Werten oder einem Wert von 0 bis 999999999 Sekunden angeben.

Diese Eigenschaften können auch in den JNDI-Darstellungen von Verbindungsfactorys über die Schnittstelle JMSAdmin oder IBM MQ Explorer festgelegt werden.

## Das Potenzial für verlorene Nachrichten beim Lastausgleich von REQUEST_REPLY -Anwendungen

In IBM MQ classes for JMS (und IBM MQ classes for Jakarta Messaging) wird die Anforderungs-/Antwortfunktionalität implementiert, indem die Eigenschaft **JMSReplyTo** in der Anforderungsnachricht festgelegt wird, die von der antwortenden Anwendung verwendet wird, um zu bestimmen, ob die Antwort gesendet wird. In JMS ist die Eigenschaft **JMSReplyTo** ein **Destination**.

Wenn dies in IBM MQ -Operationen umgesetzt wird, wird die Eigenschaft **JMSReplyTo** als vollständig qualifizierter Warteschlangen-URI gesendet, der eine Warteschlange in einem bestimmten Warteschlangenmanager angibt.

Aufgrund der asynchronen Spezifik der Behandlung von Neuausgleichsverbindungen kann eine Verbindungswiederherstellung eingeleitet werden, nachdem die Eigenschaft **JMSReplyTo** in einen vollständig qualifizierten URI umgesetzt wurde, aber bevor die Anforderungsnachricht in die Anforderungswarteschlange eingereiht wurde. Unter diesen Umständen sendet die antwortende Anwendung möglicherweise ihre Antwort an die ursprüngliche Antwortwarteschlange auf dem ursprünglichen Warteschlangenmanager, aber die anfordernde Anwendung wartet jetzt möglicherweise auf eine Antwort auf dem neuen Warteschlangenmanager.

`Request_reply` -Anwendungen in einem einheitlichen Cluster müssen daher die Möglichkeit verpasster Antworten berücksichtigen.

Flexible und skalierbare Clientanwendungen implementieren

MQBNO-Ausgleich-Optionen

## Multi *Einschränkungen und Überlegungen für Uniform-Cluster*

Einschränkungen und Überlegungen bei der Konfiguration von Uniform-Clustern.

**Anmerkung:** Allgemeine Voraussetzungen für die Konfiguration von Uniform-Clustern finden Sie unter „Neuen Uniform-Cluster erstellen” auf Seite 459.

### Bedeutung der Einheitlichkeit zwischen Warteschlangenmanagern

Standardmäßig kann jede Anwendung, die sich selbst als `reconnectable` deklariert, jederzeit auf einen alternativen Warteschlangenmanager in einem Uniform-Cluster neu verteilt werden. Dies bedeutet, dass alle Ressourcen, z. B. Warteschlangen-, Topic-oder Berechtigungsdatensätze, die für solche Anwendungen erforderlich sind, auf allen Warteschlangenmanagern im Uniform-Cluster deklariert werden müssen.

Die Konsistenz der Warteschlangenmanagerkonfiguration wird nicht auf Richtlinien angewendet. Es ist die Aufgabe Ihres Systemadministrators, die Mitglieder des Clusters so zu konfigurieren, dass sie eine ähnliche Konfiguration haben.

Sie können die Konsistenz jedoch verbessern, indem Sie die Funktion Automatische Konfiguration aus einem MQSC-Script beim Starten verwenden, um MQSC-Scripts gemeinsam zu nutzen, die Objekte für den Cluster definieren und somit sicherstellen, dass alle dieselben Definitionen haben. Weitere Informationen finden Sie unter „Neuen Uniform-Cluster erstellen” auf Seite 459.

Diese Einheitlichkeit gilt für Warteschlangenmanager mit vollständigem Repository für den Cluster. Obwohl es für traditionelle IBM MQ -Cluster häufig als bewährtes Verfahren angesehen wird, die vollständigen Repositorys auf eigenständigen Systemen zu trennen, besteht das Modell in einem einheitlichen Cluster darin, dass vollständige Repositorys vollständig an den Cluster-und Prozessanwendungsworkloads sowie an anderen Knoten teilnehmen.

### Überlappende Uniform-Cluster und traditionelle IBM MQ -Cluster

Ein Warteschlangenmanager kann an höchstens einem Uniform-Cluster teilnehmen, jedoch Mitglied einer beliebigen Anzahl standardmäßiger IBM MQ-Cluster sein. Es kann hilfreich sein, den Uniform-Cluster als einzelnen Warteschlangenmanager im größeren Cluster zu betrachten.

Ein Uniform-Cluster-Warteschlangenmanager darf nur für den Uniform-Cluster selbst als vollständiges Repository fungieren. Jeder Warteschlangenmanager, der zu einem Uniform-Cluster, aber auch zu einem breiteren traditionellen IBM MQ -Cluster gehört, kann nicht als Repository außerhalb des Uniform-Clusters verwendet werden. Weitere Informationen finden Sie unter Clusterwarteschlangenmanager für vollständige Repositorys auswählen.

Um einen einzelnen Warteschlangenmanager mit vollständigem Repository durch einen einheitlichen Cluster zu ersetzen, trennen Sie das vollständige Repository von der in Bearbeitung befindlichen Anwendungsarbeit und verschieben Sie nur die Anwendungsarbeit in den einheitlichen Cluster.

Wenn Sie automatische Definitionen für Uniform-Cluster verwenden, können Clusterkanäle nicht für die Verwendung in anderen Clustern gemeinsam genutzt werden, d. h., Sie setzen das Attribut **CLUSTER** auf den automatischen Cluster, und das Attribut **CLUSNL** muss leer sein.

## Aspekte des Anwendungsausgleichs

Anwendungsinstanzen werden nicht immer gleichmäßig verteilt, insbesondere unter den folgenden Umständen:

- Wenn weniger Anwendungsinstanzen als Warteschlangenmanager im Cluster vorhanden sind.
- Kurze Zeit, nachdem Clientanwendungen eine Verbindung zum Cluster hergestellt oder den Cluster verlassen haben.

Um zu verhindern, dass Clientanwendungen zu häufig neu verteilt werden, insbesondere wenn Anwendungsverbindungen eingehen und gehen, werden Grenzwerte dafür festgelegt, wie oft der Uniform-Cluster die Neuverteilung von Clientanwendungen anfordert. Nach einer langen Verbindungs-oder Trennungsaktivität kann es einige Minuten dauern, bis die verbleibenden Anwendungsinstanzen gleichmäßig auf den Uniform-Cluster verteilt sind.

Weitere Informationen finden Sie unter Fehlerbehebung beim Anwendungsausgleich.

## Anwendungsaffinitäten

Nicht alle Anwendungen sind für die automatische Neuverteilung in einem einheitlichen Cluster geeignet. Nur Anwendungen, die **MQCNO_RECONNECT** angeben, werden neu verteilt. Anwendungen, die eine Affinität zu einem bestimmten Warteschlangenmanager haben, müssen entweder die Option **MQCNO_NO_RECONNECT** oder **MQCNO_RECONNECT_Q_MGR**angeben. Letzteres ermöglicht HA-Failover, aber keine Neuverteilung.

Beispiele für Anwendungen, die eine implizite Affinität zu einem Warteschlangenmanager erstellen:

- Anwendungen, die permanente Subskriptionen erstellen.
- Anwendungen, die permanente dynamische Warteschlangen erstellen, z. B. zum Empfangen von Antwortnachrichten.
- Anwendungen, die eine strikte Nachrichtenreihenfolge erwarten oder erfordern, dass alle Nachrichten in einer Folge von derselben Anwendungsinstanz oder beiden verarbeitet werden.

Diese Anwendungen müssen die Optionen **MQCNO_NO_RECONNECT** oder **MQCNO_RECONNECT_Q_MGR** anstelle von **MQCNO_RECONNECT**angeben.

Weitere Informationen finden Sie unter Optionen für Verbindungswiederherstellung.

## Nachrichtenverfügbarkeit

Während der Anwendungsausgleich Verbindungen um ausgefallene oder vorübergehend nicht verfügbare Warteschlangenmanager neu verteilen kann, replizieren Uniform-Cluster Nachrichtendaten nicht über ihre Mitglieder hinweg. Wenn ein Knoten ausfällt, muss für die Datenverfügbarkeit jedes Member des Uniform-Clusters ebenfalls hoch verfügbar sein. Viele Datenreplikations-und Hochverfügbarkeitslösungen sind verfügbar und können mit einheitlichen Clustern kombiniert werden, um maximale Service-und Datenverfügbarkeit zu erreichen, z. B.:

- Replizierter Speicher, der eine Containerinstanz unterstützt, die von der Containerorchestrierung automatisch erneut gestartet wird. Weitere Informationen finden Sie im Abschnitt Einzelner ausfallsicherer Warteschlangenmanager.
- RDQM-Warteschlangenmanager. Weitere Informationen finden Sie im Abschnitt RDQM-Hochverfügbarkeit.
- Multi-Instanz-Warteschlangenmanager. Weitere Informationen finden Sie im Abschnitt Multi-Instanz-Warteschlangenmanager.
- Native Hochverfügbarkeit. Weitere Informationen finden Sie unter Native HA.

- IBM MQ Appliance HA. Weitere Informationen finden Sie unter Hochverfügbarkeit.

## Skalierbarkeit und Leistung von Uniform-Clustern

Um eine engere Integration und gemeinsame Nutzung des Anwendungsstatus zwischen Warteschlangenmanagern in einem einheitlichen Cluster zu ermöglichen, ist eine höhere Ebene der übergreifenden Kommunikation erforderlich als in einem traditionellen IBM MQ -Cluster. Daher wird die Skalierung auf eine große Anzahl von Warteschlangenmanagern in einem einzelnen einheitlichen Cluster nicht empfohlen, weil sich die zusätzliche Kommunikation nachteilig auf die Leistung auswirkt.

Aus Leistungs-und Managementgründen ist es vorzuziehen, einen einheitlichen Cluster als einen einzigen traditionellen Warteschlangenmanager zu betrachten, der Messaging für eine Reihe zusammengehöriger Anwendungen bereitstellt, aber kein einziger Messaging-Service in einem Unternehmen ist. In diesem Muster sind kleine Zahlen von bis zu 10 Warteschlangenmanagern in der Regel ausreichend, um eine große Anzahl von Clientanwendungsverbindungen zu unterstützen. Der Anwendungsausgleich macht es einfach, mit kleinen Zahlen zu beginnen, z. B. 3 Warteschlangenmanager, und durch Hinzufügen weiterer Warteschlangenmanager ein Scale-up durchzuführen.

**Achtung:** Das Aktivieren des Verhaltens eines Uniform-Clusters in einem Cluster, der nicht über die empfohlenen Merkmale verfügt (insbesondere die Verwendung von Clustern mit einer großen Zahl an Warteschlangenmanagern) führt wahrscheinlich zu schwerwiegenden Leistungsproblemen.

**Zugehörige Konzepte**

„Automatischer Ausgleich von Anwendungen” auf Seite 444
Die Verteilung und Verfügbarkeit von Anwendungen durch den automatischen Anwendungsausgleich wird erheblich verbessert, indem ein IBM MQ-Uniform-Cluster aktiviert wird, mit dem die Anwendungsverteilung im gesamten Cluster genau verwaltet wird, anstatt auf eine beliebige Festlegung oder manuelle Fixierung von Anwendungen auf bestimmte Warteschlangenmanager angewiesen zu sein.

# ALW Uniform-Cluster erstellen

Mit der automatischen Konfiguration und der Unterstützung für automatisches Clustering können Sie sich sowohl die anfängliche Einrichtung eines Uniform-Clusters als auch die Synchronisierung der Mitgliederkonfigurationen dieses Clusters vereinfachen.

## Vorbereitende Schritte

Lesen Sie den Abschnitt „Einschränkungen und Überlegungen für Uniform-Cluster” auf Seite 456, bevor Sie einen Uniform-Cluster erstellen.

## Informationen zu diesem Vorgang

Sie geben an, dass ein bestimmter IBM MQ -Cluster als Uniform-Cluster behandelt werden soll, indem Sie in der Datei `qm.ini` einen Abschnitt für AutoCluster mit mindestens **Type**=*Uniform* und **ClusterName**=*< Uniform cluster name >*angeben.

Optional können Sie den zugrunde liegenden IBM MQ-Cluster mithilfe von *Automatische Clustererstellung* über die gleiche Zeilengruppe `.ini` konfigurieren. Zur Einrichtung Ihres Clusters mithilfe der Unterstützung für die automatische Clustererstellung stellen Sie eine Konfigurationsdatei bereit, die den Cluster und seine vollständigen Repositorys beschreibt.

Sofern der gestartete Warteschlangenmanager als eines der vollständigen Repositorys aufgelistet ist, wird er automatisch als vollständiges Repository konfiguriert. Ebenso werden bei der Definition der Clusterempfängerkanäle automatisch auch die Clustersenderkanäle zu dem oder den vollständigen Repositorys definiert.

## Prozedur

Damit Sie die zusätzlichen Funktionen nutzen können, für die ein Uniform-Cluster erforderlich ist, müssen Sie einen der folgenden Schritte ausführen:

- Konvertieren Sie einen vorhandenen Cluster in einen Uniform-Cluster, der mit dem im Abschnitt „Informationen zu Uniform-Clustern” auf Seite 442 beschriebenen Muster übereinstimmt.
- Erstellen Sie einen neuen Uniform-Cluster zu diesem Zweck.

### ALW *Neuen Uniform-Cluster erstellen*

Hier finden Sie Informationen, wie ein neuer Uniform-Cluster erstellt wird.

## Vorgehensweise

1. Erstellen Sie eine Datei, die beschreibt, wie der Cluster bzw. seine vollständigen Repositorys aussehen sollen.

   Wie bei jedem Cluster werden die Informationen zum Cluster zentral in zwei vollständigen Repositorys gespeichert.

   Insbesondere müssen Sie die Namen dieser beiden vollständigen Repositorys und deren Verbindungsnamen festlegen.

   **Anmerkung:** Diese Definitionen müssen vor jeder anderen Festlegung getroffen werden, auch vor der Erstellung der Warteschlangenmanager, die anschließend mit nachfolgendem Verfahren erstellt werden.

   Beispiel: Sie richten einen Uniform-Cluster mit dem Namen UNICLUS und den Warteschlangenmanagern QMA, QMB, QMC und QMD als Mitgliedern ein. QMA und QMB sind in diesem Beispiel die vollständigen Repositorys, während QMC und QMD Teilrepositorys sind. Eine Beispielkonfigurationsdatei, `uniclus.ini`:

   ```
   AutoCluster:
      Repository2Conname=QMA.dnsname(1414)
      Repository2Name=QMA
      Repository1Conname=QMB.dnsname(1414)
      Repository1Name=QMB
      ClusterName=UNICLUS
      Type=Uniform
   ```

   Die Felder **RepositoryNConname** werden als Attribut *Verbindungsname* für andere Clustermitglieder verwendet, um Clustersender (CLUSSDR) für sie zu definieren, und können eine Verbindungsliste für einen Multi-Instanz-Warteschlangenmanager sein und optional den Port enthalten.
2. Erstellen Sie eine Beispielkonfigurationsdatei `uniclus.mqsc`, die die MQSC-Definitionen enthält, die auf alle Cluster-Member angewendet werden sollen.

   In dieser Datei ist eine Zeile zwingend für die Definition eines Clusterempfängerkanals (CLUSRCVR) erforderlich. Diese enthält ein CLUSTER-Attribut für den automatisch erstellten Clusternamen (in der Regel durch die Einfügung +AUTOCL+) und einen Kanalnamen, der die Einfügung +QMNAME+ enthält.

   Diese Zeile beschreibt, wie andere Mitglieder des Uniform-Clusters Verbindungen zu den einzelnen Warteschlangenmanagern herstellen. Sie dient also quasi als Vorlage für die Verbindung mit den anderen Warteschlangenmanagern. Diese Definition kann zum Beispiel wie folgt aussehen:

   ```
   define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trptype(tcp)
   conname(+CONNAME+) cluster('+AUTOCL+') replace
   ```

   Bei der Konfiguration eines automatischen Clusters kann die Definition eines Clusterempfängerkanals in den Feldern CLUSTER, CONNAME und CHANNEL weitere Einfügungen enthalten. Diese Definitionen sind dann auf allen Warteschlangenmanagern des Uniform-Clusters identisch. Hierzu zählt:

   **+AUTOCL+**
   Der automatische Clustername

**+QMNAME+**
Der Name des zu erstellenden Warteschlangenmanagers

**+ KONNAME +**
Eine Variable, die während der Erstellung des Warteschlangenmanagers mit dem Parameter **-iv** oder in der Zeilengruppe ' `Variables` qm.ini' zur Verwendung in der Parameterzeichenfolge für den Verbindungsnamen definiert wird. Der Name der Variable kann ein beliebiger Wert sein.

Berücksichtigen Sie bei dieser Definition, dass Kanalnamen auf 20 Zeichen beschränkt sind. Sowohl der Wert mit den Einfügungen als auch der Wert nach der Ersetzung der Einfügungen muss dieser Einschränkung entsprechen. Der Inhalt der Datei kann beispielsweise wie folgt aussehen:

```
*##########################################################################
* Compulsory section for all uniform cluster queue managers
*##########################################################################
define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trptype(tcp) conname(+CONNAME+) clus
ter('+AUTOCL+') replace
*
*##########################################################################
* Configuration for all queue managers
*##########################################################################
define QL(APPQ) maxdepth(99999999) replace
define QL(APPQ2) maxdepth(99999999) replace
define channel(CLIENTCHL) chltype(svrconn) trptype(tcp) replace
```

3. Stellen Sie diese beiden Dateien auf jeder Maschine zur Verfügung, auf der ein Mitglied des Uniform-Clusters bereitgestellt wird.

   Beispiel: `/shared/uniclus.ini` und `/shared/uniclus.mqsc`.
4. Erstellen Sie auf jeder dieser Maschinen einen Warteschlangenmanager.

   Geben Sie hierzu in der Befehlszeile Folgendes ein:

   a. Eine Anforderung zum Starten eines Listeners am erwarteten Port
   b. Eine Anforderung für die automatische INI-Konfiguration (**-ii**), die auf die Datei für die automatische Clusterkonfiguration (uniclus.ini) verweist.
   c. Eine Anforderung zur automatischen MQSC-Konfiguration (**-ic**), die auf die MQSC-Konfigurationsdatei verweist, die eine CLUSRCVR-Definition für den Uniform-Cluster enthält.
   d. Ein CONNAME für diesen Warteschlangenmanager.

   Auf dem Host von QMA:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv CONNAME=QMA.dnsna
me(1414) QMA
strmqm QMA
```

   Jeder Warteschlangenmanager im Uniform-Cluster wird mit einer fast identischen Befehlszeile erstellt - alle Unterschiede zwischen vollständigem Repository und Teilrepository werden für ein Uniform-Cluster automatisch verarbeitet.

   Auf dem Host von QMB:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv CONNAME=QMB.dnsna
me(1414) QMB
strmqm QMB
```

   Auf dem Host von QMC:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv CONNAME=QMC.dnsna
me(1414) QMC
strmqm QMC
```

   Auf dem Host von QMD:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv CONNAME=QMD.dnsna
me(1414) QMD
strmqm QMD
```

Die folgenden Konfigurationen erfolgen automatisch:

Wenn der Warteschlangenmanager gestartet wird, werden die Definitionen aus der `uniclus.ini`-Datei auf die `qm.ini`-Datei angewendet. Weitere Informationen finden Sie unter „Automatische Konfiguration von 'qm.ini' beim Start” auf Seite 105. Dadurch wird die Definition **AutoCluster** zur Datei `qm.ini` hinzugefügt.

Wenn der Warteschlangenmanager in der Zeilengruppe **AutoCluster** als eines der vollständigen Repositorys benannt ist, wird er automatisch in ein vollständiges Repository konvertiert, ähnlich wie der MQSC-Befehl ALTER QMGR REPOS (**ClusterName**). Andernfalls wird er in ein Teilrepository konvertiert, ähnlich wie der MQSC-Befehl ALTER QMGR REPOS ('').

Wenn die Definition des Clusterempfängerkanals für den automatischen Cluster verarbeitet wird, werden Clustersenderkanäle von diesem Warteschlangenmanager zu allen vollständigen Repositorys in der Zeilengruppe **AutoCluster** definiert (mit Ausnahme des lokalen Warteschlangenmanagers, wenn dies eines der vollständigen Repositorys ist). Diese Senderkanäle übernehmen alle allgemeinen Kanalattribute des definierten lokalen Clusterempfängers.

**Achtung:** Obwohl die Kanäle ohne manuelle Eingriffe erstellt werden, handelt es sich um Verwaltungsobjekte, die wie jede andere Kanaldefinition angezeigt und verwaltet werden können. Sie sollten diese Objekte nicht mit 'automatisch definierten' Clustersenderkanälen verwechseln, die vom Cluster transient und bei Bedarf erstellt werden, um den Nachrichtenverkehr weiterzuleiten.

## Nächste Schritte

### Einrichtung des Uniform-Clusters überprüfen

Wenn der Parameter **ClusterName** ordnungsgemäß festgelegt ist und der Warteschlangenmanager Mitglied des benannten Clusters ist, wird die Nachricht AMQ9883 ausgegeben, um zu bestätigen, dass der Cluster jetzt als Uniform-Cluster identifiziert wird.

Sie können jetzt die Funktionen des Uniform-Clusters wie den automatischen Anwendungsausgleich verwenden. Wenn dieser Parameter beim Warteschlangenmanagerstart festgelegt wurde, der Name aber kein gültiger IBM MQ-Clustername ist, wird der Name ignoriert und die Fehlernachricht AMQ9882 ausgegeben.

Wenn es sich beim Namen und einen gültigen Clusternamen handelt, aber keine Clusterkanäle für den angegebenen Cluster vorhanden sind, wird die Warnung AMQ9881 im Fehlerprotokoll des Warteschlangenmanagers ausgegeben, damit Ihr Administrator diese Situation erkennen und korrigieren kann.

### Einrichtung des automatischen Clusters überprüfen

Wenn Sie die Unterstützung für die automatische Clustererstellung für die Einrichtung des Uniform-Clusters verwendet haben, können Sie mit runmqsc-Befehlen überprüfen, ob die als vollständige Repositorys definierten Warteschlangenmanager korrekt konfiguriert wurden:

```
QMA:
     1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(QMA)                             REPOS(UNICLUS)
```

Teilrepositorys sind dagegen nicht als Repositorys konfiguriert:

```
QMC:
     1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(QMC)                             REPOS( )
```

Darüber hinaus sollten die Clustersenderkanäle (CLUSSDR) von jedem Warteschlangenmanager zu den anderen vollständigen Repositorys mit dem Kanalnamen aus der MQSC-Konfigurationsdatei konfiguriert worden sein:

```
QMA:
     1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMA)                    CHLTYPE(CLUSRCVR)
   CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMB)                    CHLTYPE(CLUSSDR)
   CONNAME(QMB.dnsname(1414))

QMC:
     1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMA)                    CHLTYPE(CLUSSDR)
   CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMB)                    CHLTYPE(CLUSSDR)
   CONNAME(QMB.dnsname(1414))
AMQ8414I: Display Channel details.
   CHANNEL(UNICLUS_QMC)                    CHLTYPE(CLUSRCVR)
   CONNAME(QMC.dnsname(1414))
```

**Zugehörige Konzepte**

„Informationen zu Uniform-Clustern” auf Seite 442
Eine Uniform-Clusterimplementierung hat das Ziel, dass Anwendungen im Hinblick auf Umfang und Verfügbarkeit entwickelt werden und eine Verbindung zu jedem Warteschlangenmanager im Uniform-Cluster herstellen können. Dadurch sind Anwendungen nicht von einem bestimmten Warteschlangenmanager abhängig, was zu einer verbesserten Verfügbarkeit und einem besseren Lastausgleich im Messaging-Verkehr führt. z/OS Unter IBM MQ for z/OS sind keine Uniform-Cluster verfügbar. Viele Funktionen eines Uniform-Clusters werden dort von Gruppen mit gemeinsamer Warteschlange übernommen.

„Einschränkungen und Überlegungen für Uniform-Cluster” auf Seite 456
Einschränkungen und Überlegungen bei der Konfiguration von Uniform-Clustern.

## ALW *Vorhandenen Cluster in einem Uniform-Cluster konvertieren*

Mit dieser Prozedur können Sie einen vorhandenen Cluster in einen Uniform-Cluster konvertieren.

### Informationen zu diesem Vorgang

Wenn Sie einen vorhandenen Cluster in einen Uniform-Cluster konvertieren, müssen Sie sicherstellen, dass alle Definitionen, die für die Verteilung der Anwendungslast auf die im Uniform-Cluster vorhandenen Warteschlangenmanager erforderlich sind, auf allen Clustermitgliedern vorhanden sind.

### Vorgehensweise

1. Aktivieren Sie IBM MQ-Publish/Subscribe, einschließlich geclustertem Publish/Subscribe, auf allen Warteschlangenmanagern.

   Dies ist eine Voraussetzung für die Uniform-Cluster-Funktion. Sie müssen daher sicherstellen, dass die Attribute PSMODE und PSCLUS auf dem Warteschlangenmanager, wie in der Standardeinstellung vorgesehen, auf ENABLED gesetzt sind.
2. Fügen Sie der Datei `qm.ini` einen Abschnitt **`AutoCluster`** zum Namen des IBM MQ -Clusters hinzu, wie er in Ihren MQSC-Objektdefinitionen wie Clusterkanälen verwendet wird.

   Wenn der Name des Clusters beispielsweise UNICLUS ist, fügen Sie die Zeilengruppe "AutoCluster" in Ihren `qm.ini`-Dateien wie folgt hinzu oder ändern Sie sie:

   ```
   AutoCluster:
         ClusterName=UNICLUS
       Type=Uniform
   ```

3. Starten Sie die Warteschlangenmanager erneut, um die neue Einstellung anzuwenden.
4. Die automatische Konfiguration ist als ein Mechanismus zu betrachten, durch den sichergestellt wird, dass alle Mitglieder eines Uniform-Clusters vom Start an die gleiche Konfiguration aufweisen.

   Weitere Informationen finden Sie im Abschnitt Automatische Konfiguration aus einem MQSC-Script beim Start.

**Zugehörige Konzepte**

„Informationen zu Uniform-Clustern" auf Seite 442
Eine Uniform-Clusterimplementierung hat das Ziel, dass Anwendungen im Hinblick auf Umfang und Verfügbarkeit entwickelt werden und eine Verbindung zu jedem Warteschlangenmanager im Uniform-Cluster herstellen können. Dadurch sind Anwendungen nicht von einem bestimmten Warteschlangenmanager abhängig, was zu einer verbesserten Verfügbarkeit und einem besseren Lastausgleich im Messaging-Verkehr führt. z/OS Unter IBM MQ for z/OS sind keine Uniform-Cluster verfügbar. Viele Funktionen eines Uniform-Clusters werden dort von Gruppen mit gemeinsamer Warteschlange übernommen.

„Einschränkungen und Überlegungen für Uniform-Cluster" auf Seite 456
Einschränkungen und Überlegungen bei der Konfiguration von Uniform-Clustern.

## Multi Automatische Clusterkonfiguration verwenden

Sie konfigurieren IBM MQ, um die automatische Konfiguration zu aktivieren, indem Sie die `qm.ini`-Konfigurationsinformationen ändern.

**Anmerkung:** Die Zeilengruppe 'AutoCluster' kann nur für Uniform-Cluster verwendet werden.

### Zu konfigurierende Zeilengruppen

Sie können folgende Zeilengruppen ändern:

**AutoConfig**
: Wird in der Datei `qm.ini` definiert. Aus dieser Zeilengruppe wird beim Start eines Warteschlangenmanagers ermittelt, welche Dateien für die automatische Konfiguration verwendet werden sollen.

  Sie sollten sich dieses Mechanismus bedienen, um sicherzustellen, dass innerhalb eines Uniform-Clusters identische Clusterkonfigurationen verwendet werden.

**AutoCluster**
: Wird in der Datei `qm.ini` definiert. Aus dieser Zeilengruppe wird beim Start eines Warteschlangenmanagers ermittelt, ob der Cluster ein Mitglied eines automatischen Clusters ist. Diese Zeilengruppe kann auch die vollständigen Repositorys des Clusters festlegen.

**Variablen**
: Wird in der Datei `qm.ini` definiert. Diese Zeilengruppe enthält einige Variablen für Warteschlangenmanager.

### Attribute für die Zeilengruppe 'AutoConfig'

Die folgenden zwei Attribute sind in der Zeilengruppe 'AutoConfig' zulässig:

**MQSCConfig=*<Path>***
: Der Pfad ist entweder ein vollständiger Dateipfad oder Pfad zu einem Verzeichnis, in dem alle Dateien `*.mqsc` auf jedem Warteschlangenmanager-Start auf den Warteschlangenmanager angewendet werden.

  Weitere Informationen finden Sie im Abschnitt Automatische Konfiguration aus einem MQSC-Script beim Start.

**IniConfig=*<Path>***
: Der Pfad ist entweder ein vollständiger Dateipfad oder Pfad zu einem Verzeichnis, in dem alle Dateien `*.ini` auf jedem Warteschlangenmanager-Start auf die `qm.ini`-Datei angewendet werden.

  Weitere Informationen finden Sie unter „Automatische Konfiguration von 'qm.ini' beim Start" auf Seite 105.

Diese Attribute werden häufig zur Einrichtung von Uniform-Clustern verwendet. Weitere Informationen finden Sie unter „Neuen Uniform-Cluster erstellen” auf Seite 459.

Beispiel für die Zeilengruppe:

```
AutoConfig:
MQSCConfig=C:\MQ_Configuration\uniclus.mqsc
IniConfig=C:\MQ_Configuration\uniclus.ini
```

## Attribute für die Zeilengruppe 'AutoCluster'

Die folgenden Attribute sind für die Zeilengruppe 'AutoCluster' obligatorisch:

**Type=*Uniform***
Gibt den Typ des automatischen Clusters an. Die einzige gültige Option ist *Uniform*, was für einen Uniform-Cluster steht.

**ClusterName=*<String>***
Name des Clusters, bei dem es sich um den Namen des automatischen Clusters handelt.

Die oben genannten Attribute ermöglichen die Verteilung der Anwendungslast innerhalb des Uniform-Clusters. Weitere Informationen finden Sie im Abschnitt „Automatischer Ausgleich von Anwendungen” auf Seite 444.

Zudem lässt sich ein Cluster leichter einrichten, wenn der Cluster in dieser Zeilengruppe beschrieben ist. Weitere Informationen finden Sie unter „Neuen Uniform-Cluster erstellen” auf Seite 459. Bei Verwendung dieser Funktion können Sie zwei Warteschlangenmanager mit ihren Verbindungsnamen für die vollständigen Repositorys dieses automatischen Clusters angeben.

Die folgenden Attribute sind für die Zeilengruppe 'AutoCluster' optional, müssen aber paarweise angegeben werden:

**RepositoryName1 =*< Zeichenfolge>***
Dies ist der Warteschlangenmanagername für das erste vollständige Repository im automatischen Cluster. Es kann der Name dieses Warteschlangenmanagers oder ein anderer Name sein.

**Repository1Conname=*< Verbindungsnamenszeichenfolge>***
Dies ist der Verbindungsname (CONNAME). Dieser Wert gibt an, wie die Verbindung zwischen den Mitgliedern des automatischen Clusters und diesem Warteschlangenmanager hergestellt wird.

Mit den folgenden Attributen können Sie ein zweites vollständiges Repository für den Cluster angeben:

**Repository2Name=***< Zeichenfolge>*
**Repository2Conname=***< Verbindungsnamenszeichenfolge>*

Beispiel für die Zeilengruppe:

```
AutoCluster:
   Repository2Conname=myFR1.hostname(1414)
   Repository2Name=QMFR1
   Repository1Conname= myFR2.hostname(1414)
   Repository1Name=QMFR2
   ClusterName=UNICLUS
   Type=Uniform
```

## Attribute für die Zeilengruppe 'Variables'

Im Attributfeld ist ein `attribute=value`-Paar zulässig. Dieses kann bei der Erstellung eines Warteschlangenmanagers mit der Befehlszeilenoption **-iv** des Befehls **crtmqm** bereitgestellt werden.

Sie können Attribute, die in der Zeilengruppe 'Variables' aufgelistet sind, während der automatischen Clusterkonfiguration von CONNAME und der MQSC-Felder des Kanalnamens eines Clusterempfängerkanals verwenden.

### Warteschlangenmanager aus einem Uniform-Cluster aussetzen

Während des normalen Betriebs eines Uniform-Clusters können wiederverbindbare Clientanwendungsinstanzen jederzeit automatisch auf jeden Warteschlangenmanager im Cluster verteilt werden. Mit dem Befehl SUSPEND QMGR können Sie verhindern, dass Anwendungen für einen bestimmten Zeitraum eine Verbindung zu einem bestimmten Warteschlangenmanager herstellen, beispielsweise während Wartungsoperationen oder bei der Problembestimmung.

Geben Sie den Befehl SUSPEND QMGR CLUSTER(*Uniform-Clustername*) aus.

Zusätzlich zu den üblichen Auswirkungen des Aussetzens aus einem IBM MQ -Cluster in einem einheitlichen Cluster verhindert der Befehl SUSPEND auch, dass wiederverbindbare Anwendungen auf diesen Warteschlangenmanager neu verteilt werden.

Wenn der Befehl ausgegeben wird, werden alle vorhandenen Verbindungen zum Warteschlangenmanager sofort neu auf andere verfügbare Warteschlangenmanager im Cluster verteilt.

**Anmerkungen:**

- Wenn Warteschlangenmanager in einem Cluster ausgesetzt werden, zeigt DIS APSTATUS sie als ACTIVE (NO) an, mit Ausnahme des lokalen Warteschlangenmanagers, der immer ACTIVE (YES) für seinen eigenen Statuseintrag anzeigt.
- Wenn alle Warteschlangenmanager im Cluster ausgesetzt werden, bleiben Anwendungen mit einem oder mehreren der ausgesetzten Warteschlangenmanagern verbunden.

Um zu vermeiden, dass dem zu verwaltenden Warteschlangenmanager neue Verbindungen hinzugefügt werden, sollten Sie den oder die Serververbindungskanäle, die von Ihren Clientanwendungen verwendet werden, stoppen, indem Sie beispielsweise den folgenden **runmqsc** -Befehl absetzen:

```
STOP CHANNEL(svrconn channel name)
```

Dies ist unter Umständen nicht möglich, wenn diese Kanäle beispielsweise auch zur Verbindung mit Verwaltungsanwendungen verwendet werden, die im Wartungsfenster erforderlich sind. Aus diesem Grund prüft der ausgesetzte Warteschlangenmanager regelmäßig, ob verbundene wiederverbindbare Anwendungen vorhanden sind.

Wenn wiederverbindbare Anwendungen vorhanden sind, werden sie auf andere verfügbare Warteschlangenmanager im Cluster neu verteilt. Die Wartung kann nun im ausgesetzten Warteschlangenmanager ausgeführt werden.

**Anmerkung:** Anwendungen, die nicht als verschiebbar angesehen werden, sind nicht vom ursprünglichen Befehl oder den nachfolgenden Suchen betroffen und die Verbindung zum ausgesetzten Warteschlangenmanager bleibt bestehen ( im Abschnitt MOVCOUNT finden Sie weitere Einzelheiten).

So setzen Sie einen ausgesetzten Warteschlangenmanager fort:

1. Starten Sie bei Bedarf den Serververbindungskanal, um das Akzeptieren neuer Anwendungsverbindungen fortzusetzen, indem Sie den folgende Befehl ausgeben:

   ```
   START CHANNEL(svrconn channel name)
   ```

2. Geben Sie den Befehl **runmqsc** aus:

   ```
   RESUME QMGR CLUSTER(uniform cluster name)
   ```

Der Warteschlangenmanager nimmt die Kommunikation mit dem Rest des einheitlichen Clusters wieder auf und, falls erforderlich, werden wiederherstellbare Clientanwendungsinstanzen an diesen Warteschlangenmanager umgeleitet.

## Publish/Subscribe-Messaging konfigurieren

Sie können den Status des eingereihten Publish/Subscribe starten, stoppen und anzeigen. Darüber hinaus können Sie Datenströme hinzufügen und entfernen sowie Warteschlangenmanager aus einer Brokerhierarchie hinzufügen und löschen.

### Prozedur

- Weitere Informationen zum Steuern von eingereihtem Publish/Subscribe finden Sie in den folgenden Unterabschnitten:
  - „Publish/Subscribe-Nachrichtenattribute in der Warteschlange festlegen” auf Seite 466
  - „In Warteschlange eingereihtes Publish/Subscribe starten” auf Seite 467
  - „In Warteschlange eingereihtes Publish/Subscribe stoppen” auf Seite 468
  - „Datenstrom hinzufügen” auf Seite 468
  - „Datenstrom löschen” auf Seite 469
  - „Einen Subskriptionspunkt hinzufügen” auf Seite 470
  - „Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen” auf Seite 479

## Publish/Subscribe-Nachrichtenattribute in der Warteschlange festlegen

Sie steuern das Verhalten einiger Publish/Subscribe-Nachrichtenattribute mit Hilfe von WS-Managerattributen. Die anderen Attribute, die Sie steuern, werden in der Zeilengruppe *Broker* der `qm.ini`-Datei gesteuert.

### Informationen zu diesem Vorgang

Sie können die folgenden Publish/Subscribe-Attribute festlegen: Details hierzu finden Sie unter Parameter des Warteschlangenmanagers .

*Tabelle 27. Publish/Subscribe-Konfigurationsparameter*

| **Beschreibung** | **MQSC-Parametername** |
|---|---|
| Wiederholungszähler für Befehlsnachricht | **PSRTYCNT** |
| Unzustellbare Befehlseingabenachricht verwerfen | **PSNPMSG** |
| Verhalten nach unzustellbarer Befehlsantwortnachricht | **PSNPRES** |
| Verarbeiten von Befehlsnachrichten unter Synchronisationspunkt | **PSSYNCPT** |

Die Zeilengruppe 'Broker' wird verwendet, um die folgenden Konfigurationseinstellungen zu verwalten:

- `PersistentPublishRetry = yes | force`

  Wenn Sie `Ja` angeben, schlägt eine Veröffentlichung einer persistenten Nachricht über die Publish/Subscribe-Schnittstelle in der Warteschlange fehl, und es wurde keine negative Antwort angefordert, wird die Veröffentlichungsoperation erneut versucht.

  Wenn Sie eine negative Antwortnachricht angefordert haben, wird die negative Antwort gesendet, und es findet keine weitere Wiederholung statt.

  Wenn Sie `Erzwingen` angeben, wird die Veröffentlichungsoperation, wenn eine Veröffentlichung einer persistenten Nachricht über die Publish/Subscribe-Schnittstelle in der Warteschlange fehlschlägt, erneut versucht, bis sie erfolgreich verarbeitet wurde. Es wird keine negative Antwort gesendet.
- `NonPersistentPublishRetry = yes | force`

  Wenn Sie `Ja` angeben, schlägt eine Veröffentlichung einer nicht persistenten Nachricht über die Publish/Subscribe-Schnittstelle in der Warteschlange fehl, und es wurde keine negative Antwort angefordert, wird die Veröffentlichungsoperation erneut versucht.

  Wenn Sie eine negative Antwortnachricht angefordert haben, wird die negative Antwort gesendet, und es findet keine weitere Wiederholung statt.

Wenn Sie Erzwingen angegeben haben und eine Veröffentlichung einer nicht persistenten Nachricht über die Publish/Subscribe-Schnittstelle in der Warteschlange fehlschlägt, wird die Veröffentlichungsoperation so lange erneut versucht, bis sie erfolgreich verarbeitet wurde. Es wird keine negative Antwort gesendet.

**Anmerkung:** Wenn Sie diese Funktionalität für nicht persistente Nachrichten aktivieren möchten, müssen Sie außerdem den Wert für NonPersistentPublishRetry festlegen, um sicherzustellen, dass das WS-Managerattribut **PSSYNCPT** auf Yes gesetzt ist.

Dies kann sich auch auf die Leistung der Verarbeitung nicht persistenter Veröffentlichungen auswirken, da die **MQGET** aus der STREAM-Warteschlange jetzt unter einem Synchronisationspunkt auftritt.

- PublishBatchSize= *number*

  Der Broker verarbeitet normalerweise Veröffentlichungsnachrichten innerhalb eines Synchronisationspunkts. Es kann ineffizient sein, jede Veröffentlichung einzeln festzuschreiben, und unter bestimmten Umständen kann der Broker mehrere Publizierungsnachrichten in einer einzigen UO- Einheit verarbeiten. Dieser Parameter gibt die maximale Anzahl der Publizierungsnachrichten an, die in einer einzigen UO- Unit verarbeitet werden können.

  Der Standardwert für PublishBatchSize ist 5.

- PublishBatchInterval= *number*

  Der Broker verarbeitet normalerweise Veröffentlichungsnachrichten innerhalb eines Synchronisationspunkts. Es kann ineffizient sein, jede Veröffentlichung einzeln festzuschreiben, und unter bestimmten Umständen kann der Broker mehrere Publizierungsnachrichten in einer einzigen UO- Einheit verarbeiten. Dieser Parameter gibt die maximale Zeit (in Millisekunden) zwischen der ersten Nachricht in einem Stapel und einer nachfolgenden Veröffentlichung an, die in demselben Stapel enthalten ist.

  Ein Stapelintervall von 0 zeigt an, dass bis zu PublishBatchSize -Nachrichten verarbeitet werden können, sofern die Nachrichten sofort verfügbar sind.

  Der Standardwert für PublishBatchInterval ist null.

### Vorgehensweise

Verwenden Sie den IBM MQ Explorer, programmierbare Befehle oder den Befehl **runmqsc**, um die Warteschlangenmanagerattribute zu ändern, mit denen das Publish/Subscribe-Verhalten gesteuert wird.

#### Beispiel

ALTER QMGR PSNPRES (SAFE)

## In Warteschlange eingereihtes Publish/Subscribe starten

Sie starten die Publish/Subscribe-Warteschlange in der Warteschlange, indem Sie das Attribut PSMODE des Warteschlangenmanagers festlegen.

### Vorbereitende Schritte

Lesen Sie die Beschreibung von PSMODE , um die drei Modi von Publish/Subscribe zu verstehen:

- COMPAT
- Inaktiviert
- Aktiviert

### Informationen zu diesem Vorgang

Legen Sie das Attribut QMGR PSMODE fest, um entweder die Publish/Subscribe-Schnittstelle in der Warteschlange (auch als Broker bezeichnet) oder die Publish/Subscribe-Steuerkomponente (auch als Publish/Subscribe der Version 7 bezeichnet) zu starten. Um die Publish/Subscribe-Warteschlange in die Warteschlange zu starten, müssen Sie PSMODE auf ENABLED setzen. Der Standardwert ist ENABLED .

### Vorgehensweise

Verwenden Sie den IBM MQ Explorer oder den Befehl **runmqsc**, um die Publish/Subscribe-Schnittstelle in der Warteschlange zu aktivieren, wenn die Schnittstelle noch nicht aktiviert ist.

#### Beispiel

```
ALTER QMGR PSMODE (ENABLED)
```

### Nächste Schritte

IBM MQ verarbeitet in die Warteschlange eingereihte Publish/Subscribe-Befehle und -MQI-Aufrufe (Publish/Subscribe Message Queue Interface).

## In Warteschlange eingereihtes Publish/Subscribe stoppen

Sie können die eingereihte Publish/Subscribe-Funktion stoppen, indem Sie das Attribut PSMODE des Warteschlangenmanagers festlegen.

### Vorbereitende Schritte

Lesen Sie die Beschreibung von PSMODE , um die drei Modi von Publish/Subscribe zu verstehen:

- COMPAT
- INAKTIVIERT
- ENABLED

### Informationen zu diesem Vorgang

Setzen Sie das Attribut QMGR PSMODE , um die Publish/Subscribe-Schnittstelle in der Warteschlange (auch als Broker bezeichnet) oder die Publish/Subscribe-Steuerkomponente (auch bekannt als Publish/Subscribe der Version 7) oder beides zu stoppen. Um die Publish/Subscribe-Warteschlange in der Warteschlange zu stoppen, müssen Sie PSMODE auf COMPAT setzen. Wenn Sie die Publish/Subscribe-Steuerkomponente vollständig stoppen möchten, setzen Sie PSMODE auf DISABLED .

### Vorgehensweise

Verwenden Sie den IBM MQ Explorer oder den Befehl **runmqsc**, um die Publish/Subscribe-Schnittstelle in der Warteschlange zu inaktivieren.

#### Beispiel

```
ALTER QMGR PSMODE (COMPAT)
```

## Datenstrom hinzufügen

Sie können Datenströme manuell hinzufügen, um eine Datenisolation zwischen Anwendungen zu ermöglichen oder die Interoperation mit Publish/Subscribe-Hierarchien von IBM MQ zu ermöglichen.

### Vorbereitende Schritte

Machen Sie sich mit der Funktionsweise von Publish/Subscribe-Streams vertraut. Siehe Streams und Themen.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte mit dem PCF-Befehl **runmqsc** oder in IBM MQ Explorer aus.

**Anmerkung:** Sie können die Schritte 1 und 2 in beliebiger Reihenfolge ausführen. Führen Sie Schritt 3 nur aus, wenn die Schritte 1 und 2 beide abgeschlossen sind.

## Vorgehensweise

1. Definieren Sie eine lokale Warteschlange mit demselben Namen wie der Datenstrom in der früheren Version von IBM MQ.
2. Definieren Sie ein lokales Thema mit demselben Namen wie der Datenstrom in einer früheren Version von IBM MQ.
3. Fügen Sie den Namen der Warteschlange zur Namensliste hinzu: SYSTEM.QPUBSUB.QUEUE.NAME-LIST
4. Wiederholen Sie diese Schritte für alle Warteschlangenmanager der höheren Version von IBM MQ , die sich in der Publish/Subscribe-Hierarchie befinden.

#### Wird hinzugefügt 'Sport'

Im Beispiel für die gemeinsame Nutzung des Datenstroms 'Sport' arbeiten die Warteschlangenmanager der früheren Version und der neueren Version IBM MQ in derselben Publish/Subscribe-Hierarchie. Die Warteschlangenmanager der früheren Version nutzen einen Datenstrom namens 'Sport' gemeinsam. Das Beispiel zeigt, wie eine Warteschlange und ein Thema auf Warteschlangenmanagern der höheren Version mit dem Namen 'Sport' mit einer Themenzeichenfolge 'Sport' erstellt werden, die mit dem Datenstrom 'Sport' der Warteschlangenmanager der früheren Version gemeinsam genutzt wird.

Eine Veröffentlichungsanwendung des Warteschlangenmanagers einer höheren Version, die Veröffentlichung zum Thema 'Sport' mit der Themenzeichenfolge 'Soccer/Results', erstellt die resultierende Themenzeichenfolge 'Sport/Soccer/Results'. Auf den Warteschlangenmanagern der höheren Version empfangen Subskribenten des Themas 'Sport' mit der Themenzeichenfolge 'Soccer/Results' die Veröffentlichung.

Auf Warteschlangenmanagern früherer Versionen erhalten Subskribenten des Datenstroms 'Sport' mit der Themenzeichenfolge 'Soccer/Results' die Veröffentlichung.

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
     1 : define qlocal('Sport')
AMQ8006: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
     2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM', 'SYSTEM.BRO
KER.ADMIN.STREAM')
     3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM', 'SYS
TEM.BROKER.ADMIN.STREAM')
AMQ8551: IBM MQ namelist changed.
```

**Anmerkung:** Sie müssen sowohl die vorhandenen Namen im Namenslistenobjekt als auch die neuen Namen, die Sie hinzufügen, im Befehl **alter namelist** angeben.

## Nächste Schritte

Informationen über den Datenstrom werden an andere Broker in der Hierarchie übergeben.

Sie müssen jeden IBM MQ -Warteschlangenmanager in der Hierarchie manuell konfigurieren.

# Datenstrom löschen

Sie können einen Datenstrom aus einem IBM MQ -Warteschlangenmanager löschen.

## Vorbereitende Schritte

Bevor Sie einen Datenstrom löschen, müssen Sie sicherstellen, dass keine weiteren Subskriptionen für den Datenstrom vorhanden sind, und alle Anwendungen, die den Datenstrom verwenden, in den Wartemodus versetzen. Wenn Veröffentlichungen weiterhin in einen gelöschten Datenstrom fließen, ist hoher Verwaltungsaufwand notwendig, um den einwandfreien Betriebsstatus des Systems wiederherzustellen.

### Vorgehensweise

1. Suchen Sie nach allen verbundenen Brokern, die diesen Datenstrom hosten.
2. Brechen Sie alle Subskriptionen für den Datenstrom auf allen Brokern ab.
3. Entfernen Sie die Warteschlange (mit dem gleichen Namen wie der Datenstrom) aus der Namensliste SYSTEM.QPUBSUB.QUEUE.NAMELIST.
4. Löschen oder bereinigen Sie alle Nachrichten aus der Warteschlange mit demselben Namen wie der Datenstrom.
5. Löschen Sie die Warteschlange mit demselben Namen wie der Datenstrom.
6. Löschen Sie das zugeordnete Themenobjekt.

### Nächste Schritte

Wiederholen Sie die Schritte 3 bis 5 auf allen anderen verbundenen IBM MQ -Warteschlangenmanagern, die den Datenstrom hosten.

## Einen Subskriptionspunkt hinzufügen

Vorgehensweise zum Erweitern einer vorhandenen Publish/Subscribe-Anwendung in der Warteschlange, die Sie von einer früheren Version von IBM Integration Bus mit einem neuen Subskriptionspunkt migriert haben.

### Vorbereitende Schritte

1. Stellen Sie sicher, dass der Subskriptionspunkt nicht bereits in SYSTEM.QPUBSUB.SUBPOINT.NAMELIST definiert ist.
2. Überprüfen Sie, ob ein Themenobjekt oder eine Themenzeichenfolge mit dem gleichen Namen wie der Subskriptionspunkt vorhanden ist.

### Informationen zu diesem Vorgang

IBM MQ, Anwendungen verwenden keine Subskriptionspunkte, aber sie können mit vorhandenen Anwendungen interagieren, die unter Verwendung des Migrationsmechanismus für Subskriptionspunkte arbeiten.

**Wichtig:** Der Migrationsmechanismus für Subskriptionspunkte wurde aus IBM MQ 8.0 entfernt. Wenn Sie Ihre vorhandenen Anwendungen migrieren müssen, müssen Sie die in der Dokumentation beschriebenen Prozeduren für Ihre Version des Produkts ausführen, bevor Sie die Migration auf die neueste Version durchführen.

Es ist nicht erforderlich, Subskriptionspunkte hinzuzufügen, um integrierte Publish/Subscribe-Anwendungen zu nutzen, die für Versionen von IBM MQgeschrieben wurden.

### Vorgehensweise

1. Fügen Sie den Namen des Subskriptionspunkts zu SYSTEM.QPUBSUB.SUBPOINT.NAMELIST hinzu.
   - Unter z/OS ist **NLTYPE** standardmäßig auf NONE gesetzt.
   - Wiederholen Sie den Schritt auf jedem Warteschlangenmanager, der in derselben Publish/Subscribe-Topologie verbunden ist.
2. Fügen Sie ein Themenobjekt hinzu, vorzugsweise geben Sie ihm den Namen des Subskriptionspunkts mit einer Themenzeichenfolge an, die mit dem Namen des Subskriptionspunkts übereinstimmt.
   - Wenn sich der Subskriptionspunkt in einem Cluster befindet, fügen Sie das Themenobjekt als Clusterthema auf dem Clusterthemenhost hinzu.
   - Wenn ein Themenobjekt mit der gleichen Themenzeichenfolge wie der Name des Subskriptionspunkts vorhanden ist, verwenden Sie das vorhandene Themenobjekt. Sie müssen die Auswirkungen des Subskriptionspunkts kennen, der ein vorhandenes Thema wiederverwendet. Wenn das vorhan-

dene Thema Teil einer vorhandenen Anwendung ist, müssen Sie die Kollision zwischen zwei identisch benannten Themen auflösen.

- Wenn ein Themenobjekt mit demselben Namen wie der Subskriptionspunkt vorhanden ist, aber eine andere Themenzeichenfolge enthält, erstellen Sie ein Thema mit einem anderen Namen.

3. Setzen Sie das Attribut **Topic** WILDCARD auf den Wert BLOCK.

   Wenn Sie Abonnements für # oder * blockieren, werden Platzhalterzeichen für Abonnements für Subskriptionspunkte isoliert, siehe Wildcards und Subskriptionspunkte.

4. Legen Sie alle Attribute fest, die Sie im Themenobjekt benötigen.

**Beispiel**

Das Beispiel zeigt eine **runmqsc**-Befehlsdatei, die zwei Subskriptionspunkte hinzufügt, USD und GBP.

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

**Anmerkung:**

1. Schließen Sie den Standardsubskriptionspunkt in die Liste der Subskriptionspunkte ein, die mit dem Befehl **ALTER** hinzugefügt wurden. **ALTER** löscht vorhandene Namen in der Namensliste.
2. Definieren Sie die Themen, bevor Sie die Namensliste ändern. Der Warteschlangenmanager prüft nur die Namensliste, wenn der Warteschlangenmanager gestartet wird und wenn die Namensliste geändert wird.

# Verteilte Publish/Subscribe-Netze konfigurieren

Warteschlangenmanager, die miteinander in eine verteilte Publish/Subscribe-Topologie miteinander verbunden sind, nutzen gemeinsam einen gemeinsamen Topic-Bereich. Subskriptionen, die auf einem WS-Manager erstellt werden, können Nachrichten empfangen, die von einer Anwendung veröffentlicht wurden, die mit einem anderen Warteschlangenmanager in der Topologie verbunden ist.

Sie können den Umfang der Themenbereiche steuern, die durch die Verbindung von Warteschlangenmanagern in Clustern oder Hierarchien erstellt werden. In einem Publish/Subscribe-Cluster muss ein Themenobjekt für jeden Zweig des Topic-Bereichs, der sich über den Cluster erstrecken soll, 'clustered' (gruppiert) sein. In einer Hierarchie muss jeder WS-Manager so konfiguriert werden, dass er in der Hierarchie sein 'übergeordnetes Element' identifiziert.

Sie können den Fluss von Veröffentlichungen und Subskriptionen in der Topologie weiter steuern, indem Sie auswählen, ob jede Veröffentlichung und Subskription lokal oder global ist. Lokale Veröffentlichungen und Subskriptionen werden nicht außerhalb des Warteschlangenmanagers weitergegeben, mit dem der Bereitsteller oder Subskribent verbunden ist.

**Zugehörige Konzepte**

Verteilte Publish/Subscribe-Netzwerke

Veröffentlichungsumfang

Subskriptionsumfang

Themenbereiche

**Zugehörige Tasks**

Clusterthemen definieren

## Publish/Subscribe-Cluster konfigurieren

Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

## Vorbereitende Schritte

Einige Clusterkonfigurationen können die Overheads von Direct-Routing-Publish/Subscribe nicht aufnehmen. Bevor Sie diese Konfiguration verwenden, untersuchen Sie die Aspekte und Optionen, die in Publish/Subscribe-Cluster entwerfen beschrieben sind.

Damit Änderungen an einem Cluster im gesamten Cluster weitergegeben werden können, muss immer mindestens ein vollständiges Repository verfügbar sein. Stellen Sie sicher, dass Ihre Repositorys verfügbar sind, bevor Sie diese Task starten.

Siehe auch Routing für Publish/Subscribe-Cluster: Hinweise zum Verhalten .

Szenario:

- Der INVENTORY-Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen” auf Seite 339 beschrieben eingerichtet. Sie enthält drei Warteschlangenmanager: LONDON und NEWYORK beide enthalten vollständige Repositorys, PARIS enthält ein Teilrepository.

## Informationen zu diesem Vorgang

Wenn Sie ein Thema in einem Warteschlangenmanager in einem Cluster definieren, müssen Sie angeben, ob es sich bei dem Thema um ein Clusterthema handelt, und (falls ja) das Routing innerhalb des Clusters für Veröffentlichungen und Subskriptionen für dieses Thema. Um das Thema zu einem Clusterthema zu machen, konfigurieren Sie die Eigenschaft **CLUSTER** im Objekt TOPIC mit dem Namen des Clusters. Durch die Definition eines Clusterthemas auf einem WS-Manager im Cluster stellen Sie das Thema für den gesamten Cluster zur Verfügung. Um das Nachrichtenrouting auszuwählen, das im Cluster verwendet werden soll, setzen Sie die Eigenschaft **CLROUTE** für das Objekt TOPIC auf einen der folgenden Werte:

- **DIRECT**
- **TOPICHOST**

Topic-Routing ist standardmäßig **DIRECT**. Vor IBM MQ 8.0 war dies die einzige Option. Wenn Sie ein direkt geroutetes Cluster-Topic in einem Warteschlangenmanager konfigurieren, werden sämtliche Warteschlangenmanager im Cluster aller anderen Warteschlangenmanager im Cluster gewahr. Bei der Ausführung von Publish- und Subscribe-Operationen kann jeder Warteschlangenmanager direkt eine Verbindung zu anderen Warteschlangenmanagern im Cluster herstellen. Siehe Direct routed Publish/Subscribe-Cluster .

Ab IBM MQ 8.0können Sie Topic-Routing stattdessen als **TOPICHOST**konfigurieren. Bei Verwendung der Routing-Methode TOPICHOST können alle Warteschlangenmanager im Cluster die Clusterwarteschlangenmanager erkennen, die die Definition des weitergeleiteten Themas enthalten (d. h. die Warteschlangenmanager, in denen Sie das Themenobjekt definiert haben). Beim Ausführen von Publish/Subscribe-Operationen werden Warteschlangenmanager im Cluster nur mit diesen Topic-Host-Warteschlangenmanagern und nicht direkt miteinander verbunden. Die Topic-Host-Warteschlangenmanager sind für das Routing von Publikationen aus Warteschlangenmanagern verantwortlich, in denen Publikationen für Warteschlangenmanager mit übereinstimmenden Subskriptionen veröffentlicht werden. Weitere Informationen finden Sie unter Publish/Subscribe-Cluster für Themenhost .

**Anmerkung:** Nachdem ein Themenobjekt in einem Cluster zusammengefasst wurde (durch Festlegen der Eigenschaft **CLUSTER** ), können Sie den Wert der Eigenschaft **CLROUTE** nicht ändern. Sie müssen erst die Konfiguration des Objekts als Clusterthema rückgängig machen (indem **CLUSTER** auf ' ' gesetzt wird), damit dieser Wert geändert werden kann. Durch die Aufhebung des Clusters eines Themas wird die Themendefinition in ein lokales Thema konvertiert, wodurch sich ein Zeitraum ergibt, in dem keine Veröffentlichungen an Subskriptionen auf fernen Warteschlangenmanagern geliefert werden; dies sollte bei der Ausführung dieser Änderung berücksichtigt werden. Weitere Informationen finden Sie unter Auswirkung der Definition eines Themas ohne Clusterzuordnung mit dem gleichen Namen wie dem eines Clusterthemas von einem anderen Warteschlangenmanager. Wenn Sie versuchen, den Wert der Eigenschaft **CLROUTE** während der Clusterbildung zu ändern, generiert das System die Ausnahmebedingung MQRCCF_CLROUTE_NOT_ALTERABLE .

## Vorgehensweise

1. Wählen Sie einen Warteschlangenmanager aus, um Ihr Thema zu hosten.

   Jeder Cluster-WS-Manager kann ein Thema hosten. Wählen Sie einen der drei Warteschlangenmanager ( LONDON, NEWYORK oder PARIS) aus und konfigurieren Sie die Eigenschaften des Objekts TOPIC . Wenn Sie direktes Routing verwenden möchten, ist es nicht betriebsbereit, welchen Warteschlangenmanager Sie auswählen. Wenn Sie planen, das Thema Host-Routing zu verwenden, verfügt der ausgewählte Warteschlangenmanager über zusätzliche Zuständigkeiten für die Weiterleitung von Veröffentlichungen. Wählen Sie daher für das Thema Host Routing einen Warteschlangenmanager aus, der auf einem Ihrer leistungsfähigeren Systeme gehostet wird und über eine gute Netzkonnektivität verfügt.
2. Definieren Sie ein Thema in einem Warteschlangenmanager .

   Um das Thema zu einem Clusterthema zu machen, schließen Sie den Clusternamen ein, wenn Sie das Thema definieren, und legen Sie das Routing fest, das Sie für Veröffentlichungen und Subskriptionen für dieses Thema verwenden wollen. Um beispielsweise ein Clusterthema für direktes Routing auf dem LONDON-Warteschlangenmanager zu erstellen, erstellen Sie das Thema wie folgt:

   ```
   DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
   ```

   Durch die Definition eines Clusterthemas auf einem WS-Manager im Cluster stellen Sie das Thema für den gesamten Cluster zur Verfügung.

   Weitere Informationen zur Verwendung von **CLROUTE** finden Sie unter THEMA DEFINIEREN (CLROUTE) und Routing für Publish/Subscribe-Cluster: Hinweise zum Verhalten.

## Ergebnisse

Der Cluster ist bereit, Veröffentlichungen und Subskriptionen für das Thema zu empfangen.

## Nächste Schritte

Wenn Sie einen Publish/Subscribe-Cluster für Themenhost konfiguriert haben, möchten Sie wahrscheinlich einen zweiten Themenhost für dieses Thema hinzufügen. Siehe „Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen” auf Seite 475.

Wenn Sie mehrere separate Publish/Subscribe-Cluster haben, z. B. weil Ihre Organisation geographisch verteilt ist, möchten Sie möglicherweise einige Clusterthemen in alle Cluster weitergeben. Sie können dies tun, indem Sie die Cluster in einer Hierarchie miteinander verbinden. Siehe „Kombinieren der Topic-Bereiche mehrerer Cluster” auf Seite 481. Sie können auch steuern, welche Veröffentlichungen von einem Cluster in einen anderen fließen. Siehe „Topic-Bereiche in mehreren Clustern kombinieren und isolieren” auf Seite 482.

**Zugehörige Konzepte**

Kombinieren von Veröffentlichungs-und Subskriptionsbereichen
Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen
Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

**Zugehörige Tasks**

Clusterthemendefinition in einen anderen WS-Manager verschieben
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen

In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthemenobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden
Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen
Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

Publish/Subscribe-Cluster entwerfen

Fehlerbehebung bei Problemen mit verteiltem Publish/Subscribe

Clusterveröffentlichungs-/Subskriptionssubskribieren

## Clusterthemendefinition in einen anderen WS-Manager verschieben

Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

### Informationen zu diesem Vorgang

Sie können mehrere Definitionen desselben Clusterthemenobjekts in einem Cluster haben. Dies ist ein normaler Status für einen Topic-Host-Routing-Cluster und ein ungewöhnlicher Status für einen direkt weitergeleiteten Cluster. Weitere Informationen finden Sie im Abschnitt Mehrere Cluster-Topic-Definitionen mit demselben Namen .

Führen Sie die folgenden Schritte aus, um eine Clusterthemendefinition in einen anderen WS-Manager im Cluster zu verschieben, ohne den Fluss der Veröffentlichungen zu unterbrechen. Die Prozedur verschiebt eine Definition vom WS-Manager `QM1` zum Warteschlangenmanager `QM2` .

### Vorgehensweise

1. Erstellen Sie ein Duplikat der Clusterthemendefinition auf `QM2` .

   Für direktes Routing setzen Sie alle Attribute so, dass sie mit der Definition von `QM1` übereinstimmen.

   Definieren Sie für das Topic-Host-Routing zunächst den neuen Topic-Host als `PUB(DISABLED)`. Auf diese Weise kann `QM2` die Subskriptionen im Cluster kennen lernen, aber nicht die Weiterleitung von Veröffentlichungen starten.
2. Warten Sie, bis Informationen über den Cluster weitergegeben werden.

   Warten Sie, bis die neue Clusterthemendefinition von den vollständigen Repository-WS-Managern an alle WS-Manager im Cluster weitergegeben wird. Verwenden Sie den Befehl **DISPLAY CLUSTER** , um die Clusterthemen auf jedem Cluster-Member anzuzeigen und nach einer Definition zu suchen, die von `QM2`stammt.

   Warten Sie, bis der neue Topic-Host auf `QM2` das Thema Host-Routing enthält, um alle Subskriptionen zu erfahren. Vergleichen Sie die Proxy-Subskriptionen, die `QM2` bekannt sind, und die Proxy-Subskriptionen, die `QM1` bekannt Eine Möglichkeit, die Proxy-Subskriptionen auf einem Warteschlangenmanager anzuzeigen, besteht darin, den folgenden **runmqsc** -Befehl auszugeben:

   ```
   DISPLAY SUB(*) SUBTYPE(PROXY)
   ```

3. Definieren Sie für das Topic-Host-Routing den Topic-Host auf QM2 als PUB(ENABLED) und anschließend den Topic-Host auf QM1 als PUB(DISABLED) neu.

   Da der neue Themenhost auf QM2 nun von allen Subskriptionen auf anderen Warteschlangenmanagern erfahren hat, kann der Topic-Host Routing-Veröffentlichungen starten.

   Wenn Sie die Einstellung PUB (DISABLED) verwenden, um den Nachrichtenverkehr über QM1 in den Quiescemodus zu setzen, stellen Sie sicher, dass keine Veröffentlichungen im Zug durch QM1 ausgeführt werden, wenn Sie die Clusterthemendefinition löschen.
4. Löschen Sie die Clusterthemendefinition aus QM1 .

   Sie können die Definition nur aus QM1 löschen, wenn der WS-Manager verfügbar ist. Andernfalls müssen Sie beide Definitionen verwenden, bis QM1 erneut gestartet oder zwangsweise entfernt wird.

   Wenn QM1 lange Zeit nicht verfügbar ist und Sie die Clusterthemendefinition in QM2 ändern müssen, ist die QM2 -Definition neuer als die QM1 -Definition und hat daher in der Regel Vorrang.

   Wenn es während dieses Zeitraums Unterschiede zwischen den Definitionen in QM1 und QM2 gibt, werden Fehler in die Fehlerprotokolle der beiden WS-Manager geschrieben, die Sie auf die Clusterthemendefinition hinweisen, die sich im Konflikt befindet.

   Wenn QM1 nie zum Cluster zurückkehren wird, z. B. aufgrund einer unerwarteten Stilllegung nach einem Hardwarefehler, können Sie als letztes Mittel den Befehl RESET CLUSTER verwenden, um den Warteschlangenmanager zwangsweise auszuwerfen. **RESET CLUSTER** löscht automatisch alle Themenobjekte, die sich auf dem Zielwarteschlangenmanager befinden.

**Zugehörige Konzepte**

Kombinieren von Veröffentlichungs-und Subskriptionsbereichen
Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen
Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

**Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren
Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen
In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthemenobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden
Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen
Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

## Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen

In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthe-

menobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

## Vorbereitende Schritte

Das Definieren desselben Clusterthemenobjekts auf mehreren Warteschlangenmanagern ist nur funktional nützlich für einen Topic-Host-Routing-Cluster. Durch das Definieren mehrerer übereinstimmender Themen in einem direkt weitergeleiteten Cluster wird dessen Verhalten nicht geändert. Diese Task gilt nur für Topic-Host-Routing-Cluster.

Bei dieser Task wird davon ausgegangen, dass Sie den Artikel Mehrere Cluster-Topic-Definitionen mit demselben Namen gelesen haben, insbesondere die folgenden Abschnitte:

- Mehrere Cluster-Topic-Definitionen in einem Cluster mit Topic-Host-Routing
- Spezielle Ausnahmen für den PUB-Parameter

## Informationen zu diesem Vorgang

Wenn ein Warteschlangenmanager einen weitergeleiteten Topic-Host erstellt hat, muss er zunächst die Existenz aller verwandten Themen, die im Cluster subskribiert wurden, kennen lernen. Wenn Veröffentlichungen zu diesen Themen zu dem Zeitpunkt veröffentlicht werden, zu dem ein zusätzlicher Themenhost hinzugefügt wird, und eine Veröffentlichung an den neuen Host weitergeleitet wird, bevor dieser Host die Existenz von Subskriptionen auf anderen Warteschlangenmanagern im Cluster erlernt hat, leitet der neue Host diese Veröffentlichung nicht an diese Subskriptionen weiter. Dies bewirkt, dass Subskriptionen keine Veröffentlichungen mehr enthalten.

Veröffentlichungen werden nicht über Topic-Host-Warteschlangenmanager weitergeleitet, die den Parameter **PUB** des Clusterthemenobjekts explizit auf `INAKTIVIERT`gesetzt haben. Sie können diese Einstellung verwenden, um sicherzustellen, dass keine Subskriptionen Veröffentlichungen verpassen, während ein zusätzlicher Topic-Host hinzugefügt wird.

**Anmerkung:** Während ein Warteschlangenmanager ein Clusterthema enthält, das als `PUB(DISABLED)` definiert wurde, können Publisher, die mit diesem Warteschlangenmanager verbunden sind, keine Nachrichten veröffentlichen und übereinstimmende Subskriptionen auf diesem Warteschlangenmanager empfangen keine Veröffentlichungen, die auf anderen Warteschlangenmanagern im Cluster veröffentlicht wurden. Aus diesem Grund muss sorgfältig geprüft werden, ob Topic-Host-Themen in WS-Managern, in denen Subskriptionen vorhanden sind, und Veröffentlichungsanwendungen verbunden sind.

## Vorgehensweise

1. Konfigurieren Sie einen neuen Topic-Host und definieren Sie zunächst den neuen Topic-Host als `PUB(DISABLED)`.

   Dies ermöglicht dem neuen Topic-Host, die Subskriptionen im Cluster zu lernen, aber nicht die Weiterleitung von Veröffentlichungen zu starten.

   Informationen zum Konfigurieren eines Themenhosts finden Sie in „Publish/Subscribe-Cluster konfigurieren” auf Seite 471.
2. Stellen Sie fest, wann der neue Topic-Host von allen Subskriptionen gelernt hat.

   Vergleichen Sie dazu die Proxy-Subskriptionen, die dem neuen Topic-Host bekannt sind, und die dem vorhandenen Topic-Host bekannten Proxy-Subskriptionen. Eine Möglichkeit, die Proxy-Subskriptionen anzuzeigen, besteht darin, den folgenden **runmqsc**-Befehl auszugeben: `DISPLAY SUB(*) SUBTYPE(PROXY)`
3. Definieren Sie den neuen Topic-Host als `PUB(ENABLED)`.

   Nachdem der neue Topic-Host alle Subskriptionen auf anderen Warteschlangenmanagern kennengelernt hat, kann das Thema Routing-Veröffentlichungen starten.

**Zugehörige Konzepte**

Kombinieren von Veröffentlichungs-und Subskriptionsbereichen
Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen
Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

**Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren
Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

Clusterthemendefinition in einen anderen WS-Manager verschieben
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden
Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen
Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

## Kombinieren von Veröffentlichungs-und Subskriptionsbereichen

Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Veröffentlichungen können allen Warteschlangenmanagern, die in einer Publish/Subscribe-Topologie verbunden sind, oder nur dem lokalen WS-Manager fließen. Analog zu Proxy-Subskriptionen. Welche Veröffentlichungen mit einer Subskription übereinstimmen, wird durch die Kombination dieser beiden Datenflüsse gesteuert.

Veröffentlichungen und Subskriptionen können sowohl auf `QMGR` als auch auf `ALL` definiert werden. Wenn ein Publisher und ein Subskribent beide mit demselben Warteschlangenmanager verbunden sind, wirken sich die Geltungsbereichseinstellungen nicht auf die Veröffentlichungen aus, die der Subskribent von diesem Bereitsteller erhält.

Wenn der Bereitsteller und der Subskribent mit unterschiedlichen Warteschlangenmanagern verbunden sind, müssen beide Einstellungen `ALL` sein, um ferne Veröffentlichungen zu empfangen.

Angenommen, Publisher sind mit verschiedenen Warteschlangenmanagern verbunden. Wenn Sie möchten, dass ein Subskribent Veröffentlichungen von einem beliebigen Publisher empfängt, legen Sie den Subskriptionsumfang auf `ALL` fest. Sie können dann für jeden Bereitsteller entscheiden, ob der Umfang seiner Veröffentlichungen auf Subskribenten beschränkt werden soll, die lokal für den Publisher sind.

Angenommen, die Subskribenten sind mit verschiedenen Warteschlangenmanagern verbunden. Wenn die Veröffentlichungen von einem Publisher an alle Subskribenten gesendet werden sollen, setzen Sie den Veröffentlichungsumfang auf `ALL` . Wenn Sie möchten, dass ein Subskribent nur Veröffentlichungen von einem Publisher empfängt, der mit demselben Warteschlangenmanager verbunden ist, legen Sie den Subskriptionsumfang auf `QMGR` fest.

**Beispiel: Fußball-Ergebnisdienst**

Angenommen, Sie sind ein Mitglied-Team in einer Football-Liga. Jedes Team verfügt über einen WS-Manager, der mit allen anderen Teams in einem Publish/Subscribe-Cluster verbunden ist.

Die Teams veröffentlichen die Ergebnisse aller Spiele, die auf ihrem Heimspielplatz gespielt wurden, unter dem Thema `Football/result/`*`Home team name/Away team name`*. Die Zeichenfolgen in Kursivschrift sind variable Themennamen, und die Veröffentlichung ist das Ergebnis der Übereinstimmung.

Jeder Club veröffentlicht außerdem die Ergebnisse nur für den Club unter Verwendung der Themenzeichenfolge `Football/myteam/`*`Home team name/Away team name`* erneut.

Beide Themen werden im gesamten Cluster veröffentlicht.

Die folgenden Abonnements wurden von der Liga eingerichtet, damit die Fans eines jeden Teams die Ergebnisse auf drei interessante Arten abonnieren können.

Beachten Sie, dass Sie Clusterthemen mit SUBSCOPE(QMGR) einrichten können. Die Themendefinitionen werden an jedes Member des Clusters weitergegeben, aber der Geltungsbereich der Subskription ist nur der lokale Warteschlangenmanager. So empfangen Subskribenten in jedem WS-Manager verschiedene Veröffentlichungen aus derselben Subskription.

**Alle Ergebnisse empfangen**

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

**Alle Ausgangsergebnisse empfangen**

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

Da die Subskription den Geltungsbereich `QMGR` hat, werden nur die Ergebnisse verglichen, die auf dem Homerground veröffentlicht werden.

**Alle eigenen Teamergebnisse empfangen**

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

Da die Subskription den Geltungsbereich `QMGR` hat, werden nur die Ergebnisse des lokalen Teams, die lokal erneut veröffentlicht werden, abgeglichen.

**Zugehörige Konzepte**

Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen
Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

Verteilte Publish/Subscribe-Netzwerke

Veröffentlichungsumfang

Subskriptionsumfang

**Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren
Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

Clusterthemendefinition in einen anderen WS-Manager verschieben
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen
In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthe-

menobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden
Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen
Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

## Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen

Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

Sie können verschiedene Publish/Subscribe-Topic-Bereiche erstellen, indem Sie die Bausteine der Attribute **CLUSTER**, **PUBSCOPE** und **SUBSCOPE** , Publish/Subscribe-Cluster und Publish/Subscribe-Hierarchien verwenden.

Ausgehend vom Beispiel der Skalierung von einem einzelnen WS-Manager zu einem Publish/Subscribe-Cluster veranschaulichen die folgenden Szenarios verschiedene Publish/Subscribe-Topologien.

**Zugehörige Konzepte**

Kombinieren von Veröffentlichungs-und Subskriptionsbereichen
Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

**Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren
Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

Clusterthemendefinition in einen anderen WS-Manager verschieben
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen
In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthemenobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden
Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen
Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

Clusterthemen definieren

### *Erstellen eines einzelnen Topic-Bereichs in einem Publish/Subscribe-Cluster*

Skalieren Sie ein Publish/Subscribe-System, das auf mehreren Warteschlangenmanagern ausgeführt werden soll. Verwenden Sie einen Publish/Subscribe-Cluster, um jedem Bereitsteller und Subskribenten einen einzigen identischen Topic-Bereich zur Verfügung zu stellen.

## Vorbereitende Schritte

Sie haben ein Publish/Subscribe-System in einem WS-Manager der Version 7 implementiert.

Erstellen Sie immer Topicbereiche mit eigenen Stammtopics, anstatt sich darauf zu verlassen, dass die Attribute von `SYSTEM.BASE.TOPIC` übernommen werden. Wenn Sie Ihr Publish/Subscribe-System in einem Cluster skalieren, können Sie Ihre Stammthemen als Clusterthemen auf dem Clusterthemenhost definieren und anschließend alle Themen im gesamten Cluster gemeinsam nutzen.

## Informationen zu diesem Vorgang

Sie möchten jetzt das System skalieren, um mehr Publisher und Subskribenten zu unterstützen, und haben alle Themen, die im gesamten Cluster sichtbar sind.

## Vorgehensweise

1. Erstellen Sie einen Cluster, der mit dem Publish/Subscribe-System verwendet werden soll.

   Wenn Sie einen vorhandenen traditionellen Cluster haben, ist es aus Leistungsgründen besser, einen neuen Cluster für das neue Publish/Subscribe-System einzurichten. Sie können dieselben Server für die Cluster-Repositorys beider Cluster verwenden.
2. Wählen Sie einen Warteschlangenmanager, möglicherweise eines der Repositorys, als Cluster-Topic-Host aus.
3. Stellen Sie sicher, dass alle Themen, die im gesamten Publish/Subscribe-Cluster sichtbar sein sollen, in ein Verwaltungsthemenobjekt aufgelöst werden.
   Legen Sie das Attribut **CLUSTER** für die Benennung des Publish/Subscribe-Clusters fest.

## Nächste Schritte

Verbinden Sie Publisher-und Subskribentenanwendungen mit allen WS-Managern im Cluster.

Erstellen Sie Verwaltungsthemenobjekte mit dem Attribut **CLUSTER**. Die Themen werden auch im gesamten Cluster weitergegeben. Publisher-und Subskribentenprogramme verwenden die Verwaltungsthemen, so dass ihr Verhalten nicht durch die Verbindung zu verschiedenen Warteschlangenmanagern im Cluster geändert wird.

Wenn `SYSTEM.BASE.TOPIC` wie ein Clusterthema auf jedem Warteschlangenmanager agieren soll, müssen Sie es auf jedem Warteschlangenmanager ändern.

**Zugehörige Konzepte**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

**Zugehörige Tasks**

Kombinieren der Topic-Bereiche mehrerer Cluster
Erstellen Sie Topic-Bereiche, die sich über mehrere Cluster erstrecken. Publizieren Sie zu einem Thema in einem Cluster und subskribieren Sie es in einem anderen Cluster.

Topic-Bereiche in mehreren Clustern kombinieren und isolieren
Isolieren Sie einige Topic-Bereiche in einem bestimmten Cluster, und kombinieren Sie andere Topic-Bereiche, um sie in allen verbundenen Clustern zugänglich zu machen.

Themenbereiche in mehreren Clustern veröffentlichen und subskribieren

Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subskribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

Clusterthemen definieren

### *Kombinieren der Topic-Bereiche mehrerer Cluster*

Erstellen Sie Topic-Bereiche, die sich über mehrere Cluster erstrecken. Publizieren Sie zu einem Thema in einem Cluster und subskribieren Sie es in einem anderen Cluster.

## Vorbereitende Schritte

Bei dieser Task wird davon ausgegangen, dass Sie vorhandene Publish/Subscribe-Cluster direkt weitergeleitet haben und einige Clusterthemen in allen Clustern weitergeben möchten.

**Anmerkung:** Dies kann für Topic-Host-Publish/Subscribe-Cluster nicht möglich sein.

## Informationen zu diesem Vorgang

Um Veröffentlichungen von einem Cluster an einen anderen weiterzugeben, müssen Sie die Cluster in einer Hierarchie miteinander verknüpfen (siehe Abbildung 65 auf Seite 481). Die hierarchischen Verbindungen geben Subskriptionen und Veröffentlichungen zwischen den verbundenen Warteschlangenmanagern weiter und die Cluster propagieren Clusterthemen in jedem Cluster, jedoch nicht zwischen Clustern.

Die Kombination dieser beiden Mechanismen propagiert Clusterthemen zwischen allen Clustern. Sie müssen die Clusterthemendefinitionen in jedem Cluster wiederholen.

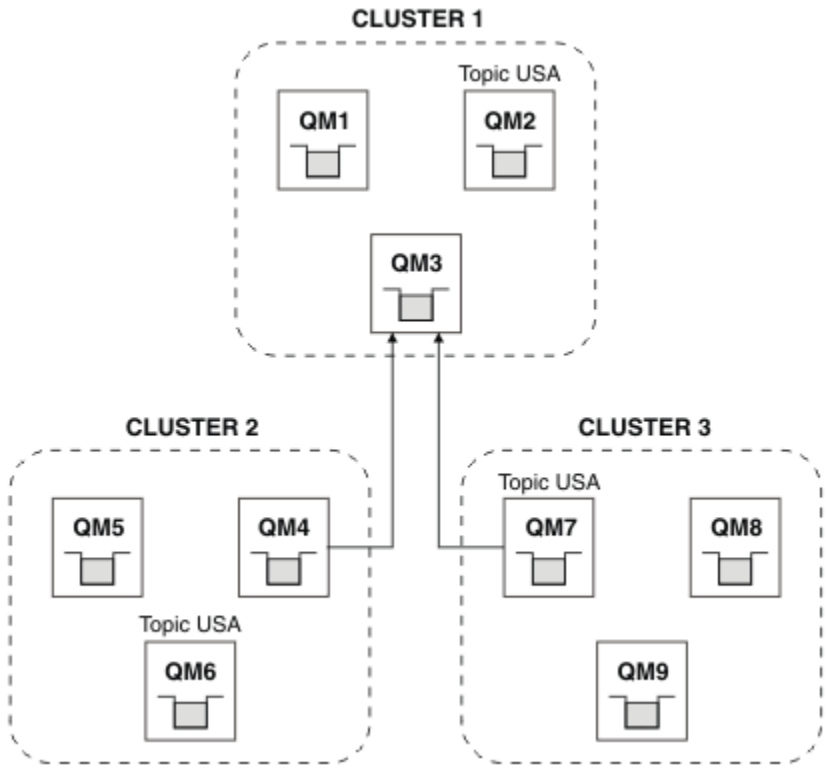



*Abbildung 65. Cluster mit Hierarchien verbinden*

Mit den folgenden Schritten können Sie die Cluster in einer Hierarchie miteinander verbinden.

## Vorgehensweise

1. Erstellen Sie zwei Gruppen von Sender-Empfänger-Kanälen, um QM3 und QM4 und QM3 und QM7 in beide Richtungen zu verbinden. Sie müssen die traditionellen Sender-Empfänger-Kanäle und -Übertragungswarteschlangen anstelle eines Clusters verwenden, um eine Hierarchie zu verbinden.
2. Erstellen Sie drei Übertragungswarteschlangen mit den Namen der Ziel-WS-Manager. Verwenden Sie WS-Manager-Aliasnamen, wenn Sie aus irgendeinem Grund den Namen des Zielwarteschlangenmanagers nicht als Name der Übertragungswarteschlange verwenden können.
3. Konfigurieren Sie die Übertragungswarteschlangen, um die Senderkanäle auszulösen.
4. Überprüfen Sie, ob **PSMODE** von QM3, QM4 und QM7 auf AKTIVIEREN gesetzt ist.
5. Ändern Sie das Attribut **PARENT** von QM4 und QM7 in QM3.
6. Überprüfen Sie den Status der Elternbeziehung zwischen den Warteschlangenmanagern in beide Richtungen.
7. Verwaltungsthema USA mit dem Attribut **CLUSTER** ( 'CLUSTER 1' ), **CLUSTER** ( 'CLUSTER 2' ) und **CLUSTER** ( 'CLUSTER 3' ) erstellen auf jedem der drei Cluster-Topic-Host-Warteschlangenmanager in den Clustern 1, 2 und 3. Der Cluster-Topic-Host muss kein hierarchisch verbundener Warteschlangenmanager sein.

## Nächste Schritte

Sie können nun das Clusterthema USA in Abbildung 65 auf Seite 481veröffentlichen oder abonnieren. Der Subskriptionsablauf der Veröffentlichungen wird in allen drei Clustern an Publisher und Subskribenten fließen.

Angenommen, Sie haben USA nicht als Clusterthema in den anderen Clustern erstellt. Wenn USA nur unter QM7 definiert ist, werden Veröffentlichungen und Subskriptionen für USA zwischen QM7, QM8, QM9 und QM3 ausgetauscht. Publisher und Subskribenten, die unter QM7, QM8, QM9 ausgeführt werden, übernehmen die Attribute des Verwaltungsthemas USA. Publisher und Subskribenten in QM3 übernehmen die Attribute von SYSTEM.BASE.TOPIC auf QM3.

Weitere Informationen hierzu finden Sie im Abschnitt „Topic-Bereiche in mehreren Clustern kombinieren und isolieren” auf Seite 482.

**Zugehörige Konzepte**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

**Zugehörige Tasks**

Erstellen eines einzelnen Topic-Bereichs in einem Publish/Subscribe-Cluster
Skalieren Sie ein Publish/Subscribe-System, das auf mehreren Warteschlangenmanagern ausgeführt werden soll. Verwenden Sie einen Publish/Subscribe-Cluster, um jedem Bereitsteller und Subskribenten einen einzigen identischen Topic-Bereich zur Verfügung zu stellen.

Topic-Bereiche in mehreren Clustern kombinieren und isolieren
Isolieren Sie einige Topic-Bereiche in einem bestimmten Cluster, und kombinieren Sie andere Topic-Bereiche, um sie in allen verbundenen Clustern zugänglich zu machen.

Themenbereiche in mehreren Clustern veröffentlichen und subskribieren
Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subskribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

Clusterthemen definieren

## *Topic-Bereiche in mehreren Clustern kombinieren und isolieren*

Isolieren Sie einige Topic-Bereiche in einem bestimmten Cluster, und kombinieren Sie andere Topic-Bereiche, um sie in allen verbundenen Clustern zugänglich zu machen.

## Vorbereitende Schritte

Untersuchen Sie das Thema „Kombinieren der Topic-Bereiche mehrerer Cluster” auf Seite 481. Es kann für Ihre Anforderungen ausreichend sein, ohne einen zusätzlichen WS-Manager als Bridge hinzuzufügen.

**Anmerkung:** Sie können diese Task nur mit Hilfe von direkt weitergeleiteten Publish/Subscribe-Clustern ausführen. Dies kann mit Topic-Host-Routing-Clustern nicht möglich sein.

## Informationen zu diesem Vorgang

Eine mögliche Verbesserung bezüglich der unter Abbildung 65 auf Seite 481 im Abschnitt „Kombinieren der Topic-Bereiche mehrerer Cluster” auf Seite 481 dargestellten Topologie besteht in der Eingrenzung von Clusterthemen, die nicht von allen Clustern gemeinsam genutzt werden. Isolieren Sie Cluster, indem Sie einen Überbrückungswarteschlangenmanager erstellen, der sich nicht in einem der Cluster befindet (siehe Abbildung 66 auf Seite 483). Verwenden Sie den Brückenwarteschlangenmanager, um zu filtern, welche Veröffentlichungen und Subskriptionen von einem Cluster in einen anderen fließen können.

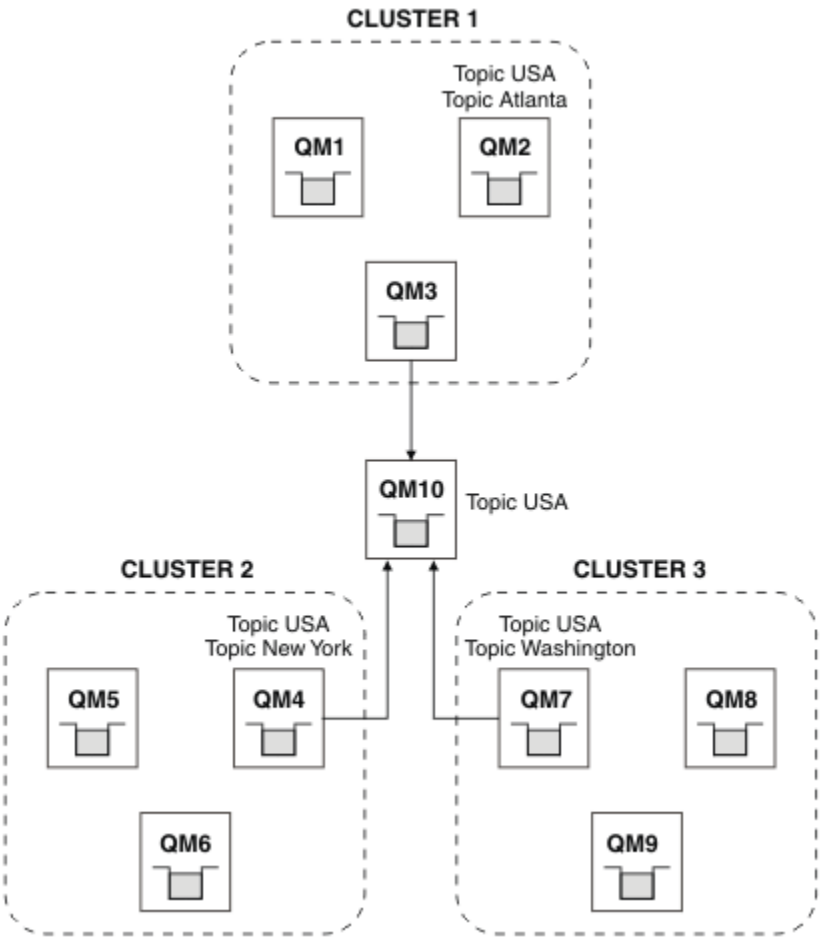



*Abbildung 66. Überbrückte Cluster*

Verwenden Sie die Brücke, um Clusterthemen zu isolieren, die nicht über die Brücke auf den anderen Clustern zugänglich gemacht werden sollen. In Abbildung 66 auf Seite 483 ist USA ein Clusterthema, das in allen Clustern gemeinsam genutzt wird, und `Atlanta`, `New York` und `Washington` sind Clusterthemen, die jeweils nur in einem Cluster gemeinsam genutzt werden.

Model-Ihre Konfiguration mit der folgenden Prozedur:

## Vorgehensweise

1. Ändern Sie alle SYSTEM.BASE.TOPIC -Themenobjekte in **SUBSCOPE** ( `Warteschlangenmanager` ). und **PUBSCOPE** ( `Warteschlangenmanager` ) auf allen Warteschlangenmanagern.

Es werden keine Themen (auch Clusterthemen) an andere Warteschlangenmanager weitergegeben, es sei denn, Sie legen explizit **SUBSCOPE** ( ALLE ) fest. und **PUBSCOPE** ( ALLE ) für das Stammthema Ihrer Clusterthemen.

2. Definieren Sie die Themen auf den drei Cluster-Topic-Host-Warteschlangenmanagern, die in jedem Cluster gemeinsam genutzt werden sollen, mit den Attributen **CLUSTER** (*Clustername*), **SUBSCOPE** ( ALLE ) und **PUBSCOPE** ( ALLE ).

   Wenn einige Clusterthemen von allen Clustern gemeinsam genutzt werden sollen, definieren Sie in jedem der Cluster dasselbe Thema. Verwenden Sie den Clusternamen jedes Cluster als Clusterattribut.
3. Definieren Sie für die Clusterthemen, die von allen Clustern gemeinsam genutzt werden sollen, die Themen erneut im Brückenwarteschlangenmanager ( QM10 ) mit den Attributen **SUBSCOPE** ( ALLE ) und **PUBSCOPE** ( ALLE ).

**Beispiel**

In dem Beispiel in Abbildung 66 auf Seite 483 werden nur Themen, die von USA übernommen werden, zwischen allen drei Clustern weitergegeben.

## Nächste Schritte

Subskriptionen für Themen, die auf dem Bridge-Warteschlangenmanager mit **SUBSCOPE** definiert sind ( ALLE ) und **PUBSCOPE** ( ALLE ) werden zwischen den Clustern weitergegeben.

Subskriptionen für Themen, die in jedem Cluster mit den Attributen **CLUSTER** (*Clustername*), **SUBSCOPE** ( ALLE ) definiert sind und **PUBSCOPE** ( ALLE ) werden in jedem Cluster weitergegeben.

Alle anderen Subskriptionen sind lokal für einen WS-Manager.

**Zugehörige Konzepte**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

Veröffentlichungsumfang

Subskriptionsumfang

**Zugehörige Tasks**

Erstellen eines einzelnen Topic-Bereichs in einem Publish/Subscribe-Cluster
Skalieren Sie ein Publish/Subscribe-System, das auf mehreren Warteschlangenmanagern ausgeführt werden soll. Verwenden Sie einen Publish/Subscribe-Cluster, um jedem Bereitsteller und Subskribenten einen einzigen identischen Topic-Bereich zur Verfügung zu stellen.

Kombinieren der Topic-Bereiche mehrerer Cluster
Erstellen Sie Topic-Bereiche, die sich über mehrere Cluster erstrecken. Publizieren Sie zu einem Thema in einem Cluster und subskribieren Sie es in einem anderen Cluster.

Themenbereiche in mehreren Clustern veröffentlichen und subskribieren
Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subskribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

Clusterthemen definieren

## *Themenbereiche in mehreren Clustern veröffentlichen und subskribieren*

Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subskribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

## Vorbereitende Schritte

Erstellen Sie mehrere traditionelle Cluster mit einigen Warteschlangenmanagern in den Schnittbereichen zwischen den Clustern.

## Informationen zu diesem Vorgang

Möglicherweise haben Sie die Überlappung von Clustern aus verschiedenen Gründen ausgewählt.

1. Sie verfügen über eine begrenzte Anzahl von Hochverfügbarkeitsservern oder Warteschlangenmanagern. Sie entscheiden, alle Cluster-Repositorys zu implementieren, und Clusterthemenhosts zu ihnen.
2. Sie verfügen über traditionelle WS-Manager-Cluster, die mit Gateway-WS-Managern verbunden sind. Sie möchten Publish/Subscribe-Anwendungen in derselben Clustertopologie implementieren.
3. Sie haben eine Reihe von eigenständigen Publish/Subscribe-Anwendungen. Aus Leistungsgründen ist es besser, Publish/Subscribe-Cluster klein zu halten und sich von herkömmlichen Clustern zu trennen. Sie haben sich entschieden, die Anwendungen in verschiedenen Clustern zu implementieren. Sie möchten jedoch auch alle Publish/Subscribe-Anwendungen in einem WS-Manager überwachen, da Sie nur eine Kopie der Überwachungsanwendung lizenziert haben. Dieser WS-Manager muss über Zugriff auf die Veröffentlichungen zu Clusterthemen in allen Clustern verfügen.

Wenn Sie dafür sorgen, dass Ihre Topics in nicht überlappenden Topic-Bereichen definiert sind, können Sie die Topics in überlappenden Publish/Subscribe-Clustern implementieren; weitere Informationen finden Sie im Abschnitt Abbildung 67 auf Seite 485. Wenn sich die Topic-Bereiche überschneiden, führt die Implementierung in überlappenden Clustern zu Problemen.

Da die Publish/Subscribe-Cluster überlappen, können Sie alle Topic-Bereiche mit den Warteschlangenmanagern in der Überlappung veröffentlichen und subskribieren.

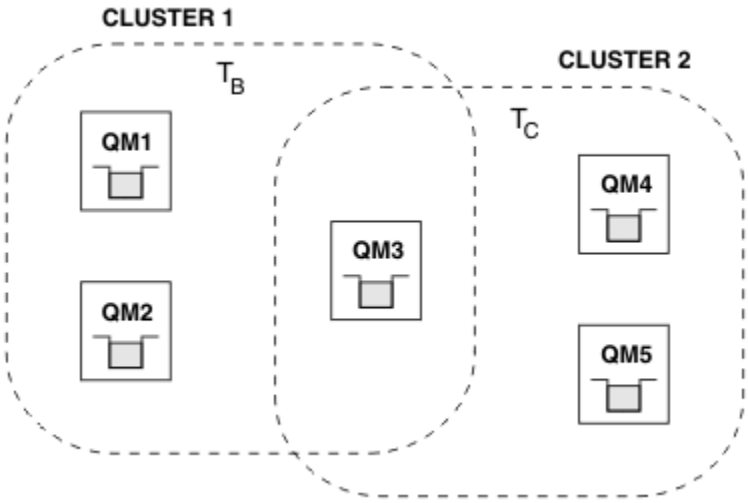



*Abbildung 67. Überschneidung von Clustern, nicht überlappenden Themenbereichen*

## Vorgehensweise

Erstellen Sie eine Möglichkeit, um sicherzustellen, dass die Topic-Bereiche sich nicht überschneiden.

Definieren Sie beispielsweise für jeden der Themenbereiche ein eindeutiges Stammthema. Machen Sie die Themen in den Stammthemen zu Themen.

a) `DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...`

b) `DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...`

### Beispiel

In Abbildung 67 auf Seite 485 können Publisher und Subskribenten, die mit QM3 verbunden sind, $T_B$ oder $T_C$ veröffentlichen oder subskribieren.

## Nächste Schritte

Verbinden Sie Publisher und Subskribenten, die Themen in beiden Clustern zu Warteschlangenmanagern in der Überlappung verwenden.

Verbinden Sie Publisher und Subskribenten, die nur Topics in einem bestimmten Cluster zu Warteschlangenmanagern verwenden dürfen, die sich nicht in der Überlappung enthalten.

**Zugehörige Konzepte**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

**Zugehörige Tasks**

Erstellen eines einzelnen Topic-Bereichs in einem Publish/Subscribe-Cluster
Skalieren Sie ein Publish/Subscribe-System, das auf mehreren Warteschlangenmanagern ausgeführt werden soll. Verwenden Sie einen Publish/Subscribe-Cluster, um jedem Bereitsteller und Subskribenten einen einzigen identischen Topic-Bereich zur Verfügung zu stellen.

Kombinieren der Topic-Bereiche mehrerer Cluster
Erstellen Sie Topic-Bereiche, die sich über mehrere Cluster erstrecken. Publizieren Sie zu einem Thema in einem Cluster und subskribieren Sie es in einem anderen Cluster.

Topic-Bereiche in mehreren Clustern kombinieren und isolieren
Isolieren Sie einige Topic-Bereiche in einem bestimmten Cluster, und kombinieren Sie andere Topic-Bereiche, um sie in allen verbundenen Clustern zugänglich zu machen.

Clusterthemen definieren

## WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden

Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

### Vorbereitende Schritte

1. Warteschlangenmanager in einer Publish/Subscribe-Hierarchie müssen eindeutige WS-Manager-Namen haben.
2. Eine Publish/Subscribe-Hierarchie basiert auf der WS-Manager-Funktion " queued publish/subscribe " . Diese Option muss sowohl auf dem übergeordneten als auch auf dem untergeordneten Warteschlangenmanager aktiviert sein. Siehe „In Warteschlange eingereihtes Publish/Subscribe starten” auf Seite 467.
3. Die Publish/Subscribe-Beziehung stützt sich auf die Sender-und Empfängerkanäle des Warteschlangenmanagers. Es gibt zwei Möglichkeiten, die Kanäle einzurichten:
   - Fügen Sie sowohl die übergeordneten als auch die untergeordneten Warteschlangenmanager zu einem IBM MQ-Cluster hinzu. Siehe „WS-Manager zu einem Cluster hinzufügen” auf Seite 339.
   - Richten Sie ein Sender-/Empfängerkanalpaar vom untergeordneten Warteschlangenmanager zum übergeordneten und vom übergeordneten Warteschlangenmanager zum untergeordneten Element ein. Jeder Kanal muss entweder eine Übertragungswarteschlange verwenden, die denselben Namen wie der Zielwarteschlangenmanager hat, oder einen WS-Manager-Aliasnamen mit demselben Namen wie der Zielwarteschlangenmanager. Weitere Informationen zur Einrichtung einer Punkt-zu-Punkt-Kanalverbindung finden Sie im Abschnitt „Verteilte Warteschlangenverfahren in IBM MQ” auf Seite 207.

   Für Beispiele, die eine Hierarchie über jede Art von Kanalkonfiguration konfigurieren, finden Sie in den folgenden Publish/Subscribe-Hierarchie-Szenarios die folgenden Szenarios:

   - Szenario 1: Punkt-zu-Punkt-Kanäle mit einem Warteschlangenmanager-Aliasnamen verwenden
   - Szenario 2: Punkt-zu-Punkt-Kanäle verwenden, wobei der Name der Übertragungswarteschlange mit dem des fernen Warteschlangenmanagers identisch ist
   - Szenario 3: Clusterkanal zum Hinzufügen eines Warteschlangenmanagers verwenden

## Informationen zu diesem Vorgang

Verwenden Sie den Befehl ALTER QMGR PARENT (*PARENT_NAME*) **runmqsc** , um untergeordnete Elemente mit übergeordneten Elementen zu verbinden. Diese Konfiguration wird für den untergeordneten Warteschlangenmanager ausgeführt, wobei *PARENT_NAME* für den Namen des übergeordneten Warteschlangenmanagers steht.

## Vorgehensweise

ALTER QMGR PARENT (*PARENT_NAME*)

### Beispiel

Das erste Beispiel zeigt, wie der Warteschlangenmanager QM2 als untergeordnetes Element von QM1 zugeordnet wird, und fragt anschließend QM2 ab, um sicherzustellen, dass er erfolgreich zu einem untergeordneten Element mit dem **STATUS** von AKTIV wurde:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
     1 : alter qmgr parent(QM1)
AMQ8005: IBM MQ queue manager changed.
display pubsub all
     2 : display pubsub all
AMQ8723: Display pub/sub status details.
   QMNAME(QM2)                             TYPE(LOCAL)
   STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
   QMNAME(QM1)                             TYPE(PARENT)
STATUS(ACTIVE)
```

Das nächste Beispiel zeigt das Ergebnis der Abfrage von QM1 für die zugehörigen Verbindungen:

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
     2 : display pubsub all
AMQ8723: Display pub/sub status details.
   QMNAME(QM1)                             TYPE(LOCAL)
   STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
   QMNAME(QM2)                             TYPE(CHILD)
   STATUS(ACTIVE)
```

Wenn **STATUS** nicht als AKTIV angezeigt wird, prüfen Sie, ob die Kanäle zwischen dem untergeordneten und dem übergeordneten Element ordnungsgemäß konfiguriert und aktiv sind. Überprüfen Sie die Fehlerprotokolle des Warteschlangenmanagers auf mögliche Fehler.

## Nächste Schritte

Standardmäßig werden die Themen, die von Publishern und Subskribenten in einem Warteschlangenmanager verwendet werden, gemeinsam mit den Publishern und Subskribenten auf den anderen Warteschlangenmanagern in der Hierarchie verwendet. Verwaltete Topics können konfiguriert werden, um die Ebene der gemeinsamen Nutzung über die Topiceigenschaften **SUBSCOPE** und **PUBSCOPE** zu steuern. Siehe „Verteilte Publish/Subscribe-Netze konfigurieren” auf Seite 471.

**Zugehörige Konzepte**

Kombinieren von Veröffentlichungs-und Subskriptionsbereichen

Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen

Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

**Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren
Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

Clusterthemendefinition in einen anderen WS-Manager verschieben
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen
In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthemenobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen
Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

**Zugehörige Verweise**

Datenströme und Themen
DISPLAY PUBSUB
Publish/Subscribe-Messaging

## Verbindung zu einem WS-Manager aus einer Publish/Subscribe-Hierarchie trennen

Trennen Sie einen untergeordneten WS-Manager von einem übergeordneten Warteschlangenmanager in einer Publish/Subscribe-Hierarchie.

### Informationen zu diesem Vorgang

Mit dem Befehl **ALTER QMGR** können Sie die Verbindung zwischen einem Warteschlangenmanager und einer Brokerhierarchie trennen. Sie können einen WS-Manager jederzeit in beliebiger Reihenfolge trennen.

Die entsprechende Anforderung zum Aktualisieren des übergeordneten Elements wird gesendet, wenn die Verbindung zwischen den Warteschlangenmanagern ausgeführt wird.

### Vorgehensweise

```
ALTER QMGR PARENT( '')
```

**Beispiel**

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
    1 : alter qmgr parent('')
AMQ8005: IBM MQ queue manager changed.
    2 : display pubsub type(child)
AMQ8147: IBM MQ object  not found.
display pubsub type(parent)
```

```
    3 : display pubsub type(parent)
AMQ8147: IBM MQ object not found.
```

## Nächste Schritte

Sie können alle Datenströme, Warteschlangen und manuell definierten Kanäle, die nicht mehr benötigt werden, löschen.

**Zugehörige Konzepte**

Kombinieren von Veröffentlichungs-und Subskriptionsbereichen
Ab IBM WebSphere MQ 7.0 arbeiten Veröffentlichungs- und Subskriptionsbereich unabhängig voneinander, um den Fluss von Veröffentlichungen zwischen Warteschlangenmanagern zu bestimmen.

Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen
Kombinieren Sie den Topic-Bereich eines Warteschlangenmanagers mit anderen WS-Managern in einem Publish/Subscribe-Cluster oder einer Hierarchie. Kombinieren Sie Publish/Subscribe-Cluster und Publish/Subscribe-Cluster mit Hierarchien.

**Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren
Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, setzen Sie die Eigenschaft **CLUSTER**. Legen Sie die Eigenschaft **CLROUTE** fest, um das Routing für Veröffentlichungen und Subskriptionen für dieses Thema auszuwählen.

Clusterthemendefinition in einen anderen WS-Manager verschieben
Für einen Themenhost oder direkte Routing-Cluster müssen Sie möglicherweise eine Clusterthemendefinition bei der Stilllegung eines Warteschlangenmanagers verschieben oder weil ein Clusterwarteschlangenmanager für einen signifikanten Zeitraum nicht verfügbar ist oder nicht verfügbar ist.

Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen
In einem Publish/Subscribe-Cluster in einem Topic-Host können mehrere Warteschlangenmanager verwendet werden, um Veröffentlichungen an Subskriptionen weiterzuleiten, indem sie dasselbe Clusterthemenobjekt auf diesen Warteschlangenmanagern definieren. Dies kann zur Verbesserung der Verfügbarkeit und des Lastausgleichs verwendet werden. Wenn Sie einen zusätzlichen Topic-Host für dasselbe Clusterthemenobjekt hinzufügen, können Sie mit dem Parameter **PUB** steuern, wann Veröffentlichungen über den neuen Topic-Host weitergeleitet werden.

WS-Manager mit einer Publish/Subscribe-Hierarchie verbinden
Sie verbinden den untergeordneten WS-Manager mit dem übergeordneten Warteschlangenmanager in der Hierarchie. Wenn der untergeordnete WS-Manager bereits Mitglied einer anderen Hierarchie oder eines anderen Clusters ist, verknüpft diese Verbindung die Hierarchien miteinander oder verknüpft den Cluster mit der Hierarchie.

# ALW Mehrere Installationen konfigurieren

Wenn Sie mehrere Installationen auf demselben System verwenden, müssen Sie die -Installationen und -Warteschlangenmanager konfigurieren.

## Informationen zu diesem Vorgang

Diese Informationen beziehen sich auf AIX, Linux, and Windows.

## Prozedur

- Verwenden Sie die Informationen in den folgenden Links, um Ihre -Installationen zu konfigurieren:
  - „Primäre Installation ändern” auf Seite 498
  - „WS-Manager einer Installation zuordnen” auf Seite 499
  - „Anwendungen in einer Umgebung mit mehreren Installationen verbinden” auf Seite 490

# ALW Anwendungen in einer Umgebung mit mehreren Installationen verbinden

Wenn auf AIX, Linux, and Windows -Systemen IBM MQ -Bibliotheken geladen werden, verwendet IBM MQ automatisch die entsprechenden Bibliotheken, ohne dass weitere Aktionen erforderlich sind. IBM MQ verwendet Bibliotheken aus der Installation, die dem Warteschlangenmanager zugeordnet ist, zu dem die Anwendung eine Verbindung herstellt.

Die folgenden Konzepte erläutern die Verbindungsherstellung von Anwendungen zu IBM MQ:

**Verlinken**
Wenn die Anwendung kompiliert wird, wird sie mit den IBM MQ-Bibliotheken verbunden, um die Funktionsexporte abzurufen, die dann bei Ausführung der Anwendung geladen werden.

**wird geladen**
Bei Ausführung der Anwendung werden die IBM MQ-Bibliotheken lokalisiert und geladen. Der spezifische Mechanismus, der zum Lokalisieren der Bibliotheken verwendet wird, variiert je nach Betriebssystem und wie die Anwendung erstellt wird. Weitere Informationen zum Suchen und Laden von Bibliotheken in einer Umgebung mit mehreren Installationsumgebungen finden Sie in „IBM MQ-Bibliotheken laden” auf Seite 491.

**Verbindung wird hergestellt**
Wenn die Anwendung beispielsweise mit dem Aufruf `MQCONN` oder `MQCONNX` eine Verbindung zu einem aktiven Warteschlangenmanager herstellt, verwendet sie dazu die geladenen IBM MQ-Bibliotheken.

Wenn eine Serveranwendung eine Verbindung zu einem WS-Manager herstellt, müssen die geladenen Bibliotheken aus der Installation stammen, die dem Warteschlangenmanager zugeordnet ist. Bei mehreren Installationen auf einem System bringt diese Einschränkung neue Herausforderungen bei der Auswahl des Mechanismus mit sich, mit dem das Betriebssystem die zu ladenden IBM MQ-Bibliotheken lokalisiert:

- Wenn der Befehl **setmqm** verwendet wird, um die einem WS-Manager zugeordnete Installation zu ändern, ändern sich die Bibliotheken, die geladen werden müssen.
- Wenn eine Anwendung eine Verbindung zu mehreren WS-Managern herstellt, deren Eigner verschiedene Installationen sind, müssen mehrere Gruppen von Bibliotheken geladen werden.

Wenn jedoch IBM MQ-Bibliotheken gefunden und geladen werden, lädt und verwendet IBM MQ die entsprechenden Bibliotheken, ohne dass Sie weitere Maßnahmen ergreifen müssen. Wenn die Anwendung eine Verbindung zu einem Warteschlangenmanager herstellt, lädt IBM MQ Bibliotheken aus der Installation, welcher der Warteschlangenmanager zugeordnet ist.

Die Migrationsszenarios sowie die Verbindung von Anwendungen mit mehreren Installationen werden im Abschnitt Koexistenz mehrerer Warteschlangenmanager unterschiedlicher Installationen unter AIX, Linux, and Windows ausführlicher behandelt.

Weitere Informationen zum Laden von Bibliotheken von IBM MQ finden Sie im Abschnitt „IBM MQ-Bibliotheken laden” auf Seite 491.

## Unterstützung und Einschränkungen

Wenn eine der folgenden IBM MQ -Bibliotheken gefunden und geladen wurde, kann das Produkt die entsprechenden Bibliotheken automatisch laden und verwenden:

- Die C-Serverbibliotheken
- Die C + + -Serverbibliotheken
- Die XA-Serverbibliotheken
- Die COBOL-Serverbibliotheken
- Die COM + -Serverbibliotheken
- .NET im nicht verwalteten Modus

IBM MQ lädt und verwendet außerdem automatisch die entsprechenden Bibliotheken für Java- und JMS-Anwendungen im Bindungsmodus.

Es gibt eine Reihe von Einschränkungen für Anwendungen, die mehrere Installationen verwenden. Weitere Informationen finden Sie unter „Einschränkungen für Anwendungen mit mehreren Installationen” auf Seite 494.

**Zugehörige Konzepte**

„Einschränkungen für Anwendungen mit mehreren Installationen” auf Seite 494
Bei der Verwendung von CICS-Serverbibliotheken, Direktaufrufverbindungen, Nachrichtenhandles und Exits in einer Umgebung mit Mehrfachinstallation sind Einschränkungen zu beachten.

„IBM MQ-Bibliotheken laden” auf Seite 491
Bei der Entscheidung, wie IBM MQ-Bibliotheken geladen werden, müssen Sie eine Reihe von Faktoren berücksichtigen, unter anderem Ihre Umgebung, ob Sie Ihre bestehenden Anwendungen ändern können, ob Sie eine primäre Installation möchten, wo IBM MQ installiert ist und ob sich der Speicherort von IBM MQ möglicherweise ändert.

**Zugehörige Tasks**

Primäre Installation auswählen

„Primäre Installation ändern” auf Seite 498
Mit dem Befehl **setmqinst** können Sie eine Installation als primäre Installation festlegen bzw. ihre Festlegung aufheben.

„WS-Manager einer Installation zuordnen” auf Seite 499
Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den **crtmqm** -Befehl ausgegeben hat. Unter AIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

## ALW IBM MQ-Bibliotheken laden

Bei der Entscheidung, wie IBM MQ-Bibliotheken geladen werden, müssen Sie eine Reihe von Faktoren berücksichtigen, unter anderem Ihre Umgebung, ob Sie Ihre bestehenden Anwendungen ändern können, ob Sie eine primäre Installation möchten, wo IBM MQ installiert ist und ob sich der Speicherort von IBM MQ möglicherweise ändert.

Wie IBM MQ-Bibliotheken lokalisiert und geladen werden, hängt von der Installationsumgebung ab:

- Wenn auf AIX and Linux -Systemen eine Kopie einer IBM MQ-Version an der Standardposition installiert wird, funktionieren vorhandene Anwendungen weiterhin auf dieselbe Weise wie frühere Versionen. Wenn die Anwendungen jedoch symbolische Links in `/usr/lib`benötigen, müssen Sie entweder eine Installation der IBM MQ -Version als primäre Installation auswählen oder die symbolischen Links manuell erstellen.
- Wenn IBM MQ nicht an der Standardposition installiert ist, müssen Sie möglicherweise Ihre vorhandenen Anwendungen ändern, damit die richtigen Bibliotheken geladen werden.

Wie IBM MQ-Bibliotheken gesucht und geladen werden, hängt auch davon ab, wie die bestehenden Anwendungen zum Laden von Bibliotheken eingerichtet sind. Weitere Informationen zum Laden von Bibliotheken finden Sie unter „Lademechanismen für die Betriebssystembibliothek” auf Seite 493.

Optimalerweise sollten Sie sicherstellen, dass die IBM MQ-Bibliothek, die vom Betriebssystem geladen wird, die Bibliothek ist, der der WS-Manager zugeordnet ist.

Die Lademethoden für IBM MQ-Bibliotheken variieren auch je nach Plattform. Jede Methode hat dabei ihre Vor- und ihre Nachteile.

*Tabelle 28. Vorteile und Nachteile der Optionen zum Laden von Bibliotheken*

| Plattform | Option | Vorteile | Rückzugsschrägen |
|---|---|---|---|
| Linux<br>AIX<br>AIX and Linux-Systeme | Legen Sie den eingebetteten Laufzeitsuchpfad (RPath) der Anwendung fest oder ändern Sie diesen.<br>Mit dieser Option müssen Sie die Anwendung erneut kompilieren und verknüpfen. Weitere Informationen zum Kompilieren und Verknüpfen von Anwendungen finden Sie unter Erstellen einer prozeduralen Anwendung . | • Der Umfang der Änderung ist klar. | • Sie müssen in der Lage sein, die Anwendung erneut zu kompilieren und zu verlinken.<br>• Wenn sich die Speicherposition von IBM MQ ändert, müssen Sie den Laufzeit-Suchpfad ändern. |
| Systeme mit AIX and Linux | Setzen Sie die Umgebungsvariable *LD_LIBRARY_PATH* unter Verwendung von setmqenv oder crtmqenv mit der Option **-k** oder **-1**. (<br>AIX Unter AIX ist diese Umgebungsvariable *LIBPATH* | • Es sind keine Änderungen an vorhandenen Anwendungen erforderlich.<br>• Überschreibt eingebettete RPaths in einer Anwendung.<br>• Die Variable lässt sich leicht ändern, wenn sich die Speicherposition von IBM MQ ändert. | • `setuid` -und `setgid` -Anwendungen oder Anwendungen, die auf andere Weise erstellt wurden, können *LD_LIBRARY_PATH* aus Sicherheitsgründen ignorieren.<br>• Umgebungsspezifisch, daher muss in jeder Umgebung, in der die Anwendung ausgeführt wird, festgelegt werden.<br>• Mögliche Auswirkungen auf andere Anwendungen, die auf *LD_LIBRARY_PATH* basieren.<br>• Linux Linux: Der zum Erstellen der Anwendung verwendete Compiler könnte die Verwendung von *LD_LIBRARY_PATH* inaktivieren. Weitere Informationen finden Sie im Abschnitt Überlegungen zur Laufzeitverknüpfung für Linux. |
| Windows<br>Windows-Systeme | Legen Sie die Variable PATH mit setmqenv oder crtmqenv fest. | • Für vorhandene Anwendungen sind keine Änderungen erforderlich.<br>• Die Variable lässt sich leicht ändern, wenn sich die Speicherposition von IBM MQ ändert. | • Umgebungsspezifisch, daher muss in jeder Umgebung, in der die Anwendung ausgeführt wird, festgelegt werden.<br>• Mögliche Auswirkungen auf andere Anwendungen. |

| Tabelle 28. Vorteile und Nachteile der Optionen zum Laden von Bibliotheken (Forts.) | | | |
|---|---|---|---|
| **Plattform** | **Option** | **Vorteile** | **Rückzugsschrägen** |
| ALW<br>AIX, Linux, and Windows-Systeme | Setzen Sie die primäre Installation auf IBM MQ oder eine höhere Installation fest. Siehe „Primäre Installation ändern” auf Seite 498.<br>Weitere Informationen zur primären Installation finden Sie im Abschnitt Primäre Installation auswählen . | • Für vorhandene Anwendungen sind keine Änderungen erforderlich.<br>• Die primäre Installation lässt sich leicht ändern, wenn sich die Speicherposition von IBM MQ ändert.<br>• Ähnliches Verhalten für vorherige Versionen von IBM MQ. | • Linux AIX AIX and Linux: funktioniert nicht, wenn `/usr/lib` nicht im Standardsuchpfad enthalten ist. |

## Überlegungen zum Laden von Bibliotheken in Linux

Linux

Anwendungen, die mit einigen Versionen von gcc, z. B. Version 3.2.x kompiliert wurden, können über einen eingebetteten RPath verfügen, der nicht mit der Umgebungsvariablen *LD_LIBRARY_PATH* überschrieben werden kann. Mit dem Befehl `readelf -d` *`applicationName`* können Sie feststellen, ob eine Anwendung betroffen ist. Der RPath kann nicht überschrieben werden, wenn das RPATH-Symbol vorhanden ist und das RUNPATH-Symbol nicht vorhanden ist.

## Lademechanismen für die Betriebssystembibliothek

Auf Windows-Systemen werden mehrere Verzeichnisse durchsucht, um die Bibliotheken zu finden:

- Das Verzeichnis, aus dem die Anwendung geladen wird.
- Das aktuelle Verzeichnis.
- Die Verzeichnisse in der Umgebungsvariablen *PATH* , sowohl die globale *PATH* -Variable als auch die *PATH* -Variable des aktuellen Benutzers.

Linux AIX Auf AIX and Linux-Systemen können verschiedene Methoden zur Lokalisierung der zu ladenden Bibliotheken verwendet worden sein:

- Verwenden Sie die Umgebungsvariable *LD_LIBRARY_PATH* (auch *LIBPATH* auf AIX). Wenn diese Variable gesetzt ist, definiert sie eine Gruppe von Verzeichnissen, die nach den erforderlichen IBM MQ-Bibliotheken durchsucht werden. Wenn in diesen Verzeichnissen Bibliotheken gefunden werden, werden sie bevorzugt für alle Bibliotheken verwendet, die mit den anderen Methoden gefunden werden können.
- Verwenden eines eingebetteten Suchpfads (RPath). Die Anwendung kann eine Reihe von Verzeichnissen enthalten, die nach den IBM MQ-Bibliotheken durchsucht werden. Wenn der *LD_LIBRARY_PATH* nicht definiert ist oder wenn die erforderlichen Bibliotheken nicht mit der Variablen gefunden wurden, wird der RPath für die Bibliotheken durchsucht. Wenn Ihre bestehenden Anwendungen einen Laufzeit-Suchpfad verwenden, Sie die Anwendung jedoch nicht neu kompilieren und verlinken können, müssen Sie entweder IBM MQ an der Standardposition installieren oder mit einer anderen Methode nach den Bibliotheken suchen.
- Der Standardbibliothekspfad wird verwendet. Wenn die IBM MQ-Bibliotheken nicht gefunden werden, nachdem Sie die *LD_LIBRARY_PATH*-Variablen und die RPath-Positionen durchsucht haben, wird der Standardbibliothekspfad durchsucht. In der Regel enthält dieser Pfad `/usr/lib` oder `/usr/lib64`. Wenn die Bibliotheken nach dem Durchsuchen des Standardbibliothekspfads nicht gefunden werden, kann die Anwendung aufgrund fehlender Abhängigkeiten nicht gestartet werden.

Sie können Betriebssystemmechanismen verwenden, um zu ermitteln, ob Ihre Anwendungen über einen eingebetteten Suchpfad verfügen. For example:

- AIX: **dump**
- Linux: **readelf**

**Zugehörige Konzepte**

„Einschränkungen für Anwendungen mit mehreren Installationen” auf Seite 494
Bei der Verwendung von CICS-Serverbibliotheken, Direktaufrufverbindungen, Nachrichtenhandles und Exits in einer Umgebung mit Mehrfachinstallation sind Einschränkungen zu beachten.

„Anwendungen in einer Umgebung mit mehreren Installationen verbinden” auf Seite 490
Wenn auf AIX, Linux, and Windows -Systemen IBM MQ -Bibliotheken geladen werden, verwendet IBM MQ automatisch die entsprechenden Bibliotheken, ohne dass weitere Aktionen erforderlich sind. IBM MQ verwendet Bibliotheken aus der Installation, die dem Warteschlangenmanager zugeordnet ist, zu dem die Anwendung eine Verbindung herstellt.

**Zugehörige Tasks**

Primäre Installation auswählen

„Primäre Installation ändern” auf Seite 498
Mit dem Befehl **setmqinst** können Sie eine Installation als primäre Installation festlegen bzw. ihre Festlegung aufheben.

„WS-Manager einer Installation zuordnen” auf Seite 499
Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den **crtmqm** -Befehl ausgegeben hat. Unter AIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

# ALW Einschränkungen für Anwendungen mit mehreren Installationen

Bei der Verwendung von CICS-Serverbibliotheken, Direktaufrufverbindungen, Nachrichtenhandles und Exits in einer Umgebung mit Mehrfachinstallation sind Einschränkungen zu beachten.

## CICS-Serverbibliotheken

Wenn Sie die CICS-Serverbibliotheken verwenden, wählt IBM MQ nicht automatisch die richtige Bibliotheksversion aus. Sie müssen Ihre Anwendungen mit der entsprechenden Bibliotheksebene für den Warteschlangenmanager kompilieren und verknüpfen, zu dem die Anwendung eine Verbindung herstellt. Weitere Informationen finden Sie unter Bibliotheken für die Verwendung mit TXSeries for Multiplatforms Version 5 erstellen.

## Nachrichtenkennungen

Nachrichtenkennungen, die den Sonderwert MQHC_UNASSOCIATED_HCONN verwenden, sind begrenzt auf die Verwendung mit der ersten Installation, die in einem Prozess geladen wird. Wenn die Nachrichtenkennung nicht von einer bestimmten Installation verwendet werden kann, wird der Ursachencode MQRC_HMSG_NOT_AVAILABLE zurückgegeben.

Diese Einschränkung betrifft Nachrichteneigenschaften. Sie können keine Nachrichtenkennungen verwenden, um Nachrichteneigenschaften von einem Warteschlangenmanager in einer Installation abzurufen und sie in einen Warteschlangenmanager in einer anderen Installation zu versetzen. Weitere Informationen zu Nachrichtenkennungen finden Sie in MQCRTMH-Create message handle .

## Exits

In einer Umgebung mit mehreren Installationen müssen vorhandene Exits für die Verwendung mit IBM MQ -Installationen aktualisiert werden. Datenkonvertierungsexits, die mit dem Befehl **crtmqcvx** generiert werden, müssen mit dem aktualisierten Befehl neu generiert werden.

Alle Exits müssen unter Verwendung der Struktur MQIEP geschrieben werden, dürfen keinen integrierten Laufzeit-Suchpfad (RPATH) zur Suche nach den IBM MQ-Bibliotheken verwenden und können keine Ver-

bindung zu den IBM MQ-Bibliotheken herstellen. Weitere Informationen finden Sie im Abschnitt Exits und installierbare Services in AIX, Linux, and Windows schreiben.

## Direktaufruf

Auf einem Server mit mehreren Installationen müssen Anwendungen, die eine Direktaufrufverbindung zu IBM MQ verwenden, die folgenden Regeln befolgen:

1. Der Warteschlangenmanager muss mit derselben Installation verknüpft sein, aus der die Anwendung die Laufzeitbibliotheken von IBM MQ geladen hat. Die Anwendung darf keine Direktaufrufverbindung zu einem Warteschlangenmanager verwenden, der mit einer anderen Installation verknüpft ist. Ein Versuch, die Verbindung herzustellen, führt zu einem Fehler mit Ursachencode `MQRC_INSTALLATION_MISMATCH`.
2. Die Verbindungsherstellung über einen Nicht-Direktaufruf mit einem Warteschlangenmanager, der mit derselben Installation verknüpft ist, aus der die Anwendung die Laufzeitbibliotheken von IBM MQ geladen hat, verhindert die Verbindungsherstellung der Anwendung per Direktaufruf – ausgenommen, eine der folgenden Bedingungen wird erfüllt:
   - Die Anwendung stellt ihre erste Verbindung zu einem Warteschlangenmanager, der derselben Installation zugeordnet ist, als Direktaufrufverbindung her.
   - Die Umgebungsvariable `AMQ_SINGLE_INSTALLATION` ist festgelegt.
3. Das Herstellen einer Verbindung zu einem Warteschlangenmanager, der einer IBM MQ -Installation zugeordnet ist, ohne Direktaufruf hat keine Auswirkung darauf, ob eine Anwendung eine Verbindung herstellen kann.

Mit der Einstellung `AMQ_SINGLE_INSTALLATION` können Sie jede Verbindung zu einem Warteschlangenmanager als Direktaufrufverbindung herstellen. Anderenfalls gelten nahezu dieselben Einschränkungen:

- Die Installation muss dieselbe sein, von der aus die Laufzeitbibliotheken von IBM MQ geladen wurden.
- Jede Verbindung im demselben Prozess muss zu derselben Installation hergestellt werden. Wenn Sie versuchen, eine Verbindung zu einem Warteschlangenmanager herzustellen, der einer anderen Installation zugeordnet ist, schlägt die Verbindung mit dem Ursachencode `MQRC_INSTALLATION_MISMATCH` fehl. Wenn `AMQ_SINGLE_INSTALLATION` festgelegt ist, gilt diese Einschränkung für alle Verbindungen, nicht nur für Direktaufrufverbindungen.
- Stellen Sie Direktaufrufverbindungen nur zu einem einzigen Warteschlangenmanager her.

**Zugehörige Verweise**

MQCONNX – Verbindung mit Warteschlangenmanager herstellen (erweitert)
MQIEP-Struktur
2583 (0A17) (RC2583): MQRC_INSTALLATION_MISMATCH
2587 (0A1B) (RC2587): MQRC_HMSG_NOT_AVAILABLE
2590 (0A1E) (RC2590): MQRC_FASTPATH_NOT_AVAILABLE

## ALW .NET-Anwendungen in einer Umgebung mit mehreren Installationen verbinden

Standardmäßig verwenden Anwendungen die .NET-Assemblys von der primären Installation. Wenn keine primäre Installation vorhanden ist oder Sie die primären Installationsassemblys nicht verwenden möchten, müssen Sie die Anwendungskonfigurationsdatei oder die *DEVPATH* -Umgebungsvariable aktualisieren.

Wenn auf dem System eine primäre Installation vorhanden ist, werden die .NET-Assemblys und die Richtliniendateien dieser Installation im globalen Assemblycache (Global Assembly Cache, GAC) registriert. Die .NET-Assemblys für alle anderen Installationen befinden sich im Installationspfad jeder Installation, aber die Assemblys sind nicht in der GAC registriert. Daher werden Anwendungen standardmäßig mit den .NET-Assemblys aus der primären Installation ausgeführt. Sie müssen die Anwendungskonfigurationsdatei aktualisieren, wenn einer der folgenden Fälle zutrifft:

- Sie verfügen nicht über eine primäre Installation.
- Sie möchten, dass die Anwendung die primären Installationsassemblys nicht verwendet.
- Die IBM MQ-Version der primären Installation ist niedriger als die Version, mit der die Anwendung kompiliert wurde.

Informationen zum Aktualisieren der Anwendungskonfigurationsdatei finden Sie unter „.NET-Anwendungen mit der Anwendungskonfigurationsdatei verbinden" auf Seite 496.

Sie müssen die Umgebungsvariable *DEVPATH* aktualisieren, wenn der folgende Fall wahr ist:

- Sie möchten, dass Ihre Anwendung die Assemblys von einer nicht primären Installation aus verwendet, die primäre Installation jedoch dieselbe Version wie die nicht primäre Installation hat.

Weitere Informationen zum Aktualisieren der Variablen *DEVPATH* finden Sie unter „.NET-Anwendungen mit DEVPATH verbinden" auf Seite 497.

## .NET-Anwendungen mit der Anwendungskonfigurationsdatei verbinden

Innerhalb der Anwendungskonfigurationsdatei müssen Sie verschiedene Tags für die Umleitung von Anwendungen festlegen, um Assemblys verwenden zu können, die nicht von der primären Installation aus verwendet werden.

In der folgenden Tabelle sind die spezifischen Änderungen aufgeführt, die an der Anwendungskonfigurationsdatei vorgenommen werden müssen, damit .NET-Anwendungen eine Verbindung mit bestimmten Assemblys herstellen können:

*Tabelle 29. Anwendungen für die Verwendung bestimmter Baugruppen konfigurieren*

| | **Anwendungen, die mit einer früheren Version von IBM MQ kompiliert wurden** | **Anwendungen, die mit einer späteren Version von IBM MQ kompiliert wurden** |
|---|---|---|
| So führen Sie eine Anwendung mit einer späteren IBM MQ-Primärinstallation aus. (spätere Versionsbaugruppen in GAC): | Keine Änderungen erforderlich | Keine Änderungen erforderlich |
| Gehen Sie wie folgt vor, um eine Anwendung mit einer früheren IBM MQ-Primärinstallation auszuführen. (frühere Versionsbaugruppen in GAC): | Keine Änderungen erforderlich | In der Anwendungskonfigurationsdatei:<br>• Verwenden Sie den Tag *bindingRedirect* , um die Verwendung der früheren Version der Assemblys anzugeben, die sich in der GAC befinden. |
| So führen Sie eine Anwendung mit einer späteren Version von IBM MQ nicht der primären Installation aus. (spätere Versionsbaugruppen im Installationsordner): | In der Anwendungskonfigurationsdatei:<br>• Verwenden Sie den Tag *codebase* , um auf die Position der späteren Assemblys zu verweisen.<br>• Verwenden Sie den Tag *bindingRedirect* , um die Verwendung der späteren Assemblys anzugeben. | In der Anwendungskonfigurationsdatei:<br>• Verwenden Sie den Tag *codebase* , um auf die Position der späteren Assemblys zu verweisen. |

| *Tabelle 29. Anwendungen für die Verwendung bestimmter Baugruppen konfigurieren (Forts.)* | | |
|---|---|---|
| | **Anwendungen, die mit einer früheren Version von IBM MQ kompiliert wurden** | **Anwendungen, die mit einer späteren Version von IBM MQ kompiliert wurden** |
| So führen Sie eine Anwendung mit einer früheren Version von IBM MQ nicht der primären Installation aus. (ältere Versionsbaugruppen im Installationsordner): | In der Anwendungskonfigurationsdatei:<br>• Verwenden Sie den Tag *codebase* , um auf die Position der früheren Versionsbaugruppen zu verweisen.<br>• Tag *publisherpolicy Apply=no* einschließen | In der Anwendungskonfigurationsdatei:<br>• Verwenden Sie den Tag *codebase* , um auf die Position der früheren Versionsbaugruppen zu verweisen.<br>• Verwenden Sie den Tag *bindingRedirect* , um die Verwendung der früheren Versionsbaugruppen anzugeben.<br>• Tag *publisherpolicy Apply=no* einschließen |

Eine Musteranwendungskonfigurationsdatei `NonPrimaryRedirect.config` ist im Ordner *MQ_INSTALLATION_PATH*`\tools\dotnet\samples\base` enthalten. Diese Datei können Sie mit dem IBM MQ-Installationspfad einer beliebigen nicht-primären Installation ändern. Die Datei kann auch mit dem Tag *linkedConfiguration* direkt in andere Konfigurationsdateien aufgenommen werden. Es werden Muster für `nmqsget.exe.config` und `nmqsput.exe.config` bereitgestellt. Beide Beispiele verwenden den Tag *linkedConfiguration* und schließen die Datei `NonPrimaryRedirect.config` ein.

## .NET-Anwendungen mit DEVPATH verbinden

Sie können die Assemblys mit der Umgebungsvariablen *DEVPATH* finden. Die durch die Variable *DEVPATH* angegebenen Assemblys werden bevorzugt für alle Baugruppen in der GAC verwendet. Weitere Informationen zur Verwendung dieser Variablen finden Sie in der entsprechenden Microsoft-Dokumentation unter *DEVPATH*.

Um die Assemblys mit der Umgebungsvariablen *DEVPATH* zu suchen, müssen Sie die Variable *DEVPATH* auf den Ordner setzen, der die zu verwendenden Assemblys enthält. Anschließend müssen Sie die Anwendungskonfigurationsdatei aktualisieren und die folgenden Laufzeitkonfigurationsdaten hinzufügen:

```
<configuration>
<runtime>
<developmentMode developerInstallation="true" />
</runtime>
</configuration>
```

**Zugehörige Konzepte**

„Anwendungen in einer Umgebung mit mehreren Installationen verbinden” auf Seite 490
Wenn auf AIX, Linux, and Windows -Systemen IBM MQ -Bibliotheken geladen werden, verwendet IBM MQ automatisch die entsprechenden Bibliotheken, ohne dass weitere Aktionen erforderlich sind. IBM MQ verwendet Bibliotheken aus der Installation, die dem Warteschlangenmanager zugeordnet ist, zu dem die Anwendung eine Verbindung herstellt.

Mehrere Installationen

**Zugehörige Tasks**

Primäre Installation auswählen

.NET verwenden

# ALW Primäre Installation ändern

Mit dem Befehl **setmqinst** können Sie eine Installation als primäre Installation festlegen bzw. ihre Festlegung aufheben.

## Informationen zu diesem Vorgang

Diese Task gilt für AIX, Linux, and Windows.

Bei der primären Installation handelt es sich um die Installation, auf die systemweite Standorte verweisen. Weitere Informationen zur primären Installation sowie Hinweise zur Auswahl der primären Installation finden Sie im Abschnitt Primäre Installation auswählen .

Windows Während des Installationsprozesses unter Windows können Sie angeben, dass die Installation die primäre Installation sein soll.

Linux AIX Auf AIX and Linux-Systemen müssen Sie nach der Installation den Befehl **setmqinst** ausgeben, um die Installation als die primäre Installation festzulegen.

## Prozedur

- Gehen Sie wie folgt vor, um eine Installation als primäre Installation festzulegen:
  a) Überprüfen Sie, ob eine Installation bereits die primäre Installation ist, indem Sie den folgenden Befehl eingeben:

     ```
     MQ_INSTALLATION_PATH/bin/dspmqinst
     ```

     Dabei ist *MQ_INSTALLATION_PATH* der Installationspfad einer IBM MQ -Installation.
  b) Wenn eine vorhandene IBM MQ -Installation als primäre Installation festgelegt ist, heben Sie die Festlegung auf , bevor Sie mit dem nächsten Schritt fortfahren.
  c) Stellen Sie sicher, dass Sie mit der entsprechenden Berechtigung angemeldet sind:
     – Linux AIX Als Root in AIX and Linux.
     – Windows Als Mitglied der Gruppe "Administratoren" auf Windows-Systemen.
  d) Geben Sie einen der folgenden Befehle ein:
     – Gehen Sie wie folgt vor, um die primäre Installation mithilfe des Pfads der Installation festzulegen, die die primäre Installation sein soll:

       ```
       MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
       ```

     – Gehen Sie wie folgt vor, um die primäre Installation mit dem Namen der Installation festzulegen, die die primäre Installation sein soll:

       ```
       MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
       ```

  e) Windows
     Führen Sie auf Windows-Systemen einen Neustart des Systems durch.
- Gehen Sie wie folgt vor, um eine Installation als primäre Installation zu dekonfigurieren:
  a) Überprüfen Sie, welche Installation die primäre Installation ist, indem Sie den folgenden Befehl eingeben:

     ```
     MQ_INSTALLATION_PATH/bin/dspmqinst
     ```

     Dabei ist *MQ_INSTALLATION_PATH* der Installationspfad einer IBM MQ -Installation.

b) Stellen Sie sicher, dass Sie mit der entsprechenden Berechtigung angemeldet sind:

- Linux AIX Als Root in AIX and Linux.
- Windows Als Mitglied der Gruppe "Administratoren" auf Windows-Systemen.

- Geben Sie einen der folgenden Befehle ein:
  - Gehen Sie wie folgt vor, um die primäre Installation mithilfe des Pfads der Installation zu dekonfigurieren, die Sie nicht mehr als primäre Installation verwenden möchten:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

  - Gehen Sie wie folgt vor, um die primäre Installation unter Verwendung des Namens der Installation zu dekonfigurieren, die Sie nicht mehr als primäre Installation verwenden möchten:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

**Zugehörige Tasks**

Deinstallieren, Durchführen eines Upgrades und Wartung der primären Installation

Auswählen eines Installationsnamens

**Zugehörige Verweise**

Komponenten, die nur mit der primären Installation unter Windows verwendet werden können

Verknüpfungen von externen Speicherarchiven und Steuerbefehlen zur primären Installation von AIX and Linux

setmqinst

## ALW WS-Manager einer Installation zuordnen

Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den **crtmqm** -Befehl ausgegeben hat. Unter AIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

### Informationen zu diesem Vorgang

Die Installation, der ein Warteschlangenmanager zugeordnet ist, schränkt den Warteschlangenmanager ein, sodass er nur mit Befehlen verwaltet werden kann, die in dieser Installation ausgegeben werden. Es gibt drei wichtige Ausnahmen:

- **setmqm** ändert die dem Warteschlangenmanager zugeordnete Installation. Dieser Befehl muss von der Installation abgesetzt werden, die Sie dem Warteschlangenmanager zuordnen möchten, nicht die Installation, der der WS-Manager derzeit zugeordnet ist. Der mit dem Befehl **setmqm** angegebene Installationsname muss mit der Installation übereinstimmen, von der der Befehl abgesetzt wird.
- **strmqm** muss von der Installation ausgegeben werden, die dem Warteschlangenmanager zugeordnet ist.
- **dspmq** zeigt Informationen zu allen Warteschlangenmanagern auf einem System an, nicht nur zu den Warteschlangenmanagern, die derselben Installation zugeordnet sind wie der Befehl **dspmq** . Der Befehl `dspmq -o installation` zeigt Informationen zu den Warteschlangenmanagern an, denen die Installationen zugeordnet sind.

Bei HA-Umgebungen ordnet der Befehl **addmqinf** den WS-Manager automatisch der Installation zu, von der der Befehl **addmqinf** ausgegeben wird. Solange der Befehl **strmqm** von derselben Installation wie der **addmqinf** -Befehl ausgegeben wird, ist keine weitere Konfiguration erforderlich. Um den Warteschlangenmanager mit einer anderen Installation zu starten, müssen Sie zuerst die zugehörige Installation mit dem **setmqm** -Befehl ändern.

Wenn Sie einen Warteschlangenmanager einer Installation zuordnen möchten, können Sie den Befehl **setmqm** wie folgt verwenden:

- Verschieben einzelner Warteschlangenmanager zwischen funktional entsprechenden Versionen von IBM MQ. Beispiel: Verschieben eines Warteschlangenmanagers von einem Test in ein Produktionssystem.
- Migration einzelner Warteschlangenmanager von einer älteren Version von IBM MQ auf eine neuere Version von IBM MQ. Die Migration von Warteschlangenmanagern zwischen Versionen hat verschiedene Auswirkungen, die Sie kennen müssen. Weitere Informationen zur Migration finden Sie im Abschnitt Verwalten und Migrieren .

## Vorgehensweise

1. Stoppen Sie den WS-Manager mit dem Befehl **endmqm** aus der Installation, die derzeit dem Warteschlangenmanager zugeordnet ist.
2. Ordnen Sie den Warteschlangenmanager mit dem Befehl **setmqm** aus dieser Installation einer anderen Installation zu.

   Geben Sie beispielsweise den folgenden Befehl in Installation2 ein, um den Warteschlangenmanager QMB einer Installation mit dem Namen Installation2 zu setzen:

   ```
   MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
   ```

   Dabei steht *MQ_INSTALLATION_PATH* für den Pfad, in dem Installation2 installiert ist.
3. Starten Sie den Warteschlangenmanager mit dem Befehl **strmqm** aus der Installation, die jetzt dem Warteschlangenmanager zugeordnet ist.

   Dieser Befehl führt alle erforderlichen Warteschlangenmanager-Migrationen aus und führt dazu, dass der WS-Manager betriebsbereit ist.

## Nächste Schritte

Wenn die Installation, der ein Warteschlangenmanager zugeordnet ist, gelöscht wurde oder die Statusinformationen des Warteschlangenmanagers nicht verfügbar sind, kann der Befehl **setmqm** den Warteschlangenmanager keiner anderen Installation zuordnen. Nehmen Sie in dieser Situation die folgenden Aktionen vor:

1. Mit dem Befehl **dspmqinst** können Sie die anderen Installationen auf Ihrem System anzeigen.
2. Ändern Sie das InstallationName -Feld der Zeilengruppe QueueManager in mqs.ini manuell, um eine andere Installation anzugeben.
3. Verwenden Sie den Befehl **dltmqm** aus dieser Installation, um den Warteschlangenmanager zu löschen.

**Zugehörige Konzepte**

„Installationen von IBM MQ auf einem System finden” auf Seite 501
Wenn Sie mehrere Installationen von IBM MQ auf einem System verwenden, können Sie überprüfen, welche Versionen an welcher Position installiert sind.

„IBM MQ-Konfigurationsdatei, mqs.ini” auf Seite 91
Die IBM MQ-Konfigurationsdatei mqs.ini enthält Informationen, die für alle Warteschlangenmanager auf dem Knoten relevant sind. Sie wird automatisch während der Installation erstellt.

**Zugehörige Tasks**

Primäre Installation auswählen

**Zugehörige Verweise**

addmqinf
dspmq
dspmqinst
endmqm
setmqm
strmqm

## ALW Installationen von IBM MQ auf einem System finden

Wenn Sie mehrere Installationen von IBM MQ auf einem System verwenden, können Sie überprüfen, welche Versionen an welcher Position installiert sind.

Mit folgenden Methoden finden Sie die IBM MQ-Installationen auf Ihrem System:

- Fragen Sie mithilfe der Plattforminstallationstools ab, wo IBM MQ installiert wurde. Verwenden Sie anschließend den Befehl **dspmqver** aus einer IBM MQ -Installation. Mit den folgenden Beispielbefehlen können Sie abfragen, wo IBM MQ installiert ist:
  - AIX Auf AIX-Systemen können Sie den Befehl **lslpp** verwenden:

    ```
    lslpp -R ALL -l mqm.base.runtime
    ```

  - Linux Auf Linux-Systemen können Sie den Befehl **rpm** verwenden:

    ```
    rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
    ```

  - Windows Auf Windows-Systemen können Sie den Befehl **wmic** verwenden. Mit diesem Befehl kann der wmic-Client installiert werden:

    ```
    wmic product where "(Name like '%MQ%') AND (not Name like '%bitSupport')" get Name, Versi
    on, InstallLocation
    ```

- Linux AIX Geben Sie auf AIX and Linux-Systemen den folgenden Befehl aus, um zu ermitteln, wo IBM MQ installiert wurde:

  ```
  cat /etc/opt/mqm/mqinst.ini
  ```

  Verwenden Sie anschließend den Befehl **dspmqver** aus einer IBM MQ -Installation.

- Windows Geben Sie den folgenden Befehl aus, um Details zu Installationen auf dem System auf der 32-Bit-Version von Windows anzuzeigen:

  ```
  reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
  ```

- Windows Geben Sie auf der 64-Bit-Version von Windows den folgenden Befehl aus:

  ```
  reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
  ```

**Zugehörige Verweise**

dspmqver
dspmqinst
Mehrere Installationen

# Hochverfügbarkeit, Wiederherstellung und Neustart konfigurieren

Sie können Ihre Anwendungen hoch verfügbar machen, indem Sie die Warteschlangenverfügbarkeit verwalten, wenn ein Warteschlangenmanager ausfällt und Nachrichten nach dem Server-oder Speicherausfall wiederhergestellt werden.

## Informationen zu diesem Vorgang

z/OS Unter z/OS wird die Hochverfügbarkeit in die Plattform integriert. Siehe Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange.

Multi Unter Multiplatforms können Sie die Verfügbarkeit der Clientanwendungen verbessern, indem Sie die Clientwiederverbindung verwenden, um einen Client automatisch zwischen einer Gruppe von Warteschlangenmanagern oder der neuen aktiven Instanz eines Multi-Instanz-Warteschlangenmanagers nach einem Fehler des Warteschlangenmanagers zu wechseln. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt. Ein Warteschlangenmanager mit mehreren Instanzen ist so konfiguriert, dass er als einzelner Warteschlangenmanager auf mehreren Servern ausgeführt wird. Serveranwendungen werden in diesem WS-Manager implementiert. Wenn der Server, auf dem die aktive Instanz ausgeführt wird, fehlschlägt, wird die Ausführung automatisch auf eine Standby-Instanz desselben Warteschlangenmanagers auf einem anderen Server umgeschaltet. Wenn Sie Serveranwendungen so konfigurieren, dass sie als WS-Manager-Services ausgeführt werden, werden sie erneut gestartet, wenn eine Standby-Instanz zur aktiven WS-Manager-Instanz wird.

Eine andere Möglichkeit, die Serveranwendungsverfügbarkeit auf Multiplatforms zu erhöhen, ist die Implementierung von Serveranwendungen auf mehreren Computern in einem WS-Manager-Cluster. Ab IBM WebSphere MQ 7.1 werden bei der Behebung von Clusterfehlern diejenigen Operationen, die zu Problemen geführt haben, erneut ausgeführt, bis die Probleme behoben sind. Weitere Informationen finden Sie unter Änderungen an der Clusterfehlerwiederherstellung auf anderen Servern als z/OS. Sie können IBM MQ for Multiplatforms auch als Teil einer plattformspezifischen Clustering-Lösung konfigurieren, z. B.:

- Microsoft Cluster-Server
- IBM i HA-Cluster unter IBM i
- Linux AIX PowerHA für AIX (früher HACMP unter AIX) und andere UNIX and Linux-Clusterlösungen

Linux Auf Linux-Systemen können Sie Warteschlangenmanager für replizierte Daten (RDQMs) für die Implementierung von Hochverfügbarkeits- oder Wiederherstellungslösungen konfigurieren. Für die Hochverfügbarkeit werden Instanzen des gleichen Warteschlangenmanagers auf jedem Knoten in einer Gruppe mit drei Linux-Servern konfiguriert. Eine der drei Instanzen ist die aktive Instanz. Die Daten aus dem aktiven Warteschlangenmanager werden synchron auf die beiden anderen Instanzen repliziert, sodass eine dieser Instanzen im Falle eines Fehlers die Überlassung übernehmen kann. Für die Wiederherstellung wird ein Warteschlangenmanager auf einem primären Knoten an einer Position ausgeführt, wobei sich eine sekundäre Instanz dieses Warteschlangenmanagers auf einem Wiederherstellungsknoten an einer anderen Position befindet. Daten werden zwischen der primären Instanz und der sekundären Instanz repliziert, und wenn der Primärknoten aus einem bestimmten Grund verloren geht, kann die sekundäre Instanz in die primäre Instanz aufgenommen und gestartet werden.

CP4I Bei der nativen HA handelt es sich um eine Hochverfügbarkeitslösung, die sich an Container richtet. Bei der nativen HA werden mit der Protokollreplikation drei Instanzen eines Warteschlangenmanagers, die auf unterschiedlichen Knoten ausgeführt werden, auf dem aktuellsten Stand gehalten. Eine Instanz ist zu jedem Zeitpunkt aktiv und verarbeitet Nachrichten. Der aktive Warteschlangenmanager sendet seine Protokollaktualisierungen an die anderen beiden Instanzen, damit diese aktualisiert bleiben. Wenn die aktive Instanz fehlschlägt, übernimmt eine der Replikatsinstanzen automatisch die aktive Rolle.

MQ Appliance Eine andere Option für eine Hochverfügbarkeits- oder Wiederherstellungslösung ist die Implementierung von IBM MQ-Appliances. Weitere Informationen finden Sie unter Hochverfügbarkeit und Disaster Recovery in der IBM MQ Appliance-Dokumentation.

Ein Messaging-System stellt sicher, dass Nachrichten, die in das System eingegeben werden, an ihr Ziel zugestellt werden. IBM MQ kann den Leitweg einer Nachricht verfolgen, wenn er mit dem Befehl **dspmqrte** von einem WS-Manager zu einem anderen versetzt wird. Wenn ein System ausfällt, können Nachrichten je nach Art des Fehlers und der Art und Weise, wie ein System konfiguriert wird, auf verschiedene Weise wiederhergestellt werden. IBM MQ verwaltet die Wiederherstellungsprotokolle der Aktivitä-

ten der Warteschlangenmanager, die den Empfang, die Übertragung und die Zustellung von Nachrichten verarbeiten.Sie verwendet diese Protokolle für drei Arten der Wiederherstellung:

1. *Wiederherstellung erneut starten*, wenn Sie IBM MQ in einer geplanten Weise stoppen.
2. *Fehlerbehebung*, wenn ein Fehler IBM MQ stoppt.
3. *Datenträgerwiederherstellung* , um beschädigte Objekte wiederherzustellen.

In allen Fällen stellt die Wiederherstellung den Warteschlangenmanager in den Status zurück, in dem er sich befand, als der Warteschlangenmanager gestoppt wurde, mit der Ausnahme, dass alle inflight-Transaktionen rückgängig gemacht werden. Aus den Warteschlangen werden alle Aktualisierungen entfernt, die zu dem Zeitpunkt, zu dem der Warteschlangenmanager gestoppt wurde, in den Status 'In-Flight' waren. Bei der Wiederherstellung werden alle persistenten Nachrichten zurückgespeichert. Nicht persistente Nachrichten können während des Prozesses verloren gehen.

**Vorsicht:** Sie können die Wiederherstellungsprotokolle nicht in ein anderes Betriebssystem versetzen.

## Automatische Clientverbindungswiederholung

Sie können Ihre Clientanwendungen automatisch erneut verbinden, ohne zusätzlichen Code schreiben zu müssen, indem Sie eine Reihe von Komponenten konfigurieren.

Die automatische Verbindungswiederholung von Clients erfolgt *integriert*. Die Verbindung wird automatisch an einem beliebigen Punkt im Clientanwendungsprogramm wiederhergestellt und es werden alle Kennungen für geöffnete Objekte wiederhergestellt.

Bei einer manuellen Verbindungswiederholung dagegen muss die Clientanwendung die Verbindung mithilfe von `MQCONN` oder `MQCONNX` erneut herstellen und die Objekte erneut öffnen. Die automatische Clientverbindungswiederholung ist für viele, nicht jedoch für alle Clientanwendungen geeignet.

In Tabelle 30 auf Seite 504 ist das früheste Release der IBM MQ-Clientunterstützung aufgeführt, das auf einer Client-Workstation installiert werden muss. Sie müssen für eine Anwendung ein Upgrade für Client-Workstations auf eine dieser Stufen durchführen, um die automatische Clientwiederverbindung zu verwenden. In Tabelle 31 auf Seite 504 sind weitere Voraussetzungen für die Aktivierung der automatischen Clientwiederverbindung aufgeführt.

Mit dem Programmzugriff auf Verbindungswiederverbindungsoptionen kann eine Clientanwendung Verbindungsoptionen festlegen. Mit Ausnahme von JMS- und XMS-Clients kann, wenn eine Clientanwendung Zugriff auf Verbindungswiederherstellungsoptionen hat, auch ein Ereignishandler erstellt werden, um Verbindungswiederherstellungseignisse zu verarbeiten.

Eine vorhandene Clientanwendung kann ohne erneute Kompilierung und Verlinkung von der Unterstützung für die erneute Verbindung profitieren können:

- Legen Sie für einen Nicht-JMS -Client die `mqclient.ini` -Umgebungsvariable `DefRecon` fest, um Optionen für die Verbindungswiederholung festzulegen. Verwenden Sie eine CCDT, um eine Verbindung zu einem WS-Manager herzustellen. Wenn der Client eine Verbindung zu einem Multi-Instanz-Warteschlangenmanager herstellen soll, stellen Sie die Netzadressen der aktiven und der Standby-WS-Manager-Instanzen in der CCDT bereit. Bei einem Warteschlangenmanager mit replizierten Daten (RDQM) oder einem Hochverfügbarkeits-Warteschlangenmanager auf einer IBM MQ-Appliance können Sie zum Vereinfachen der Konfiguration eine variable IP-Adresse angeben, die sowohl von aktiven als auch von Standby-Warteschlangenmanagern verwendet wird.
- Geben Sie für einen JMS-Client die Verbindungswiederholungsoptionen in der Konfiguration der Verbindungsfactory an. Wenn MDBs im EJB-Container eines Java EE -Servers ausgeführt werden, können sie die Verbindung zu IBM MQ wiederherstellen, indem sie den Mechanismus für die Verbindungswiederherstellung verwenden, der von den Aktivierungsspezifikationen des IBM MQ -Ressourcenadapters (bzw. Listener-Ports bei Ausführung in WebSphere Application Server) bereitgestellt wird. Wenn die Anwendung jedoch keine MDB ist (oder im Webcontainer ausgeführt wird), muss die Anwendung eine eigene Reconnect-Logik implementieren, da die automatische Clientwiederverbindung in diesem Szenario nicht unterstützt wird. Der IBM MQ-Ressourcenadapter stellt diese Verbindungsfähigkeit für die

Zustellung von Nachrichten an nachrichtengesteuerte Beans bereit, aber andere Java EE-Elemente wie Servlets müssen ihre eigene Verbindung implementieren.

**Anmerkung:** Die automatische Clientverbindungswiederholung wird von IBM MQ classes for Java nicht unterstützt.

*Tabelle 30. Unterstützte Clients*

| Clientschnittstelle | Client | Programmzugriff auf Verbindungswiederanschlussoptionen | Reconnection-Unterstützung |
|---|---|---|---|
| Messaging-APIs | C, C++, COBOL, Unmanaged Visual Basic, XMS (Unmanaged XMS unter Windows) | 7.0.1 | 7.0.1 |
| | JMS (JSE, Java EE-Client-Container und verwaltete Container) | 7.0.1.3 | 7.0.1.3 |
| | IBM MQ classes for Java | Nicht unterstützt | Nicht unterstützt |
| | Verwaltete XMS- und verwaltete .NET-Clients: C#, Visual Basic, | 7.1 | 7.1 |
| Andere APIs | Windows Communication Foundation (nicht verwaltet [1]) | Nicht unterstützt | 7.0.1 |
| | Windows Communication Foundation (verwaltet [1]) | Nicht unterstützt | Nicht unterstützt |
| | Achse 1 | Nicht unterstützt | Nicht unterstützt |
| | Achse 2 | Nicht unterstützt | 7.0.1.3 |
| | HTTP (Web 2.0) | Nicht unterstützt | 7.0.1.3 |

1. Definieren Sie den verwalteten oder nicht verwalteten Modus in der WCF-Bindungskonfiguration.

Die automatische Verbindungswiederverbindung hat die folgenden Konfigurationsanforderungen:

*Tabelle 31. Konfigurationsanforderungen für die automatische Verbindungswiederverbindung*

| Komponente | Voraussetzung | Auswirkung der Anforderung nicht erfüllen |
|---|---|---|
| IBM MQ MQI client-Installation | Siehe Tabelle 30 auf Seite 504 | `MQRC_OPTIONS_ERROR` |
| IBM MQ-Server-Installation | Stufe 7.0.1 | `MQRC_OPTIONS_ERROR` |
| Kanal | `SHARECNV > 0` | `MQRC_ENVIRONMENT_ERROR` |
| Anwendungsumgebung | Muss mit Threads versehen sein | `MQRC_ENVIRONMENT_ERROR` |

| *Tabelle 31. Konfigurationsanforderungen für die automatische Verbindungswiederverbindung (Forts.)* | | |
|---|---|---|
| **Komponente** | **Voraussetzung** | **Auswirkung der Anforderung nicht erfüllen** |
| MQI | Einer von:<br>• `MQCONNX` , wobei `MQCNO` Optionen auf `MQCNO_RECONNECT` oder `MQCNO_RECONNECT_Q_MGR`gesetzt ist.<br>• `Defrecon=YES\|QMGR`in`mqclient.ini`<br>• Legen Sie in JMS die Eigenschaft **`CLIENTRECONNECTOPTIONS`** der Verbindungsfactory fest. | `MQCC_FAILED`, wenn eine Verbindung unterbrochen wird oder der Warteschlangenmanager beendet wird oder fehlschlägt. |

Abbildung 68 auf Seite 505 zeigt die Hauptinteraktionen zwischen Komponenten, die an der Clientwiederverbindung beteiligt sind.

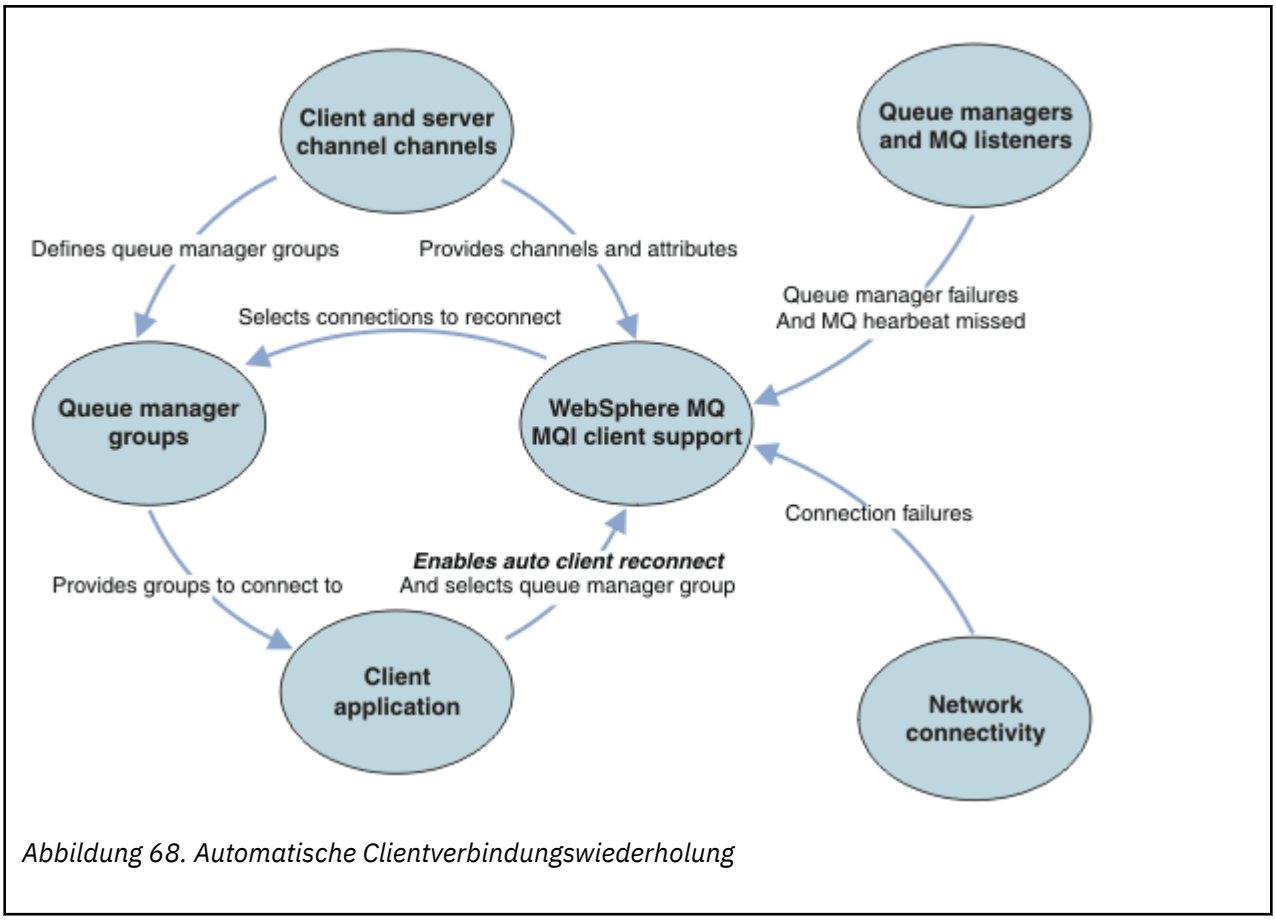



*Abbildung 68. Automatische Clientverbindungswiederholung*

## Clientanwendung

Die Clientanwendung ist ein IBM MQ MQI client. Ausführliche Informationen zur automatischen Clientverbindungswiederherstellung für einen JMS-Client finden Sie im Abschnitt Automatische JMS -Clientverbindungswiederherstellung verwenden.

- Standardmäßig werden die Clients nicht automatisch erneut verbunden. Aktivieren Sie die automatische Clientverbindungswiederholung, indem Sie die Option MQCONNX MQCNO MQCNO_RECONNECT oder MQCNO_RECONNECT_Q_MGRfestlegen.
- Viele Anwendungen sind so geschrieben, dass sie in der Lage sind, die automatische Verbindungswiederverbindung ohne zusätzliche Codierung zu nutzen. Aktivieren Sie die automatische Neuverbindung für vorhandene Programme, ohne Änderungen an der Codierung vorzunehmen, indem Sie das Attribut DefRecon in der Zeilengruppe für Kanäle der Konfigurationsdatei mqclient.ini festlegen.
- Verwenden Sie eine der folgenden drei Optionen:

  1. Ändern Sie das Programm so, dass die Logik nicht durch eine erneute Verbindung beeinträchtigt wird. Sie müssen beispielsweise MQI-Aufrufe innerhalb des Synchronisationspunkts absetzen und die zurückliegenden Transaktionen erneut übergeben. Asynchrone Konsumenten sollten prüfen, ob sie ausgesetzt wurden, wenn eine Transaktion zurückgesetzt wird.
  2. Fügen Sie einen Ereignishandler hinzu, um die Verbindungswiederherstellung zu ermitteln, und stellen Sie den Status der Clientanwendung wieder her, wenn die Verbindung wiederhergestellt wird.
  3. Aktivieren Sie die automatische Verbindungswiederholung nicht: Trennen Sie stattdessen den Client und geben Sie einen neuen MQCONN -oder MQCONNX MQI-Aufruf aus, um eine weitere Warteschlangenmanagerinstanz zu finden, die in derselben Warteschlangenmanagergruppe ausgeführt wird.

  Weitere Informationen zu diesen drei Optionen finden Sie im Abschnitt „Anwendungswiederherstellung” auf Seite 601.
- Die erneute Verbindung zu einem Warteschlangenmanager mit demselben Namen garantiert nicht, dass Sie mit derselben Instanz eines Warteschlangenmanagers erneut verbunden sind.

  Verwenden Sie die Option MQCNEIN MQCNO_RECONNECT_Q_MGR, um die Verbindung zu einer Instanz desselben Warteschlangenmanagers wiederherzustellen.
- Ein Client kann einen Ereignishandler registrieren, so dass er den Status der Verbindungswiederverbindung erhalten kann. Die an den Ereignishandler übergebene MQHCONN kann nicht verwendet werden. Es werden die folgenden Ursachencodes bereitgestellt:

  **MQRC_RECONNECTING**
  Die Verbindung ist fehlgeschlagen, und das System versucht, die Verbindung herzustellen. Sie empfangen mehrere MQRC_RECONNECTING -Ereignisse, wenn mehrere Versuche zur Verbindungswiederherstellung unternommen werden.

  **MQRC_RECONNECTED**
  Die Verbindungswiederverbindung wurde hergestellt und alle Handles wurden erfolgreich neu aufgebaut.

  **MQRC_RECONNECT_FAILED**
  Die Verbindungswiederverbindung war nicht erfolgreich.

  **MQRC_RECONNECT_QMID_MISMATCH**
  Eine wiederverbindbare Verbindung hat MQCNO_RECONNECT_Q_MGR angegeben und die Verbindung hat versucht, die Verbindung mit einem anderen Warteschlangenmanager wiederherzustellen.

  **MQRC_RECONNECT_Q_MGR_REQD**
  Eine Option, wie z. B. MQMO_MATCH_MSG_TOKEN in einem MQGET -Aufruf, wurde im Clientprogramm angegeben, das eine erneute Verbindung zu demselben Warteschlangenmanager erfordert.
- Ein wiederverbindungsfähiger Client kann erst nach Aufbau einer Verbindung eine automatische Verbindungswiederholung durchführen. Das heißt, der MQCONNX -Aufruf selbst wird nicht wiederholt, wenn er fehlschlägt. Wenn Sie beispielsweise den Rückkehrcode 2543 - MQRC_STANDBY_Q_MGR von MQCONNXempfangen, geben Sie den Aufruf nach einer kurzen Verzögerung erneut aus.

  **MQRC_RECONNECT_INCOMPATIBLE**
  Dieser Ursachencode wird zurückgegeben, wenn die Anwendung versucht, MQPMO_LOGICAL_ORDER (mit MQPUT1 und MQPUT1) oder MQGMO_LOGICAL_ORDER (mit MQGET ) zu verwenden wenn Optionen für die Verbindungswiederherstellung festgelegt sind. Der Grund für die Rückgabe des Ursachencodes ist es, sicherzustellen, dass Anwendungen in solchen Fällen nie wieder eine Verbindung herstellen.

**MQRC_CALL_INTERRUPTED**
Dieser Ursachencode wird zurückgegeben, wenn die Verbindungsunterbrechungen während der Ausführung des Commit-Aufrufs unterbrochen werden und der Client die Verbindung wiederherstellt. Ein MQPUT-Aufruf einer persistenten Nachricht außerhalb des Synchronisationspunkts führt ebenfalls zu demselben Ursachencode, der an die Anwendung zurückgegeben wird.

## Hochverfügbarkeits-Warteschlangenmanager

Hochverfügbarkeits-Warteschlangenmanager verfügen über eine aktive Instanz und eine oder mehrere Standby-Instanzen eines Warteschlangenmanagers. Der aktive Warteschlangenmanager wird mit den Standby-Warteschlangenmanagern synchronisiert, so dass automatisch auf eine Standby-Instanz umgeschaltet werden kann, wenn die aktive Instanz ausfällt. Es gibt eine Reihe unterschiedlicher Lösungen für die Bereitstellung von Hochverfügbarkeits-Warteschlangenmanagern (siehe „Hochverfügbarkeitskonfigurationen” auf Seite 514).

Sie können die Neustart von IBM MQ MQI client-Anwendungen vereinfachen, nachdem ein Hochverfügbarkeits-Warteschlangenmanager seine Standby-Instanz aktiviert hat, indem Sie die automatische Clientverbindungswiederherstellung verwenden.

Die Standby-Instanz eines Hochverfügbarkeits-Warteschlangenmanagers läuft in der Regel mit einer anderen Netzadresse als die aktive Instanz. Geben Sie die Netzadressen der Instanzen in der Definitionstabelle für die Clientverbindung an (CCDT). Geben Sie entweder eine Liste mit Netzadressen für den Parameter **CONNAME** an oder definieren Sie mehrere Zeilen für den WS-Manager in der CCDT. Warteschlangenmanager mit replizierten Daten (RDQM) und IBM MQ-Appliance-Hochverfügbarkeits-Warteschlangenmanager unterstützen variable IP-Adressen, bei denen Sie eine einzige Adresse für die Verwendung mit aktiven oder Standby-Warteschlangenmanagern angeben.

## WS-Manager-Gruppen

Normalerweise stellen IBM MQ MQI clients die Verbindung zu einem beliebigen WS-Manager in einer Warteschlangenmanagergruppe wieder her. Manchmal möchte man aber, dass ein IBM MQ MQI client die Verbindung nur zu demselben Warteschlangenmanager wiederherstellt. Es kann eine Affinität zu einem Warteschlangenmanager haben.

Sie können auswählen, ob die Clientanwendung immer eine Verbindung zu einem WS-Manager desselben Namens, desselben Warteschlangenmanagers oder einer Gruppe von Warteschlangenmanagern herstellt, die mit demselben `QMNAME` -Wert in der Clientverbindungstabelle definiert sind.

- Das Attribut `QMNAME` des Warteschlangenmanagernamens in der Clientkanaldefinition ist der Name einer Warteschlangenmanagergruppe.
- Wenn Sie in Ihrer Clientanwendung den Wert des Parameters `MQCONN` oder `MQCONNX` `QmgrName` auf einen Warteschlangenmanagernamen setzen, stellt der Client nur eine Verbindung zu Warteschlangenmanagern mit diesem Namen her. Wenn Sie dem WS-Manager-Namen einen Stern (*) voranstellen, stellt der Client eine Verbindung zu jedem WS-Manager in der Warteschlangenmanagergruppe mit demselben `QMNAME` -Wert her. Eine vollständige Erläuterung finden Sie im Abschnitt WS-Manager-Gruppen in der CCDT .

Sie können verhindern, dass ein Client die Verbindung zu einem anderen WS-Manager erneut herstellen kann. Legen Sie die Option `MQCNEIN` fest: `MQCNO_RECONNECT_Q_MGR`. Der IBM MQ MQI client schlägt fehl, wenn er die Verbindung zu einem anderen WS-Manager wiederherstellt. Wenn Sie die Option `MQCNEIN` festlegen, `MQCNO_RECONNECT_Q_MGR`, schließen Sie keine anderen Warteschlangenmanager in dieselbe Warteschlangenmanagergruppe ein. Der Client gibt einen Fehler zurück, wenn der Warteschlangenmanager, zu dem er die Verbindung herstellt, nicht mit dem Warteschlangenmanager identisch ist, mit dem er verbunden ist.

## Gruppen mit gemeinsamer Warteschlange

z/OS Die automatische Clientverbindungswiederholung zu z/OS-Gruppen mit gemeinsamer Warteschlange verwendet die gleichen Verfahren zur Verbindungswiederholung wie jede andere Umgebung. Der Client stellt die Verbindung zu derselben Auswahl von Warteschlangenmanagern wieder her, wie für

die ursprüngliche Verbindung konfiguriert ist. Beispielsweise sollte der Administrator beim Verwenden der Definitionstabelle für den Clientkanal sicherstellen, dass alle Einträge in der Tabelle in die gleiche z/OS-Gruppe mit gemeinsamer Warteschlange aufgelöst werden.

## Client-und Serverkanaldefinitionen

Client-und Serverkanaldefinitionen definieren die Gruppen von Warteschlangenmanagern, mit der eine Clientanwendung erneut eine Verbindung herstellen kann. Die Definitionen steuern die Auswahl und die Ablaufsteuerung von Verbindungswiederanschlüssen sowie andere Faktoren, wie z. B. die Sicherheit; siehe zugehörige Themen. Die relevantesten Kanalattribute, die für die Verbindungswiederverbindung zu berücksichtigen sind, werden in zwei Gruppen aufgelistet:

**Clientverbindungsattribute**

**Verbindungsaffinität (AFFINITY) AFFINITY**
Verbindungsaffinität.

**Clientkanalgewichtung (CLNTWGHT) CLNTWGHT**
Gewichtung des Clientkanals.

**Verbindungsname (CONNAME) CONNAME**
Verbindungsinformationen.

**Heartbeat-Intervall (HBINT) HBINT**
Intervall der Überwachungssignale. Legen Sie das Intervall der Überwachungssignale auf dem Serververbindungskanal fest.

**Keepalive-Intervall (KAINT) KAINT**
Keepalive-Intervall. Legen Sie das Keepalive-Intervall auf dem Serververbindungskanal fest.

z/OS Beachten Sie, dass KAINT nur für z/OS gültig ist.

**Warteschlangenmanagername (QMNAME) QMNAME**
Warteschlangenmanagername.

**Serververbindungsattribute**

**Heartbeat-Intervall (HBINT) HBINT**
Intervall der Überwachungssignale. Legen Sie das Intervall der Überwachungssignale auf dem Clientverbindungskanal fest.

**Keepalive-Intervall (KAINT) KAINT**
Keepalive-Intervall. Legen Sie das Keepalive-Intervall auf dem Clientverbindungskanal fest.

z/OS Beachten Sie, dass KAINT nur für z/OS gültig ist.

KAINT ist ein Überwachungssignal für die Netzebene, und HBINT ist ein IBM MQ-Überwachungssignal zwischen dem Client und dem Warteschlangenmanager. Die Einstellung dieser Überwachungssignale in kürzerer Zeit dient zwei Zwecken:

1. Durch die Simulation der Aktivität auf der Verbindung wird die Netzschichtsoftware, die für das Schließen inaktiver Verbindungen verantwortlich ist, weniger wahrscheinlich die Verbindung beenden.
2. Wenn die Verbindung beendet wird, wird die Verzögerung vor dem Erkennen der unterbrochenen Verbindung verkürzt.

Das standardmäßige TCP/IP-Keepalive-Intervall beträgt zwei Stunden. Ziehen Sie in Betracht, die Attribute KAINT und HBINT auf eine kürzere Zeit zu setzen. Gehen Sie nicht davon aus, dass das normale Verhalten eines Netzes den Anforderungen der automatischen Verbindungswiederverbindung entspricht. Einige Firewalls können z. B. eine inaktive TCP/IP-Verbindung nach weniger als 10 Minuten beenden.

## Netzkonnektivität

Nur Netzausfälle, die vom Netz an den IBM MQ MQI client übergeben werden, werden von der automatischen Verbindungswiederholungsfunktionalität des Clients verarbeitet.

- Automatisch vom Transport durchgeführte Verbindungsabgleich sind für IBM MQ nicht sichtbar.

- Durch die Einstellung von HBINT können Netzausfälle behandelt werden, die für IBM MQ nicht sichtbar sind.

## Warteschlangenmanager und IBM MQ-Listener

Die Clientwiederverbindung wird durch Serverfehler, Ausfall des Warteschlangenmanagers, Netzkonnektivitätsfehler und durch einen Administrator ausgelöst, der zu einer anderen WS-Manager-Instanz umschaltet.

- Wenn Sie einen Warteschlangenmanager mit mehreren Instanzen verwenden, tritt eine zusätzliche Ursache für die Clientwiederverbindung auf, wenn Sie die Steuerung von der aktiven WS-Manager-Instanz zu einer Standby-Instanz wechseln.
- Beim Beenden eines Warteschlangenmanagers mit dem Standardbefehl **endmqm** wird die automatische Clientverbindungswiederholung nicht ausgelöst. Fügen Sie die Option `-r` im Befehl **endmqm** hinzu, um die automatische Clientverbindungswiederholung anzufordern, bzw. die Option `-s` , um nach dem Herunterfahren eine Übertragung an eine Standby-Warteschlangenmanagerinstanz anzufordern.

## Unterstützung für automatische Verbindungswiederholung im IBM MQ MQI client

Wenn Sie die Unterstützung für die automatische Clientverbindungswiederholung im IBM MQ MQI client verwenden, stellt die Clientanwendung die Verbindung automatisch wieder her und setzt die Verarbeitung fort, ohne dass Sie einen MQI-Aufruf `MQCONN` oder `MQCONNX` absetzen müssen, um die Verbindung zum WS-Manager wiederherzustellen.

- Die automatische Neuverbindung des Clients wird durch eines der folgenden Vorkommen ausgelöst:
  - Warteschlangenmanagerfehler
  - Beenden eines Warteschlangenmanagers und Angebe der Option `-r` zur Verbindungswiederholung im Befehl **endmqm**
- Die `MQCONNX` MQCNO-Optionen steuern, ob Sie die automatische Clientverbindungswiederholung aktiviert haben. Die Optionen sind im Abschnitt Optionen für die Verbindungsabgleich beschrieben.
- Automatische Clientverbindungsfehler MQI-Aufrufe im Namen Ihrer Anwendung, um die Verbindungskennung und die Kennungen für andere offene Objekte wiederherzustellen, so dass Ihr Programm die normale Verarbeitung wieder aufnehmen kann, nachdem es alle MQI-Fehler verarbeitet hat, die sich aus der unterbrochenen Verbindung ergeben haben. Siehe „Wiederherstellung eines automatisch verbundenen Clients” auf Seite 603.
- Wenn Sie ein Kanalexitprogramm für die Verbindung geschrieben haben, empfängt der Exit diese zusätzlichen MQI-Aufrufe.
- Sie können einen Umverbindungsereignishandler registrieren, der ausgelöst wird, wenn die Verbindungswiederverbindung beginnt und wann er beendet wird.

Obwohl die geplante Verbindungswiederverbindung nicht länger als eine Minute ist, kann die Verbindungswiederverbindung länger dauern, da ein Warteschlangenmanager möglicherweise zahlreiche Ressourcen zu verwalten hat. Während dieser Zeit kann es vorkommen, dass eine Clientanwendung Sperren hält, die nicht zu IBM MQ-Ressourcen gehören. Es gibt einen Zeitlimitwert, der konfiguriert werden kann, um die Zeit zu begrenzen, die ein Client auf die erneute Verbindung wartet. Der Wert (in Sekunden) wird in der Datei `mqclient.ini` festgelegt.

```
Channels:
MQReconnectTimeout = 1800
```

Nach Ablauf des Zeitlimits werden keine Umverbindungsversuche durchgeführt. Wenn das System feststellt, dass das Zeitlimit abgelaufen ist, gibt es einen `MQRC_RECONNECT_FAILED` -Fehler zurück.

**Zugehörige Konzepte**

Wiederverbindungsfähige Clients

**Zugehörige Tasks**

Stoppen eines Warteschlangenmanagers

# z/OS Überwachung der Konsolnachricht

Unter IBM MQ for z/OS gibt es eine Reihe von Informationsnachrichten, die vom Warteschlangenmanager oder vom Kanalinitiator ausgegeben werden und die als besonders wichtig betrachtet werden sollten. Diese Nachrichten weisen nicht auf ein Problem hin, können aber in der Verfolgung nützlich sein, da sie auf ein potenzielles Problem hinweisen, das möglicherweise Adressierung erfordern könnte.

Das Vorhandensein dieser Konsolnachrichten kann auch darauf hinweisen, dass eine Benutzeranwendung eine große Anzahl von Nachrichten an die Seitengruppe stellt, was möglicherweise ein Symptom eines größeren Problems sein könnte:

- Ein Problem mit der Benutzeranwendung, die PUTs-Nachrichten, wie z. B. eine unkontrollierte Schleife, enthält.
- Eine Benutzeranwendung, die die Nachrichten aus der Warteschlange GETs aufgibt, funktioniert nicht mehr.

## Konsolnachrichten, die überwacht werden sollen

In der folgenden Liste werden Nachrichten aufgeführt, die möglicherweise größere Probleme anzeigen können. Stellen Sie fest, ob es erforderlich ist, diese Nachrichten mit der Systemautomatisierung zu verfolgen und eine entsprechende Dokumentation bereitzustellen, damit potenzielle Probleme effektiv verfolgt werden können.

**CSQI004I: *csect-name* CONSIDER INDEXING *queue-name* BY *index-type* FOR *connection-type* CONNECTION *connection-name*, *num-msgs* MESSAGES SKIPPED**

- Der Warteschlangenmanager hat eine Anwendung erkannt, die Nachrichten nach Nachrichten-ID oder Korrelations-ID aus einer Warteschlange empfängt, für die kein Index definiert ist.
- Ziehen Sie die Erstellung eines Index für die angegebene Warteschlange in Betracht, indem Sie das lokale Warteschlangenobjekt *queue-name*, das Attribut INDXTYPE ändern, um den Wert *index-type* zu haben.

**CSQI031I: *csect-name* Der neue EXTENT OF SEITE SET *psid* HAT FORMATTED SUCCESSFULLY**

- Überprüfen Sie die Sperrtiefe der Warteschlangen, die dieser Seitengruppe zugeordnet sind.
- Untersuchen Sie die Ursache für den Fehler beim Verarbeiten der Nachrichten.

.

**CSQI041I: *csect-name* JOB *jobname* USER *userid* HAD ERROR ZUGRIFFPAGE SET *psid***

- Stellen Sie fest, ob die Seitengruppe dem Warteschlangenmanager zugeordnet ist.
- Setzen Sie einen **DISPLAY USAGE** -Befehl ab, um den Status der Seitengruppe zu bestimmen.
- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Fehlernachrichten.

.

**CSQI045I: *csect-name* Protokoll-RBA hat *rba*erreicht. Plan a log reset**

- Planen Sie den Stopp des Warteschlangenmanagers zu einem geeigneten Zeitpunkt und setzen Sie die Protokolle zurück.
- Wenn Ihr Warteschlangenmanager Protokoll-RBAs mit 6 Byte verwendet, sollten Sie den Warteschlangenmanager für die Verwendung von Protokoll-RBAs mit 8 Byte konvertieren.

**CSQI046E: *csect-name* Protokoll RBA hat *rba*erreicht. Perform a log reset**

- Planen Sie den Stopp des Warteschlangenmanagers zu einem geeigneten Zeitpunkt und setzen Sie die Protokolle zurück.
- Wenn Ihr Warteschlangenmanager Protokoll-RBAs mit 6 Byte verwendet, sollten Sie den Warteschlangenmanager für die Verwendung von Protokoll-RBAs mit 8 Byte konvertieren.

**CSQI047E: *csect-name* Protokoll RBA hat *rba*erreicht. Stop queue manager and reset logs**

- Stoppen Sie sofort den Warteschlangenmanager und setzen Sie die Protokolle zurück.

- Wenn Ihr Warteschlangenmanager Protokoll-RBAs mit 6 Byte verwendet, sollten Sie den Warteschlangenmanager für die Verwendung von Protokoll-RBAs mit 8 Byte konvertieren.

**CSQJ004I: ACTIVE LOG COPY *n* INACTIVE, LOG IN SINGLE MODE, ENDRBA= *TTT***

- Der Warteschlangenmanager hat den Protokollmodus 'Einfach' aktiviert. Dies ist häufig ein Hinweis auf ein Problem bei der Protokollauslagerung.
- Setzen Sie den Befehl **DISPLAY LOG** ab, um die Einstellungen für die Duplizierung aktiver und archivierender Protokolle zu bestimmen. Diese Anzeige zeigt außerdem, wie viele aktive Protokolle eine Auslagerungsverarbeitung erfordern.
- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Fehlernachrichten.

**CSQJ031D: *csect-name*: Der RBA-Protokollbereich muss zurückgesetzt werden. REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN**

- Stoppen Sie den Warteschlangenmanager und setzen Sie die Protokolle so schnell wie möglich zurück.
- Wenn Ihr Warteschlangenmanager Protokoll-RBAs mit 6 Byte verwendet, sollten Sie den Warteschlangenmanager für die Verwendung von Protokoll-RBAs mit 8 Byte konvertieren.

**CSQJ032E: *csect-name alert-lvl* -nähert sich dem Ende des Protokoll-RBA-Bereichs von *max-rba*. CURRENT LOG RBA IS *current-rba*.**

- Planen Sie den Stopp des Warteschlangenmanagers und setzen Sie die Protokolle so bald wie möglich zurück.
- Wenn Ihr Warteschlangenmanager Protokoll-RBAs mit 6 Byte verwendet, sollten Sie den Warteschlangenmanager für die Verwendung von Protokoll-RBAs mit 8 Byte konvertieren.

**CSQJ110E: LAST COPY *n* ACTIVE LOG DATA SET IS *nnn* PERCENT FULL**

- Führen Sie die erforderlichen Schritte aus, um andere ausstehende Auslagerungstasks auszuführen, indem Sie eine Anzeigeanforderung ausführen, um die ausstehenden Anforderungen im Zusammenhang mit dem Protokollauslagerungsprozess zu ermitteln. Leiten Sie die erforderlichen Schritte ein, um allen Anforderungen zu entsprechen, und lassen Sie zu, dass die Auslagerung fortgesetzt wird.
- Überlegen Sie, ob genügend aktive Protokolldateien vorhanden sind. Falls erforderlich, können Sie mit dem Befehl DEFINE LOG zusätzliche Protokolldatengruppen dynamisch hinzufügen.

**CSQJ111A: KEIN SPEICHERPLATZ IN AKTIVEN PROTOKOLLDATEIEN**

- Führen Sie eine Anzeigeanforderung aus, um sicherzustellen, dass keine ausstehenden Anforderungen vorhanden sind, die sich auf den Protokollauslagerungsprozess beziehen. Leiten Sie die erforderlichen Schritte ein, um allen Anforderungen zu entsprechen, und lassen Sie zu, dass die Auslagerung fortgesetzt wird.
- Überlegen Sie, ob genügend aktive Protokolldateien vorhanden sind. Falls erforderlich, können Sie mit dem Befehl DEFINE LOG zusätzliche Protokolldatengruppen dynamisch hinzufügen.
- Wenn die Verzögerung durch den Mangel an einer Ressource verursacht wurde, die für die Auslagerung erforderlich ist, muss die erforderliche Ressource verfügbar gemacht werden, damit die Auslagerung abgeschlossen werden kann und die Protokollierung somit fortgesetzt werden kann. Informationen zur Wiederherstellung aus dieser Bedingung finden Sie im Abschnitt Probleme im Archivprotokoll.

**CSQJ114I: FEHLER ON ARCHIVE DATA SET, OFFLOAD CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING GENERATED**

- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Fehlernachrichten.
- Erstellen Sie eine zweite Kopie des Archivprotokolls und aktualisieren Sie Ihre BSDS manuell.

**CSQJ115E: OFFLOAD FAILED, COULD NOT ALLOCATE AN ARCHIVE DATA SET**

Überprüfen Sie die Fehlerstatusinformationen der Nachricht CSQJ103E oder CSQJ073E. Korrigieren Sie die Bedingung, die den Zuordnungsfehler für die Datei verursacht hat, so dass die Auslagerung bei der Wiederholung ausgeführt werden kann.

**CSQJ136I: UNABLE TO ALLOCATE TAPE UNIT FOR CONNECTION-ID= *xxxx* CORRELATION-ID= *yyyyyy*, *m* ALLOCATED *n* ALLOWED**

- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Fehlernachrichten.

.

**CSQJ151I: *csect-name* ERROR READING RBA *rrr*, CONNECTION-ID = *xxxx* CORRELATION-ID= *yyyyyy* REASON CODE= *ccc***

- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Nachrichten.
- Setzen Sie den Befehl **DISPLAY CONN** ab, um festzustellen, welche Verbindung die zugehörige Aktivität nicht festgeschrieben hat.
- Stellen Sie sicher, dass die Anwendung ihre Aktualisierungen festschreiben kann.

**CSQJ160I: LONG-RUNNING UOW FOUND, URID= *urid* CONNECTION NAME= *name***

- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Nachrichten.
- Setzen Sie den Befehl **DISPLAY CONN** ab, um festzustellen, welche Verbindung die zugehörige Aktivität nicht festgeschrieben hat.
- Stellen Sie sicher, dass die Anwendung ihre Aktualisierungen festschreiben kann.

**CSQJ161I: UOW UNRESOLVED AFTER *n* OFFLOADS, URID= *urid* CONNECTION NAME= *name***

- Stellen Sie fest, ob die Seitengruppe dem Warteschlangenmanager zugeordnet ist.
- Setzen Sie einen **DISPLAY USAGE** -Befehl ab, um den Status der Seitengruppe zu bestimmen.
- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Nachrichten.

**CSQP011E: CONNECT ERROR STATUS -*Rückkehrcode* FOR PAGE SET *psid***

- Überprüfen Sie die Sperrtiefe der Warteschlangen, die dieser Seitengruppe zugeordnet sind.
- Untersuchen Sie die Ursache für den Fehler beim Verarbeiten von Nachrichten.

**CSQP013I: *csect-name* NEW EXTENT CREATED FOR PAGE SET psid. NEW EXTENT WILL NOW BE FORMATTED**

- Überprüfen Sie die Sperrtiefe der Warteschlangen, die dieser Seitengruppe zugeordnet sind.
- Untersuchen Sie die Ursache für das Fehlschlagen von Nachrichten.
- Stellen Sie fest, ob Warteschlangen in eine andere Seitengruppe umgestellt werden müssen.
- Wenn der Datenträger voll ist, stellen Sie fest, ob Sie die Seitengruppe als Datei mit mehreren Datenträgern definieren müssen. Wenn die Seitengruppe bereits mehrere Datenträger enthält, sollten Sie in Betracht ziehen, der Speichergruppe, die verwendet wird, weitere Datenträger hinzuzufügen. Wenn mehr Speicherplatz verfügbar ist, wiederholen Sie die Erweiterung, indem Sie die Seitengruppe **EXPAND** auf **SYSTEM** setzen. Wenn eine Wiederholung erforderlich ist, schalten Sie **EXPAND** in **SYSTEM** und anschließend wieder in Ihre normale Einstellung ein.

**CSQP014E: *csect-name* EXPANSION FEHLGESCHLAGEN FÜR SEITE SETZEN SID. FUTURE REQUESTS TO EXTEND IT WILL BE REJECTED**

- Überprüfen Sie die Sperrtiefe der Warteschlangen, die dieser Seitengruppe zugeordnet sind.
- Untersuchen Sie die Ursache für das Fehlschlagen von Nachrichten.
- Stellen Sie fest, ob Warteschlangen in eine andere Seitengruppe umgestellt werden müssen.

**CSQP016E: *csect-name* SEITE SET *psid* HAT REACHED THE MAXIMUM NUMBER OF EXTENTS. IT CANNOT BE EXTENDED AGAIN**

- Überprüfen Sie die Sperrtiefe der Warteschlangen, die dieser Seitengruppe zugeordnet sind.
- Untersuchen Sie die Ursache für das Fehlschlagen von Nachrichten.

**CSQP017I: *csect-name* EXPANSION STARTED FOR PAGE SET *psid***

Geben Sie den Befehl DISPLAY THREAD aus, um den Status der Arbeitseinheiten in IBM MQ zu ermitteln.

**CSQP047E: Nicht verfügbare Seitengruppen können Probleme verursachen-Maßnahmen ergreifen, um diese Situation zu korrigieren**

- Führen Sie die Systemprogrammiereraktion aus.

**CSQQ008I: *nn* Arbeitseinheiten mit Wiederherstellung sind im Warteschlangenmanager *qqq* noch unbestätigt.**

- Überprüfen Sie den Status Ihrer Warteschlange für nicht zustellbare Mail. Stellen Sie sicher, dass die Warteschlange für nicht zustellbare Nachrichten nicht inaktiviert ist
- Stellen Sie sicher, dass die Warteschlange für nicht zustellbare Nachrichten nicht den Grenzwert für MAXMSG aufweist.

**CSQQ113I: *psb-name region-id* Diese Nachricht kann nicht verarbeitet werden**

- Überprüfen Sie die CSQOUTX-Datei, um die Ursache für den CSQINPX-Fehler zu ermitteln.
- Einige Befehle werden möglicherweise nicht verarbeitet.

**CSQX035I: *csect-name* Verbindung zu Warteschlangenmanager *qmgr-name* wird gestoppt oder unterbrochen, MQCC= *mqcc* MQRC= *mqrc* (*mqrc-text***

- Überprüfen Sie den MQRC, um die Ursache des Fehlers zu ermitteln.
- Diese Codes sind in IBM MQ for z/OS-Nachrichten, -Beendigungscodes und -Ursachencodes dokumentiert.

**CSQX032I: *csect-name* -Steuerroutine für Initialisierungsbefehl beendet**

- Überprüfen Sie den MQRC, um die Ursache des Fehlers zu ermitteln.
- Diese Codes sind in IBM MQ for z/OS-Nachrichten, -Beendigungscodes und -Ursachencodes dokumentiert.

**CSQX048I: *csect-name* Die Nachricht für *name* kann nicht konvertiert werden. MQCC= *mqcc* MQRC= *mqrc* (*mqrc-text*)**

- Überprüfen Sie das Jobprotokoll, um die Ursache für den TCP/IP-Fehler zu ermitteln.
- Überprüfen Sie den TCP/IP-Adressraum auf Fehler.

**CSQX234I: *csect-name* Listener gestoppt, TRPTYPE= *trptype* INDISP= *disposition***

- Wenn der Listener nach einem **STOP**-Befehl nicht gestoppt wird, überprüfen Sie den TCP/IP-Adressraum auf Fehler.
- Führen Sie die Systemprogrammiereraktion aus.

**CSQX407I: *csect-name* Definitionen der Clusterwarteschlange *q-name* inkonsistent**

- Mehrere Clusterwarteschlangen innerhalb des Clusters haben inkonsistente Werte. Untersuchen und beheben Sie die Unterschiede.

**CSQX411I: *csect-name* Repository-Manager gestoppt**

- Wenn der Repository-Manager aufgrund eines Fehlers gestoppt wurde, überprüfen Sie das Jobprotokoll auf Nachrichten.

**CSQX417I: *csect-name* Clustersender verbleiben für entfernten Warteschlangenmanager *qmgr-name***

- Führen Sie die Systemprogrammiereraktion aus.

**CSQX418I: *csect-name* Nur ein Repository für den Cluster *cluster_name***

- Zur Erhöhung der Hochverfügbarkeit sollten Cluster mit zwei vollständigen Repositorys konfiguriert werden.

**CSQX419I: *csect-name* No cluster-receiver for cluster *clustername***

- Führen Sie die Systemprogrammiereraktion aus.

**CSQX420I: *csect-name* Keine Repositorys für Cluster *cluster_name***

- Führen Sie die Systemprogrammiereraktion aus.

**CSQX448E: *csect-name* Der Repository-Manager wird aufgrund von Fehlern gestoppt. Restart in *n* seconds**

- Führen Sie die Systemprogrammiereraktion aus.

  Diese Nachricht wird alle 600 Sekunden (10 Minuten) ausgegeben, bis die Warteschlange SYSTEM.CLUSTER.COMMAND.QUEUE aktiviert ist. Verwenden Sie dazu den folgenden Befehl:

  ```
  ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
  ```

  Vor dem Aktivieren der Warteschlange muss möglicherweise ein manueller Eingriff erforderlich sein, um das Problem zu beheben, das die Beendigung des Repository-Managers verursacht hat, bevor die erste Nachricht CSQX448E ausgegeben wird.

**CSQX548E: *csect-name* Nachrichten an lokale Warteschlange für nicht zustellbare Nachrichten gesendet, Kanal *channel-name* reason=*mqrc* (*mqrc-text*)**

- Führen Sie die Systemprogrammiereraktion aus.

**CSQX788I: *csect-name* Die DNS-Suche nach der Adresse *adresse* mit der Funktion '*func*' dauerte *n* Sekunden**

- Führen Sie die Systemprogrammiereraktion aus.

**CSQY225E: *csect-name* Queue manager is kritisch short of local storage above the bar-take action**

- Der Warteschlangenmanager verfügt nicht mehr über ausreichend virtuellen Speicher oberhalb des Grenzwerts; der Zustand ist kritisch. Es müssen Maßnahmen ergriffen werden, um das Problem zu beheben und eine mögliche abnormale Beendigung des Warteschlangenmanagers zu vermeiden.

**CSQ5038I: *csect-name* Service-Task Service-Task reagiert seit hh.mm.ss.nnnnnnnicht mehr. Suchen Sie nach Problemen mit Db2**

- Führen Sie die Systemprogrammiereraktion aus.

## Hochverfügbarkeitskonfigurationen

Wenn Sie Ihre IBM MQ-Warteschlangenmanager in einer Hochverfügbarkeitskonfiguration betreiben möchten, können Sie Ihre Warteschlangenmanager so konfigurieren, dass sie entweder mit einem High Availability Manager arbeiten, z. B. PowerHA for AIX (vormals HACMP ). oder den Microsoft -Clusterservice (MSCS) oder mit IBM MQ-Multi-Instanz-Warteschlangenmanagern. Auf Linux-Systemen können Sie auch replizierte Datenwarteschlangenmanager (RDQMs) implementieren, die eine Quorum-basierte Gruppe verwenden, um eine hohe Verfügbarkeit bereitzustellen. Eine weitere Option, die native Hochverfügbarkeit, richtet sich an Containerbereitstellungen.

Eine andere Option für eine Hochverfügbarkeits- oder Wiederherstellungslösung ist die Implementierung von IBM MQ-Appliances. Weitere Informationen finden Sie unter Hochverfügbarkeit und Disaster Recovery in der IBM MQ Appliance-Dokumentation.

Sie müssen über die folgenden Konfigurationsdefinitionen informiert sein:

**Cluster aus Warteschlangenmanagern**
Gruppen von mindestens zwei Warteschlangenmanagern auf einem oder mehreren Computern, die eine automatische Verbindung bereitstellen, und die gemeinsame Nutzung von Warteschlangen für Lastausgleich und Redundanz ermöglichen. Ab IBM WebSphere MQ 7.1 werden bei der Behebung von Clusterfehlern diejenigen Operationen, die zu Problemen geführt haben, erneut ausgeführt, bis die Probleme behoben sind.

**HA-Cluster**
HA-Cluster sind Gruppen von zwei oder mehr Computern und Ressourcen wie Platten und Netze, die miteinander verbunden und so konfiguriert sind, dass ein hoher Verfügbarkeitsmanager, wie z. B. HACMP (AIX and Linux) oder MSCS (Windows), einen *Failover* ausführt, wenn ein Fehler ausfällt.

Die Übernahme überträgt die Statusdaten von Anwendungen vom fehlerhaften Computer auf einen anderen Computer im Cluster und leitet dort ihre Operation erneut ein. Dies bietet eine hohe Verfügbarkeit von Services, die innerhalb des HA-Clusters ausgeführt werden. Die Beziehung zwischen IBM MQ-Clustern und HA-Clustern wird in „Beziehung zwischen HA-Clustern und WS-Manager-Clustern" auf Seite 516 beschrieben.

**Warteschlangenmanager mit mehreren Instanzen**
Instanzen desselben Warteschlangenmanagers, die auf zwei oder mehr Computern konfiguriert sind. Wenn Sie mehrere Instanzen starten, wird eine Instanz zur aktiven Instanz, und die anderen Instanzen werden zu Standardinstanzen. Wenn die aktive Instanz ausfällt, übernimmt eine Standby-Instanz, die auf einem anderen Computer ausgeführt wird, automatisch die Ausführung. Sie können Warteschlangenmanager mit mehreren Instanzen verwenden, um Ihre eigenen hoch verfügbaren Messaging-Systeme auf der Basis von IBM MQ zu konfigurieren, ohne dass eine Clustertechnologie wie HACMP oder MSCS erforderlich ist. HA-Cluster und Warteschlangenmanager mit mehreren Instanzen sind alternative Methoden zum Hochverfügbarkeitsmanagement von Warteschlangenmanagern. Kombinieren Sie sie nicht, indem Sie einen WS-Manager mit mehreren Instanzen in einem HA-Cluster einreihen.

**Hochverfügbarkeit replizierter Datenwarteschlangenmanager (HA RDQMs)**
Instanzen desselben Warteschlangenmanagers, die auf jedem Knoten in einer Gruppe von drei Linux-Servern konfiguriert sind. Eine der drei Instanzen ist die aktive Instanz. Die Daten aus dem aktiven Warteschlangenmanager werden synchron auf die beiden anderen Instanzen repliziert, sodass eine dieser Instanzen im Falle eines Fehlers die Überlassung übernehmen kann. Die Gruppierung der Server wird von Pacemaker gesteuert, und die Replikation durch DRBD.

**Disaster Recovery replizierte Datenwarteschlangenmanager (DR RDQMs)**
Ein Warteschlangenmanager wird auf einem Primärknoten an einer Site ausgeführt, wobei eine sekundäre Instanz dieses Warteschlangenmanagers auf einem Wiederherstellungsknoten an einer anderen Niederlassung angeordnet ist. Daten werden zwischen der primären Instanz und der sekundären Instanz repliziert, und wenn der Primärknoten aus einem bestimmten Grund verloren geht, kann die sekundäre Instanz in die primäre Instanz aufgenommen und gestartet werden. Beide Knoten müssen Linux-Server sein. Die Replikation wird von DRBD gesteuert.

**Replizierte Datenwarteschlangenmanager für Disaster-Recovery/Hochverfügbarkeit (DR/HA RDQMs)**
Sie können einen RDQM (Replicated Data Queue Manager) konfigurieren, der in einer Hochverfügbarkeitsgruppe an einem Standort ausgeführt wird, aber in eine andere Hochverfügbarkeitsgruppe (HA-Gruppe) an einem anderen Standort übernommen werden kann, falls die erste Gruppe nach einem Störfall nicht mehr verfügbar ist. Dieser Warteschlangenmanager wird als DR/HA-RDQM bezeichnet.

CP4I **Native HA**
Bei der nativen HA handelt es sich um eine Hochverfügbarkeitslösung, die sich an Containerbereitstellungen für IBM MQ richtet. Bei der nativen HA werden mit der Protokollreplikation drei Instanzen eines Warteschlangenmanagers, die auf unterschiedlichen Knoten ausgeführt werden, auf dem aktuellsten Stand gehalten. Eine Instanz ist zu jedem Zeitpunkt aktiv und verarbeitet Nachrichten. Der aktive Warteschlangenmanager sendet seine Protokollaktualisierungen an die anderen beiden Instanzen, damit diese aktualisiert bleiben. Wenn die aktive Instanz fehlschlägt, übernimmt eine der Replikatsinstanzen automatisch die aktive Rolle.

## Unterschiede zwischen Multi-Instanz-WS-Managern und HA-Clustern

Multi-Instanz-WS-Manager und HA-Cluster sind alternative Methoden, um hohe Verfügbarkeit für Ihre Warteschlangenmanager zu erreichen. Hier sind einige Punkte, die die Unterschiede zwischen den beiden Ansätzen hervorheben.

Multi-Instanz-Warteschlangenmanager enthalten die folgenden Funktionen:

- In IBM MQ integrierte Basisunterstützung für Funktionsübernahme
- Schnellere Übernahme als HA-Cluster
- Einfache Konfiguration und Bedienung
- Integration in IBM MQ Explorer

Die Einschränkungen bei Warteschlangenmanagern mit mehreren Instanzen umfassen:

- Hochverfügbarer, leistungsstark vernetzter Netzspeicher erforderlich
- Komplexere Netzkonfiguration, da die IP-Adresse des WS-Managers bei einem Ausfall geändert wird

HA-Cluster enthalten die folgenden Funktionen:

- Die Fähigkeit, mehrere Ressourcen, wie z. B. einen Anwendungsserver oder eine Datenbank, zu koordinieren.
- Flexibler Konfigurationsoptionen, einschließlich Cluster mit mehr als zwei Knoten
- Funktionsübernahme mehrfach ohne Bedienereingriff möglich
- Übernahme der IP-Adresse des WS-Managers als Teil der Funktionsübernahme

Zu den Einschränkungen bei HA-Clustern gehören:

- Weitere Produktkäufe und -qualifikationen sind erforderlich
- Disks, die zwischen den Knoten des Clusters umgeschaltet werden können, sind erforderlich.
- Konfiguration von HA-Clustern ist relativ komplex
- Failover ist historisch eher langsam, aber die neuesten HA-Cluster-Produkte verbessern diese
- Unnötige Failover können auftreten, wenn in den Scripts, die zum Überwachen von Ressourcen wie Warteschlangenmanagern verwendet werden, Mängel vorhanden sind.

## Beziehung zwischen HA-Clustern und WS-Manager-Clustern

WS-Manager-Cluster stellen den Lastausgleich von Nachrichten über verfügbare Instanzen von WS-Manager-Clusterwarteschlangen hinweg bereit. Dies bietet eine höhere Verfügbarkeit als ein einzelner Warteschlangenmanager, da die Messaging-Anwendungen nach einem Ausfall eines Warteschlangenmanagers weiterhin Nachrichten an die überlebenden Instanzen einer WS-Manager-Clusterwarteschlange senden können und auf diese zugreifen können. Obwohl WS-Manager-Cluster jedoch automatisch neue Nachrichten an die verfügbaren Warteschlangenmanager in einem Cluster weiterleiten, stehen Nachrichten, die sich derzeit in einem nicht verfügbaren Warteschlangenmanager in der Warteschlange befinden, erst wieder zur Verfügung, wenn dieser Warteschlangenmanager erneut gestartet wird. Aus diesem Grund bieten Warteschlangenmanager-Cluster allein keine hohe Verfügbarkeit aller Nachrichtendaten oder die automatische Erkennung des Warteschlangenmanagerfehlers und die automatische Auslösung von WS-Manager-Neustarts oder -Failover. Hochverfügbarkeitscluster (HA) stellen diese Funktionen bereit. Die beiden Typen von Clustern können gemeinsam genutzt werden, um gute Wirkung zu erreichen. Eine Einführung in WS-Manager-Cluster finden Sie unter Cluster entwerfen .

**Zugehörige Konzepte**

Linux MQ Adv. CD Hohe Verfügbarkeit für IBM MQ Advanced container

## Linux AIX HA-Cluster unter AIX and Linux

Sie können IBM MQ mit einem Hochverfügbarkeitscluster auf AIX and Linux-Plattformen verwenden: z. B. PowerHA für AIX (vormals HACMP ), Veritas Cluster Server, HP Serviceguard oder ein Red Hat Enterprise Linux-Cluster mit Red Hat Cluster Suite.

Dieser Abschnitt enthält eine Einführung in „HA-Clusterkonfigurationen” auf Seite 517, die Beziehung von HA-Clustern zu Warteschlangenmanagerclustern, „IBM MQ-Clients” auf Seite 517 und „IBM MQ in einem HA-Cluster” auf Seite 518 und führt Sie durch die Schritte und stellt Beispielscripts bereit, die Sie anpassen können, um Warteschlangenmanager mit einem HA-Cluster zu konfigurieren.

Weitere Informationen zu den in diesem Abschnitt beschriebenen Konfigurationsschritten finden Sie in der HA-Clusterdokumentation zu Ihrer Umgebung.

## HA-Clusterkonfigurationen

In diesem Abschnitt wird der Begriff *Knoten* verwendet, um auf die Entität zu verweisen, auf der ein Betriebssystem und die HA-Software ausgeführt werden; "Computer", "System" oder "Maschine" oder "Partition" oder "Blade" können in dieser Verwendung als Synonyme betrachtet werden. Sie können IBM MQ verwenden, um die Konfiguration von Standby- oder Übernahmekonfigurationen zu unterstützen, einschließlich der gegenseitigen Übernahme, bei der alle Clusterknoten die Workload von IBM MQ ausführen.

Eine *Standby*-Konfiguration ist die grundlegendste HA-Clusterkonfiguration, in der ein Knoten funktioniert, während der andere Knoten nur als Standby fungiert. Der Standby-Knoten führt keine Arbeit aus und wird als inaktiv bezeichnet. Diese Konfiguration wird auch als *Cold-Standby* bezeichnet. Eine solche Konfiguration erfordert einen hohen Grad an Redundanz der Hardware. Um Hardware einzusparen, ist es möglich, diese Konfiguration so zu erweitern, dass sie mehrere Workerknoten mit einem einzigen Standby-Knoten hat. Der Punkt in diesem Punkt ist, dass der Standby-Knoten die Arbeit eines anderen Workerknotens übernehmen kann. Diese Konfiguration wird immer noch als Standby-Konfiguration und manchmal auch als "N + 1" -Konfiguration bezeichnet.

Eine *Übernahmekonfiguration* ist eine erweiterte Konfiguration, bei der alle Knoten einige Arbeiten ausführen und kritische Arbeiten im Falle eines Knotenausfalls übernommen werden können.

Eine *einseitige Übernahme* ist eine Konfiguration, bei der ein Standby-Knoten einige zusätzliche, unkritische und unbewegliche Arbeit ausführt. Diese Konfiguration gleicht einer Standby-Konfiguration, aber mit (unkritischen) Arbeiten, die vom Standby-Knoten ausgeführt werden.

Eine *gegenseitige Übernahme* -Konfiguration ist eine Konfiguration, in der alle Knoten hochverfügbare (bewegliche) Arbeiten ausführen. Dieser Typ der HA-Clusterkonfiguration wird auch manchmal als "Aktiv/Aktiv" bezeichnet, um anzuzeigen, dass alle Knoten die kritische Auslastung aktiv verarbeiten.

Bei der erweiterten Bereitschaftskonfiguration oder bei einer der beiden Übernahmekonfigurationen ist es wichtig, die Spitzenlast zu berücksichtigen, die auf einem Knoten platziert werden kann, der die Arbeit anderer Knoten übernehmen kann. Ein solcher Knoten muss über eine ausreichende Kapazität verfügen, um ein akzeptables Leistungsniveau zu gewährleisten.

## Beziehung zwischen HA-Clustern und WS-Manager-Clustern

WS-Manager-Cluster reduzieren die Verwaltung und stellen den Lastausgleich von Nachrichten über Instanzen von WS-Manager-Clusterwarteschlangen hinweg bereit. Sie bieten auch eine höhere Verfügbarkeit als ein einzelner Warteschlangenmanager, da die Messaging-Anwendungen nach einem Ausfall eines Warteschlangenmanagers immer noch auf überlebenslange Instanzen einer WS-Manager-Clusterwarteschlange zugreifen können. Allerdings bieten WS-Manager-Cluster allein keine automatische Erkennung des Warteschlangenmanagerfehlers und die automatische Auslösung von WS-Manager-Neustarts oder Failover. HA-Cluster stellen diese Funktionen bereit. Die beiden Typen von Clustern können gemeinsam genutzt werden, um gute Wirkung zu erreichen.

## IBM MQ-Clients

IBM MQ-Clients, die mit einem Warteschlangenmanager kommunizieren, der möglicherweise einem Neustart oder einer Übernahme unterzogen wird, müssen geschrieben werden, um eine unterbrochene Verbindung zu tolerieren, und wiederholt versuchen, die Verbindung erneut herzustellen. IBM MQenthält Features in der Verarbeitung der Definitionstabelle für Clientkanäle (CCDT), die Sie bei der Verbindungsverfügbarkeit und beim Lastausgleich unterstützen. Diese sind jedoch nicht direkt relevant, wenn Sie mit einem Failover-System arbeiten.

Die transaktionsorientierte Funktionalität ermöglicht es einem IBM MQ MQI client, an zweiphasigen Transaktionen teilzunehmen, solange der Client mit demselben Warteschlangenmanager verbunden ist. Die transaktionsorientierte Funktionalität kann keine Verfahren verwenden, z. B. eine IP-Lastausgleichsfunktion, um eine Auswahl aus einer Liste von Warteschlangenmanagern zu treffen. Wenn Sie ein HA-Produkt verwenden, behält ein Warteschlangenmanager seine Identität (Name und Adresse) bei, je nachdem, auf welchem Knoten er ausgeführt wird, so dass transaktionsorientierte Funktionen mit Warteschlangenmanagern verwendet werden können, die unter HA-Steuerung stehen.

## IBM MQ in einem HA-Cluster

Alle HA-Cluster haben das Konzept einer Funktionseinheit. Dies ist eine Gruppe von Definitionen, die alle Ressourcen enthält, aus denen der hoch verfügbare Service besteht. Die Einheit der Funktionsübernahme schließt den Service selbst sowie alle anderen Ressourcen ein, von denen er abhängig ist.

HA-Lösungen verwenden unterschiedliche Bedingungen für eine Funktionseinheit:

- Unter PowerHA für AIX wird die Funktionseinheit als *Ressourcengruppe* bezeichnet.
- Auf Veritas Cluster Server wird sie als *Servicegruppe* bezeichnet.
- Auf Serviceguard wird sie als *Paket* bezeichnet.

In diesem Abschnitt wird der Begriff *Ressourcengruppe* verwendet, um eine Funktionseinheit zu definieren.

Die kleinste Einheit der Funktionsübernahme für IBM MQ ist ein Warteschlangenmanager. In der Regel enthält die Ressourcengruppe, die den Warteschlangenmanager enthält, auch gemeinsam genutzte Platten in einer Datenträgergruppe oder Plattengruppe, die ausschließlich für die Verwendung durch die Ressourcengruppe reserviert ist, sowie die IP-Adresse, die für die Verbindung zum Warteschlangenmanager verwendet wird. Es ist auch möglich, andere IBM MQ-Ressourcen, wie z. B. einen Listener oder einen Auslösemonitor in derselben Ressourcengruppe, entweder als separate Ressourcen oder unter der Steuerung des Warteschlangenmanagers selbst, einzuschließen.

Ein Warteschlangenmanager, der in einem HA-Cluster verwendet werden soll, muss über seine Daten und Protokolle auf Platten verfügen, die zwischen den Knoten im Cluster gemeinsam genutzt werden. Der HA-Cluster stellt sicher, dass nur ein Knoten im Cluster zu einem Zeitpunkt auf die Platten schreiben kann. Der HA-Cluster kann ein Überwachungsscript verwenden, um den Status des Warteschlangenmanagers zu überwachen.

Es ist möglich, sowohl für die Daten als auch für die Protokolle, die sich auf den Warteschlangenmanager beziehen, eine einzige gemeinsam genutzte Platte zu verwenden. Es ist jedoch üblich, separate gemeinsam genutzte Dateisysteme zu verwenden, so dass sie unabhängig voneinander dimensioniert und optimiert werden können.

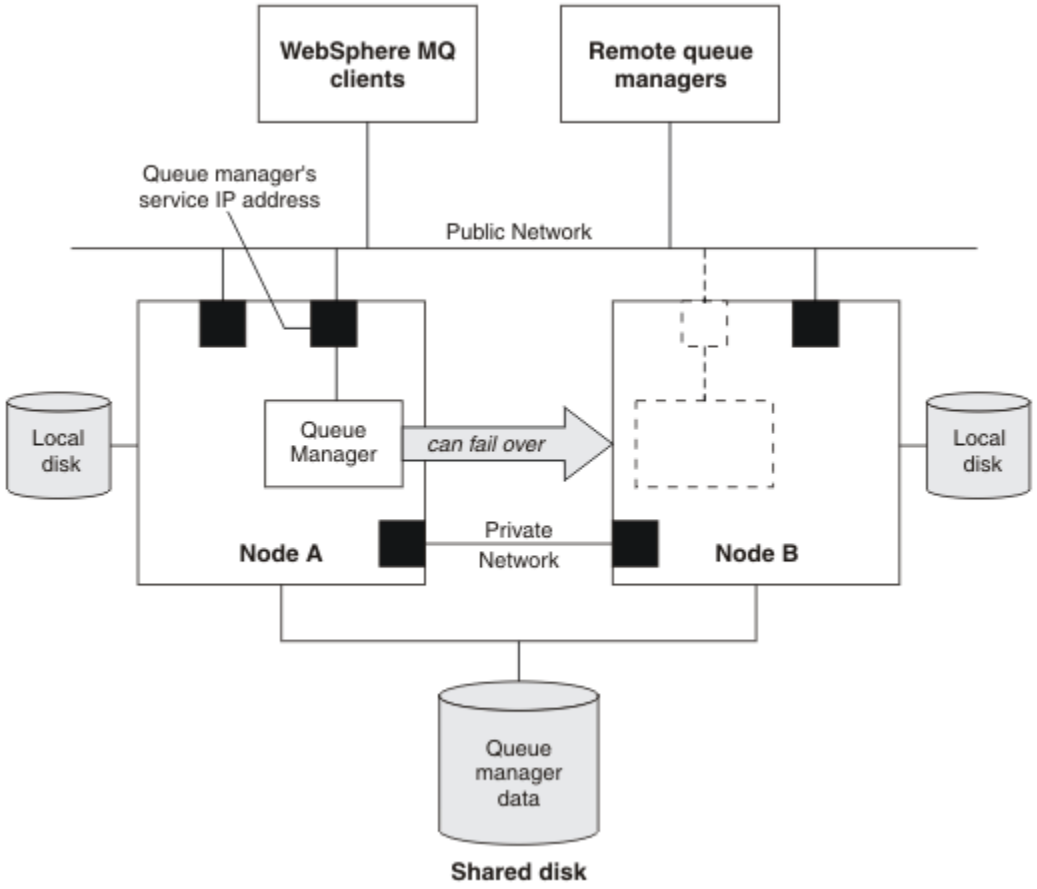



*Abbildung 69. HV-Cluster*

In Abbildung 1 ist ein HA-Cluster mit zwei Knoten dargestellt. Der HA-Cluster verwaltet die Verfügbarkeit eines Warteschlangenmanagers, der in einer Ressourcengruppe definiert wurde. Hierbei handelt es sich um eine Aktiv/Passiv-oder eine Cold-Standby-Konfiguration, da momentan nur ein Knoten, Knoten A, einen Warteschlangenmanager ausführt. Der WS-Manager wurde mit seinen Daten und Protokolldateien auf einer gemeinsam genutzten Platte erstellt. Der WS-Manager verfügt über eine Service-IP-Adresse, die auch vom HA-Cluster verwaltet wird. Der Warteschlangenmanager ist von der gemeinsam genutzten Platte und der zugehörigen Service-IP-Adresse abhängig. Wenn der HA-Cluster den WS-Manager von Knoten A zu Knoten B nicht mehr absetzt, verschiebt er zunächst die abhängigen Ressourcen des WS-Managers auf den Knoten B und startet dann den Warteschlangenmanager.

Wenn der HA-Cluster mehr als einen Warteschlangenmanager enthält, kann Ihre HA-Clusterkonfiguration dazu führen, dass zwei oder mehr Warteschlangenmanager nach einem Failover auf demselben Knoten ausgeführt werden. Jedem WS-Manager im HA-Cluster muss eine eigene Portnummer zugeordnet werden, die er für den jeweils aktiven Clusterknoten verwendet, der zu einem bestimmten Zeitpunkt aktiv ist.

Im Allgemeinen wird der HA-Cluster als Rootbenutzer ausgeführt. IBM MQ wird als mqm-Benutzer ausgeführt. Die Verwaltung von IBM MQ wird Mitgliedern der Gruppe 'mqm' erteilt. Stellen Sie sicher, dass der Benutzer und die Gruppe mqm auf allen HA-Clusterknoten vorhanden sind. Die Benutzer-ID und die Gruppen-ID müssen im gesamten Cluster konsistent sein. Die Verwaltung von IBM MQ durch den Rootbenutzer ist nicht zulässig; Scripts, die Scripts starten, stoppen oder überwachen, müssen auf den mqm-Benutzer umschalten.

**Anmerkung:** IBM MQ muss auf allen Knoten ordnungsgemäß installiert sein. Sie können die ausführbaren Produktdateien nicht gemeinsam nutzen.

### Linux AIX *Gemeinsam genutzte Platten auf AIX and Linux konfigurieren*

Ein IBM MQ-Warteschlangenmanager in einem HA-Cluster erfordert Datendateien und Protokolldateien, um gemeinsam benannte ferne Dateisysteme auf einer gemeinsam genutzten Platte zu verwenden.

## Informationen zu diesem Vorgang

Abbildung 1 zeigt ein mögliches Layout für einen WS-Manager in einem HA-Cluster. Die Daten-und Protokollverzeichnisse des Warteschlangenmanagers befinden sich auf der gemeinsam genutzten Platte, die unter /MQHA/QM1 angehängt ist. Diese Platte wird bei einem Failover zwischen den Knoten des HA-Clusters umgeschaltet, so dass die Daten überall dort verfügbar sind, wo der WS-Manager erneut gestartet wird. Die `mqs.ini`-Datei enthält eine Zeilengruppe für den QM1-Warteschlangenmanager. Die Zeilengruppe 'Log' in der Datei `qm.ini` hat einen Wert für 'LogPath'.

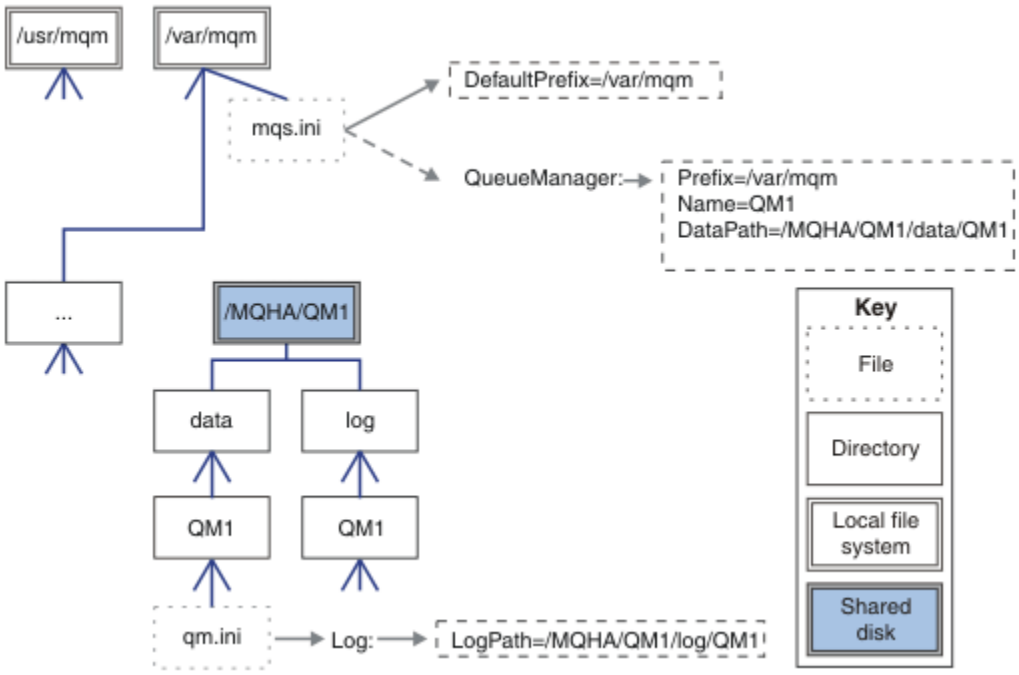



*Abbildung 70. Gemeinsam genutzte Verzeichnisse mit dem Namen `data` und `log`*

## Vorgehensweise

1. Entscheiden Sie die Namen der Mountpunkte für die Dateisysteme des Warteschlangenmanagers.

   Beispiel: `/MQHA/qmgrname/data` für die Datendateien des WS-Managers und `/MQHA/qmgrname/log` für seine Protokolldateien.
2. Erstellen Sie eine Datenträgergruppe (oder eine Plattengruppe), die die Daten-und Protokolldateien des Warteschlangenmanagers enthält.

   Diese Datenträgergruppe wird vom HA-Cluster (HA-Cluster) in derselben Ressourcengruppe wie der Warteschlangenmanager verwaltet.
3. Erstellen Sie die Dateisysteme für die Daten und Protokolldateien des Warteschlangenmanagers in der Datenträgergruppe.
4. Erstellen Sie für jeden Knoten wiederum die Mountpunkte für die Dateisysteme, und stellen Sie sicher, dass die Dateisysteme angehängt werden können.

   Der mqm-Benutzer muss Eigner der Mountpunkte sein.

# Linux AIX *HA-Cluster-WS-Manager unter AIX and Linux erstellen*

Der erste Schritt bei der Verwendung eines Warteschlangenmanagers in einem Hochverfügbarkeitscluster ist die Erstellung des Warteschlangenmanagers auf einem der Knoten.

## Informationen zu diesem Vorgang

Wenn Sie einen Warteschlangenmanager für die Verwendung in einem HA-Cluster erstellen möchten, müssen Sie zuerst einen der Knoten im Cluster auswählen, auf dem der Warteschlangenmanager erstellt werden soll, und anschließend die folgenden Schritte auf diesem Knoten ausführen.

## Vorgehensweise

1. Hängen Sie die Dateisysteme des Warteschlangenmanagers auf dem Knoten an.
2. Erstellen Sie den WS-Manager mit dem Befehl **crtmqm** .
   For example:

   ```
   crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
   ```
3. Starten Sie den Warteschlangenmanager manuell mit dem Befehl **strmqm** .
4. Führen Sie eine beliebige Erstkonfiguration des Warteschlangenmanagers aus, z. B. die Erstellung von Warteschlangen und Kanälen und die Einstellung des Warteschlangenmanagers, um einen Listener automatisch zu starten, wenn der Warteschlangenmanager gestartet wird.
5. Stoppen Sie den WS-Manager mit dem Befehl **endmqm** .
6. Verwenden Sie den Befehl **dspmqinf** , um den Befehl **addmqinf** anzuzeigen:

   ```
   dspmqinf -o command qmgrname
   ```

   Dabei ist `qmgrname` der Name des Warteschlangenmanagers.

   Weitere Informationen zur Verwendung des Befehls **addmqinf** finden Sie unter „WS-Manager-Konfiguration zu anderen HA-Clusterknoten unter AIX and Linux hinzufügen” auf Seite 521.

   Der Befehl **addmqinf** wird ähnlich wie im folgenden Beispiel angezeigt:

   ```
   addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \
   -vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
   ```
7. Notieren Sie den angezeigten Befehl sorgfältig.
8. Entladen Sie die Dateisysteme des Warteschlangenmanagers.

## Nächste Schritte

Sie sind jetzt bereit, die in „WS-Manager-Konfiguration zu anderen HA-Clusterknoten unter AIX and Linux hinzufügen” auf Seite 521 beschriebenen Schritte auszuführen.

# Linux AIX *WS-Manager-Konfiguration zu anderen HA-Clusterknoten unter AIX and Linux hinzufügen*

Sie müssen die WS-Manager-Konfigurationsdaten zu den anderen Knoten im HA-Cluster hinzufügen.

## Vorbereitende Schritte

Bevor Sie diese Task ausführen können, müssen Sie die Schritte in „HA-Cluster-WS-Manager unter AIX and Linux erstellen” auf Seite 520 ausgeführt haben. Nachdem Sie den Warteschlangenmanager erstellt haben, müssen Sie die Konfigurationsinformationen für den Warteschlangenmanager zu jedem anderen Knoten im HA-Cluster hinzufügen, indem Sie die folgenden Schritte auf jedem der anderen Knoten ausführen.

## Informationen zu diesem Vorgang

Wenn Sie einen Warteschlangenmanager für die Verwendung in einem HA-Cluster erstellen, müssen Sie zuerst einen der Knoten in dem Cluster auswählen, auf dem der Warteschlangenmanager erstellt werden soll, wie im Abschnitt „HA-Cluster-WS-Manager unter AIX and Linux erstellen” auf Seite 520beschrieben.

## Vorgehensweise

1. Hängen Sie die WS-Manager-Dateisysteme an.
2. Fügen Sie die WS-Manager-Konfigurationsinformationen zum Knoten hinzu.
   Es gibt zwei Möglichkeiten, die Konfigurationsinformationen hinzuzufügen:
   - Durch direkte Bearbeitung von `/var/mqm/mqs.ini`.

- Durch Absetzen des Befehls **addmqinf**, der durch den Befehl **dspmqinf** in Schritt 6 in „HA-Cluster-WS-Manager unter AIX and Linux erstellen” auf Seite 520 angezeigt wurde.

3. Starten und stoppen Sie den WS-Manager, um die Konfiguration zu überprüfen.

   Die Befehle, die zum Starten und Stoppen des Warteschlangenmanagers verwendet werden, müssen von derselben IBM MQ-Installation wie der Befehl **addmqinf** ausgegeben werden. Zum Starten und Stoppen des Warteschlangenmanagers aus einer anderen Installation als der, die dem Warteschlangenmanager derzeit zugeordnet ist, müssen Sie zuerst die Installation festlegen, die dem Warteschlangenmanager zugeordnet ist, mit dem Befehl **setmqm** . Weitere Informationen finden Sie in setmqm .
4. Entpacken Sie die WS-Manager-Dateisysteme.

### Linux AIX *Beispiel-Shell-Scripts zum Starten eines HA-Cluster-Warteschlangenmanagers unter AIX and Linux*

Der WS-Manager wird im HA-Cluster als Ressource dargestellt. Der HA-Cluster muss in der Lage sein, den Warteschlangenmanager zu starten und zu stoppen. In den meisten Fällen können Sie ein Shell-Script zum Starten des Warteschlangenmanagers verwenden. Sie müssen diese Scripts an derselben Position auf allen Knoten im Cluster verfügbar machen, indem Sie entweder ein Netzdateisystem verwenden oder sie auf jede der lokalen Platten kopieren.

**Anmerkung:** Bevor Sie einen fehlgeschlagenen Warteschlangenmanager erneut starten, müssen Sie die Verbindung zu den Anwendungen von dieser Instanz des Warteschlangenmanagers trennen. Ist dies nicht der Fall, wird der WS-Manager möglicherweise nicht ordnungsgemäß erneut gestartet.

Beispiele für geeignete Shell-Scripts sind hier angegeben. Sie können diese an Ihre Anforderungen anpassen und sie zum Starten des Warteschlangenmanagers unter der Steuerung Ihres HA-Clusters verwenden.

Das folgende Shell-Script ist ein Beispiel dafür, wie Sie vom HA-Cluster-Benutzer zum mqm-Benutzer wechseln können, damit der Warteschlangenmanager erfolgreich gestartet werden kann:

```
#!/bin/ksh

# A simple wrapper script to switch to the mqm user.

su mqm -c name_of_your_script $*
```

Das folgende Shell-Script ist ein Beispiel dafür, wie ein Warteschlangenmanager gestartet wird, ohne Annahmen über den aktuellen Status des Warteschlangenmanagers zu treffen. Beachten Sie, dass es eine extrem abrupte Methode verwendet, alle Prozesse zu beenden, die zum Warteschlangenmanager gehören:

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.

# The only argument is the queue manager name. Save it as QM variable
QM=$1

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

# End any queue manager processes which might be running.

srchstr="( |-m)$QM *.*$"
for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
 do
  ps -ef | tr "\t" " " | grep $process | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}'| \
       xargs kill -9 > /dev/null 2>&1
```

```
done

# It is now safe to start the queue manager.
# The strmqm command does not use the -x flag.
strmqm ${QM}
```

Sie können das Script ändern, um andere zugehörige Programme zu starten.

### Linux AIX *Beispiel-Shell-Script zum Stoppen eines HA-Cluster-Warteschlangenmanagers unter AIX and Linux*

In den meisten Fällen können Sie ein Shell-Script verwenden, um einen Warteschlangenmanager zu stoppen. Beispiele für geeignete Shell-Scripts sind hier angegeben. Sie können diese an Ihre Anforderungen anpassen und sie verwenden, um den Warteschlangenmanager unter der Steuerung Ihres HA-Clusters zu stoppen.

Das folgende Script ist ein Beispiel dafür, wie ein Warteschlangenmanager sofort gestoppt werden kann, ohne Annahmen über den aktuellen Status des Warteschlangenmanagers zu treffen. Das Script muss von dem Benutzer mqm ausgeführt werden. Es kann daher erforderlich sein, dieses Script in ein Shell-Script einzuschließen, um den Benutzer vom HA-Cluster-Benutzer auf mqm umzuschalten. (Ein Beispiel für ein Shell-Script wird in „Beispiel-Shell-Scripts zum Starten eines HA-Cluster-Warteschlangenmanagers unter AIX and Linux” auf Seite 522 bereitgestellt.)

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$TIMEOUT" ]
then
  echo "ERROR! No timeout specified"
  exit 1
fi

for severity in immediate brutal
do
  # End the queue manager in the background to avoid
  # it blocking indefinitely. Run the TIMEOUT timer
  # at the same time to interrupt the attempt, and try a
  # more forceful version. If the brutal version fails,
  # nothing more can be done here.

  echo "Attempting ${severity} end of queue manager '${QM}'"
  case $severity in

  immediate)
    # Minimum severity of endmqm is immediate which severs connections.
    # HA cluster should not be delayed by clients
    endmqm -i ${QM} &
    ;;

  brutal)
    # This is a forced means of stopping queue manager processes.

    srchstr="( |-m)$QM *.*$"
    for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
                amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
                amqzfuma amqzmuf0 amqzmur0 amqzmgr0
    do
      ps -ef | tr "\t" " " | grep $process | grep -v grep | \
          egrep "$srchstr" | awk '{print $2}'| \
             xargs kill -9 > /dev/null 2>&1
```

```
    done

  esac

  TIMED_OUT=yes
  SECONDS=0
  while (( $SECONDS < ${TIMEOUT} ))
  do
   TIMED_OUT=yes
   i=0
   while [ $i -lt 5 ]
   do
     # Check for execution controller termination
     srchstr="( |-m)$QM *.*$"
     cnt=`ps -ef | tr "\t" " " | grep amqzxma0 | grep -v grep | \
       egrep "$srchstr" | awk '{print $2}' | wc -l `
     i=`expr $i + 1`
     sleep 1
     if [ $cnt -eq 0 ]
     then
       TIMED_OUT=no
       break
     fi
   done

   if [ ${TIMED_OUT} = "no" ]
   then
     break
   fi

   echo "Waiting for ${severity} end of queue manager '${QM}'"
   sleep 1
  done # timeout loop

  if [ ${TIMED_OUT} = "yes" ]
  then
    continue        # to next level of urgency
  else
    break           # queue manager is ended, job is done
  fi

done # next phase
```

**Anmerkung:** Abhängig von den Prozessen, die für einen bestimmten Warteschlangenmanager ausgeführt werden, kann die Liste der WS-Manager-Prozesse, die in diesem Script enthalten sind, entweder nicht vollständig sein oder mehr Prozesse enthalten, als die Prozesse, die für diesen Warteschlangenmanager ausgeführt werden:

```
for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
```

Ein Prozess kann abhängig davon, welche Funktion konfiguriert ist und welche Prozesse für einen bestimmten Warteschlangenmanager ausgeführt werden, in diese Liste aufgenommen oder aus dieser Liste ausgeschlossen werden. Eine vollständige Liste der Prozesse und Informationen zum Stoppen der Prozesse in einer bestimmten Reihenfolge finden Sie im Abschnitt Warteschlangenmanager manuell unter UNIX und Linux stoppen.

### Linux AIX *HA-Cluster-Warteschlangenmanager unter AIX and Linux überwachen*

Es ist üblich, einen Weg für den Hochverfügbarkeitscluster bereitzustellen, um den Status des Warteschlangenmanagers in regelmäßigen Abständen zu überwachen. In den meisten Fällen können Sie hierfür ein Shell-Script verwenden. Beispiele für geeignete Shell-Scripts sind hier angegeben. Sie können diese Scripts an Ihre Anforderungen anpassen und sie verwenden, um zusätzliche Überwachungsprüfungen speziell für Ihre Umgebung zu erstellen.

Es ist möglich, dass mehrere Installationen von IBM MQ auf einem System koexistieren. Weitere Informationen zu mehreren Installationen finden Sie unter Mehrere Installationen. Wenn Sie das Überwachungsscript für mehrere Installationen verwenden möchten, müssen Sie möglicherweise einige zusätzliche Schritte ausführen. Wenn Sie über eine primäre Installation verfügen, müssen Sie *MQ_INSTALLATI-*

*ON_PATH* nicht angeben, um das Script zu verwenden. Verwenden Sie andernfalls die folgenden Schritte, um sicherzustellen, dass die *MQ_INSTALLATION_PATH* ordnungsgemäß identifiziert wird:

1. Verwenden Sie den Befehl **crtmqenv** von einer IBM MQ-Installation aus, um den richtigen *MQ_INSTALLATION_PATH* für einen Warteschlangenmanager zu identifizieren:

   ```
   crtmqenv -m qmname
   ```

   Dieser Befehl gibt den richtigen *MQ_INSTALLATION_PATH*-Wert für den mit *qmname* angegebenen Warteschlangenmanager zurück.
2. Führen Sie das Überwachungsscript mit dem entsprechenden *qmname* und den entsprechenden *MQ_INSTALLATION_PATH*-Parametern aus.

**Anmerkung:** PowerHA für AIX stellt keine Möglichkeit zur Verfügung, einen Parameter für das Überwachungsprogramm für den Warteschlangenmanager bereitzustellen. Sie müssen für jeden WS-Manager ein separates Überwachungsprogramm erstellen, das den Namen des WS-Managers einbindet. Im Folgenden finden Sie ein Beispiel für ein Script, das in AIX verwendet wird, um den Namen des Warteschlangenmanagers einzubinden:

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname  MQ_INSTALLATION_PATH
```

Dabei steht *MQ_INSTALLATION_PATH* für einen optionalen Parameter, der den Pfad zur Installation von IBM MQ angibt, die dem Warteschlangenmanager *qmname* zugeordnet ist.

Das folgende Script ist nicht robust gegenüber der Möglichkeit, dass **runmqsc** blockiert. In der Regel behandeln HA-Cluster ein blockes Überwachungsscript als Fehler und sind selbst robust gegenüber dieser Möglichkeit.

Das Script toleriert jedoch, dass sich der WS-Manager im Startstatus befindet. Dies liegt daran, dass der HA-Cluster die Überwachung des WS-Managers so schnell wie möglich gestartet hat. Einige HA-Cluster unterscheiden zwischen einer Startphase und einer aktiven Phase für Ressourcen, aber es ist notwendig, die Dauer der Startphase zu konfigurieren. Da die Zeit, die zum Starten eines Warteschlangenmanagers benötigt wird, von der Anzahl der zu ergreifenden Arbeiten abhängt, ist es schwierig, eine maximale Zeit für den Start eines Warteschlangenmanagers auszuwählen. Wenn Sie einen zu niedrigen Wert auswählen, geht der HA-Cluster fälschlicherweise davon aus, dass der WS-Manager nicht gestartet wurde, wenn er nicht vollständig ausgeführt wurde. Dies könnte zu einer endlosen Folge von Failover führen.

Dieses Script muss vom mqm-Benutzer ausgeführt werden. Es kann daher erforderlich sein, dieses Script in ein Shell-Script einzuschließen, um den Benutzer vom HA-Clusterbenutzer zu mqm zu wechseln (ein Beispiel-Shell-Script wird in „Beispiel-Shell-Scripts zum Starten eines HA-Cluster-Warteschlangenmanagers unter AIX and Linux” auf Seite 522 bereitgestellt):

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0  => Either the queue manager is starting or the queue manager is running and responds.
#       Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
  # No path specified, assume system primary install or MQ level < 7.1.0
  echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
```

```
else
  echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
  PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

  echo "Queue manager '${QM}' is responsive"
  result=0

else # ping failed

  # Don't condemn the queue manager immediately, it might be starting.
  srchstr="( |-m)$QM *.*$"
  cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
              | awk '{print $2}' | wc -l`
  if [ $cnt -gt 0 ]
  then
    # It appears that the queue manager is still starting up, tolerate
    echo "Queue manager '${QM}' is starting"
    result=0
  else
    # There is no sign of the queue manager starting
    echo "Queue manager '${QM}' is not responsive"
    result=$pingresult
  fi

fi

exit $result
```

## Linux AIX *Warteschlangenmanager unter AIX and Linux unter die Steuerung des HA-Clusters versetzen*

Sie müssen den Warteschlangenmanager unter der Steuerung des HA-Clusters mit der IP-Adresse und den gemeinsam genutzten Platten des WS-Managers konfigurieren.

### Informationen zu diesem Vorgang

Um den Warteschlangenmanager unter die Steuerung des Hochverfügbarkeitsclusters zu versetzen, müssen Sie eine Ressourcengruppe definieren, die den Warteschlangenmanager und alle zugehörigen Ressourcen enthält.

### Vorgehensweise

1. Erstellen Sie die Ressourcengruppe, die den Warteschlangenmanager, den Datenträger oder die Plattengruppe des Warteschlangenmanagers und die IP-Adresse des WS-Managers enthält.

   Die IP-Adresse ist eine virtuelle IP-Adresse, nicht die IP-Adresse des Computers.
2. Stellen Sie sicher, dass der HA-Cluster die Ressourcen zwischen den Clusterknoten korrekt umschaltet und bereit ist, den Warteschlangenmanager zu steuern.

## Linux AIX *HA-Cluster-Warteschlangenmanager unter AIX and Linux löschen*

Möglicherweise möchten Sie einen Warteschlangenmanager von einem Knoten entfernen, der nicht mehr für die Ausführung des Warteschlangenmanagers erforderlich ist.

### Informationen zu diesem Vorgang

Wenn Sie den Warteschlangenmanager von einem Knoten in einem Hochverfügbarkeitscluster entfernen möchten, müssen Sie die zugehörigen Konfigurationsdaten entfernen.

### Vorgehensweise

1. Entfernen Sie den Knoten aus dem HA-Cluster, so dass der HA-Cluster nicht mehr versuchen wird, den Warteschlangenmanager auf diesem Knoten zu aktivieren.
2. Verwenden Sie den folgenden **rmvmqinf** -Befehl, um die Konfigurationsdaten des Warteschlangenmanagers zu entfernen:

   ```
   rmvmqinf qmgrname
   ```
3. Optional: Verwenden Sie den Befehl **dltmqm** , um den Warteschlangenmanager vollständig zu löschen.

   **Wichtig:** Beachten Sie, dass beim Löschen des Warteschlangenmanagers mit dem Befehl **dltmqm** die Daten und Protokolldateien des Warteschlangenmanagers vollständig gelöscht werden.

   Nachdem Sie den WS-Manager gelöscht haben, können Sie den Befehl **rmvmqinf** verwenden, um verbleibende Konfigurationsdaten von den anderen Knoten zu entfernen.

## Windows Unterstützung für Microsoft Cluster Service (MSCS)

Einführung und Konfiguration von MSCS für die Unterstützung der Funktionsübernahme von virtuellen Servern. MSCS wird auch als Windows Server Failover Clustering (WSFC) bezeichnet.

Diese Informationen beziehen sich nur auf IBM MQ for Windows.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Mit MSCS/WSFC können Sie Server in einem Cluster verbinden, wodurch eine höhere Verfügbarkeit von Daten und Anwendungen ermöglicht wird und die Verwaltung des Systems vereinfacht wird. MSCS/WSFC kann Server-oder Anwendungsfehler automatisch erkennen und wiederherstellen.

MSCS/WSFC unterstützt die Funktionsübernahme von virtuellen Servern, die Anwendungen, Websites, Druckwarteschlangen oder Dateifreigaben entsprechen (z. B. Plattenspindeln, Dateien und IP-Adressen).

*Failover* ist der Prozess, mit dessen Hilfe MSCS/WSFC einen Fehler in einer Anwendung auf einem Computer im Cluster erkennt und die unterbrochene Anwendung ordnungsgemäß beendet, ihre Statusdaten an den anderen Computer überträgt und die Anwendung dort erneut initialisiert.

Informationen zur Konfiguration und Verwendung von Failover-Clustern finden Sie in den Unterabschnitten.

### Windows *Einführung in MSCS-Cluster*

Microsoft Cluster Service -Cluster (MSCS-Cluster) sind Gruppen von zwei oder mehr Computern, die miteinander verbunden und so konfiguriert sind, dass MSCS bei einem Ausfall eine *Funktionsübernahme*durchführt, die Statusdaten von Anwendungen vom fehlerhaften Computer auf einen anderen Computer im Cluster überträgt und dort erneut einleitet.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

„Hochverfügbarkeitskonfigurationen” auf Seite 514 enthält einen Vergleich zwischen MSCS-Clustern, Multi-Instanz-Warteschlangenmanagern und IBM MQ-Clustern.

In diesem Abschnitt und seinen untergeordneten Themen bedeutet der Begriff *Cluster* , wenn er von sich selbst verwendet wird, **immer** einen MSCS-Cluster. Dies stellt einen Unterschied zu einem IBM MQ-Cluster dar, der an anderer Stelle in diesem Handbuch beschrieben wird.

Ein Cluster mit zwei Maschinen besteht aus zwei Computern (z. B. A und B), die gemeinsam mit einem Netz für den Clientzugriff über eine *virtuelle IP-Adresse* verbunden sind. Sie können auch durch ein oder mehrere private Netze miteinander verbunden sein. A und B verwenden mindestens eine Platte für die Serveranwendungen, die jeweils verwendet werden sollen. Es gibt auch eine andere gemeinsam genutzte Platte, die ein redundantes Array unabhängiger Platten ( *RAID* ) Stufe 1 für die ausschließliche Verwendung von MSCS sein muss. Dies wird auch als *Quorum* -Platte bezeichnet. MSCS überwacht beide Computer, um zu überprüfen, ob die Hardware und die Software ordnungsgemäß ausgeführt werden.

In einer einfachen Konfiguration wie dieser sind auf beiden Computern alle Anwendungen installiert, aber nur Computer A läuft mit Live-Anwendungen; Computer B läuft einfach nur laufen und warten. Wenn Computer A einen beliebigen Bereich von Problemen feststellt, beendet MSCS die unterbrechungsfreie Anwendung ordnungsgemäß, überträgt die zugehörigen Statusdaten auf den anderen Computer und leitet die Anwendung dort erneut ein. Dies wird als *Funktionsübernahme* bezeichnet. Anwendungen können *clusterbewusst* ausgeführt werden, damit sie vollständig mit MSCS und Failover zusammenarbeiten.

Abbildung 71 auf Seite 528 zeigt eine typische Konfiguration für einen Cluster mit zwei Computern.

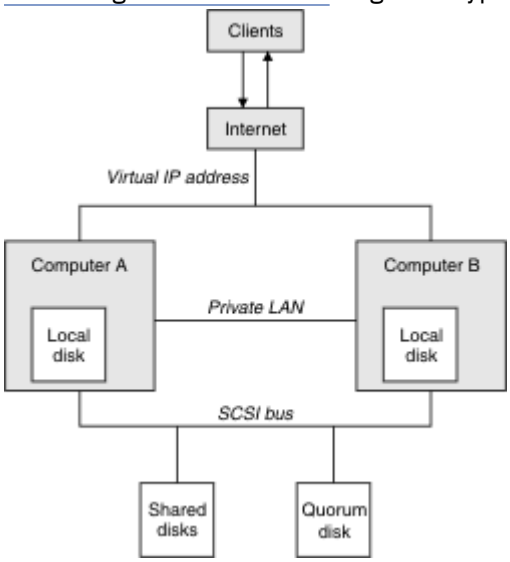



*Abbildung 71. MSCS-Cluster mit zwei Computern*

Jeder Computer kann unter der Steuerung von MSCS auf die gemeinsam genutzte Platte zugreifen, aber nur jeweils einen Computer. Im Falle einer Funktionsübernahme schaltet MSCS den Zugriff auf den anderen Computer um. Die gemeinsam genutzte Platte selbst ist in der Regel eine RAID-Platte, muss aber nicht vorhanden sein.

Jeder Computer ist mit dem externen Netz für den Clientzugriff verbunden, und jeder Computer verfügt über eine IP-Adresse. Ein externer Client, der mit diesem Cluster kommuniziert, kennt jedoch nur eine *virtuelle IP-Adresse* und MSCS leitet den IP-Datenverkehr im Cluster entsprechend weiter.

MSCS führt außerdem eine eigene Kommunikation zwischen den beiden Computern aus, entweder über eine oder mehrere private Verbindungen oder über das öffentliche Netz, z. B. um ihre Status mit dem Überwachungssignal zu überwachen und ihre Datenbanken zu synchronisieren.

## Windows *IBM MQ für MSCS-Clustering einrichten*

Zur Konfiguration von IBM MQ für Clustering machen Sie den Warteschlangenmanager zur Einheit für die Funktionsübernahme für MSCS. Sie definieren einen Warteschlangenmanager als eine Ressource für MSCS, der sie dann überwachen kann, und sie an einen anderen Computer im Cluster übertragen kann, wenn ein Problem auftritt.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Um Ihr System hierfür einzurichten, installieren Sie zunächst IBM MQ auf jedem Computer im Cluster.

Da der Warteschlangenmanager dem IBM MQ-Installationsnamen zugeordnet ist, sollte der IBM MQ-Installationsname auf allen Computern im Cluster identisch sein. Weitere Informationen finden Sie im Abschnitt Installieren und deinstallieren .

Die WS-Manager selbst müssen nur auf dem Computer, auf dem sie erstellt werden, vorhanden sein. Im Falle einer Funktionsübernahme leitet der MSCS die Warteschlangenmanager auf dem anderen Computer ein. Die WS-Manager müssen jedoch ihre Protokoll-und Datendateien auf einer gemeinsam genutzten Clusterplatte und nicht auf einem lokalen Laufwerk haben. Wenn bereits ein Warteschlangenmanager auf einem lokalen Laufwerk installiert ist, können Sie ihn mit einem Tool migrieren, das mit IBM MQbereitge-

stellt wird. Weitere Informationen hierzu finden Sie im Abschnitt „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531. Wenn Sie neue Warteschlangenmanager für die Verwendung mit MSCS erstellen möchten, beachten Sie hierzu den Abschnitt „Warteschlangenmanager für die Verwendung mit MSCS erstellen” auf Seite 530.

Verwenden Sie nach der Installation und Migration den MSCS-Clusteradministrator, um MSCS auf Ihre Warteschlangenmanager hinzuweisen. Weitere Informationen finden Sie im Abschnitt „WS-Manager unter MSCS-Steuerung einschalten” auf Seite 533.

Wenn Sie einen Warteschlangenmanager aus der MSCS-Steuerung entfernen möchten, verwenden Sie das in „Entfernen eines Warteschlangenmanagers aus MSCS-Steuerung” auf Seite 540 beschriebene Verfahren.

### Windows *Konfigurationssymmetrie und MSCS*

Wenn eine Anwendung von einem Knoten zum anderen wechselt, muss er sich in der gleichen Weise verhalten, unabhängig vom Knoten. Der beste Weg, dies zu gewährleisten, ist, die Umgebungen identisch zu machen.

Wenn Sie können, können Sie einen Cluster mit identischer Hardware, Betriebssystemsoftware, Produktsoftware und Konfiguration auf jedem Computer konfigurieren. Stellen Sie insbesondere sicher, dass die gesamte erforderliche Software auf den beiden Computern in Bezug auf Version, Wartungsstufe, SupportPacs, Pfade und Exits identisch ist und dass ein allgemeiner Namensbereich (Sicherheitsumgebung) wie in „MSCS-Sicherheit” auf Seite 529 beschrieben vorhanden ist.

### Windows *MSCS-Sicherheit*

Für eine erfolgreiche MSCS-Sicherheit befolgen Sie die folgenden Richtlinien.

Die Richtlinien lauten wie folgt:

- Stellen Sie sicher, dass Sie identische Softwareinstallationen auf den einzelnen Computern im Cluster haben.
- Erstellen Sie einen allgemeinen Namensbereich (Sicherheitsumgebung) im gesamten Cluster.
- Machen Sie die Knoten der MSCS-Clustermitglieder einer Domäne, in der das Benutzerkonto, das der *Clustereigner* ist, ein Domänenaccount ist.
- Erstellen Sie die anderen Benutzerkonten im Cluster auch als Domänenkonten, so dass sie auf beiden Knoten verfügbar sind. Dies ist automatisch der Fall, wenn Sie bereits über eine Domäne verfügen und die für IBM MQ relevanten Accounts Domänenkonten sind. Wenn Sie zurzeit noch keine Domäne haben, können Sie eine *Minidomäne* einrichten, um die Clusterknoten und die relevanten Accounts zu berücksichtigen. Ihr Ziel ist es, Ihren Cluster von zwei Computern wie einer einzigen Rechenressource zu gestalten.

  Denken Sie daran, dass ein Konto, das auf einem Computer lokal ist, nicht auf dem anderen Computer vorhanden ist. Selbst wenn Sie ein Konto mit demselben Namen auf dem anderen Computer erstellen, ist die zugehörige Sicherheits-ID (SID) unterschiedlich. Wenn Ihre Anwendung also auf den anderen Knoten verschoben wird, sind die Berechtigungen auf diesem Knoten nicht vorhanden.

Bei einem Failover oder einer Bewegung stellt die MSCS-Unterstützung von IBM MQ sicher, dass alle Dateien, die Warteschlangenmanagerobjekte enthalten, über entsprechende Berechtigungen auf dem Zielknoten verfügen. Explizit überprüft der Code, ob die Gruppen "Administratoren" und "mqm" und der Account "SYSTEM" die vollständige Kontrolle haben, und dass, wenn `Everyone` den Lesezugriff auf den alten Knoten hätte, diese Berechtigung auf dem Zielknoten hinzugefügt wird.

Sie können einen Domänenaccount verwenden, um Ihren IBM MQ-Service auszuführen. Stellen Sie sicher, dass sie in der lokalen mqm-Gruppe auf jedem Computer im Cluster vorhanden ist.

### Windows *Mehrere WS-Manager mit MSCS verwenden*

Wenn Sie mehr als einen Warteschlangenmanager auf einem Computer ausführen, können Sie eine dieser Konfigurationen auswählen.

Die Setups sind wie folgt:

- Alle WS-Manager in einer einzigen Gruppe. Wenn in dieser Konfiguration ein Problem mit einem WS-Manager auftritt, werden alle Warteschlangenmanager in der Gruppenübernahme auf den anderen Computer als Gruppe überstellt.
- Ein einzelner WS-Manager in jeder Gruppe. Wenn in dieser Konfiguration ein Problem mit dem Warteschlangenmanager auftritt, scheitert es allein an dem anderen Computer, ohne die anderen Warteschlangenmanager zu beeinträchtigen.
- Eine Mischung aus den ersten beiden Setups.

Windows *Clustermodi und MSCS*

Es gibt zwei Modi, in denen Sie ein Clustersystem mit IBM MQ unter Windows ausführen können: Aktiv/Passiv oder Aktiv/Aktiv.

**Anmerkung:** Wenn Sie MSCS zusammen mit dem Microsoft Transaction Server (COM+) verwenden, können Sie den Modus 'Aktiv/Aktiv' nicht verwenden.

## Aktiv/Passiver Modus

Im Modus 'Aktiv/Passiv' verfügt Computer A über die aktive Anwendung und Computer B als Sicherung, die nur verwendet wird, wenn MSCS ein Problem feststellt.

Sie können diesen Modus nur mit einer gemeinsam genutzten Platte verwenden, aber wenn eine Anwendung eine Funktionsübernahme (Failover) bewirkt, müssen **alle** Anwendungen als Gruppe übertragen werden (da nur ein einziger Computer gleichzeitig auf die gemeinsam genutzte Platte zugreifen kann).

Sie können MSCS mit A als *bevorzugten* Computer konfigurieren. Wenn dann der Computer A repariert oder ausgetauscht wurde und wieder ordnungsgemäß funktioniert, erkennt MSCS dies und schaltet die Anwendung automatisch wieder auf Computer A um.

Wenn Sie mehr als einen Warteschlangenmanager ausführen, sollten Sie eine separate gemeinsam genutzte Platte für die einzelnen WS-Manager verwenden. Anschließend wird jeder WS-Manager in eine separate Gruppe in MSCS gestellt. Auf diese Weise kann jeder WS-Manager eine Funktionsübernahme durch den anderen Computer ohne Auswirkungen auf die anderen WS-Manager erreichen.

## Aktiv/Aktiver Modus

Im Modus Aktiv/Aktiv verfügen Computer A und B beide über aktive Anwendungen, und die Gruppen auf jedem Computer sind so konfiguriert, dass sie den anderen Computer als Sicherung verwenden. Wenn auf Computer A ein Fehler festgestellt wird, überträgt MSCS die Zustandsdaten an Computer B und leitet die Anwendung dort erneut ein. Computer B führt dann seine eigene Anwendung und A's aus.

Für diese Konfiguration benötigen Sie mindestens zwei gemeinsam genutzte Platten. Sie können MSCS mit A als bevorzugten Computer für A-Anwendungen konfigurieren und B als bevorzugten Computer für B-Anwendungen. Nach dem Failover und der Reparatur wird jede Anwendung automatisch wieder auf ihrem eigenen Computer ausgeführt.

Für IBM MQ bedeutet dies, dass Sie z. B. zwei Warteschlangenmanager ausführen können, jeweils einen auf A und B, wobei jeder die volle Leistung des eigenen Computers nutzt. Nach einem Fehler auf dem Computer A werden beide Warteschlangenmanager auf Computer B ausgeführt. Dies bedeutet, die Leistung des einen Computers zu teilen, mit einer verringerten Fähigkeit, große Datenmengen mit Geschwindigkeit zu verarbeiten. Ihre kritischen Anwendungen stehen jedoch weiterhin zur Verfügung, während Sie den Fehler auf A finden und reparieren.

## Windows *Warteschlangenmanager für die Verwendung mit MSCS erstellen*

Diese Prozedur stellt sicher, dass ein neuer Warteschlangenmanager so erstellt wird, dass er für die Vorbereitung und Platzierung unter Microsoft Cluster Service (MSCS) -Steuerung geeignet ist.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Sie beginnen, indem Sie den Warteschlangenmanager mit allen zugehörigen Ressourcen auf einem lokalen Laufwerk erstellen und anschließend die Protokolldateien und Datendateien auf eine gemeinsam genutzte Platte migrieren. (Sie können diese Operation umkehren.) Versuchen Sie **nicht** , einen WS-Manager mit seinen Ressourcen auf einem gemeinsam genutzten Laufwerk zu erstellen.

Sie können einen Warteschlangenmanager zur Verwendung mit MSCS auf zwei Arten erstellen, entweder über eine Eingabeaufforderung oder in IBM MQ Explorer. Der Vorteil bei der Verwendung einer Eingabeaufforderung besteht darin, dass der Warteschlangenmanager *gestoppt* erstellt und auf *manueller Start* gesetzt ist, der für MSCS bereit ist. (Der IBM MQ Explorer startet automatisch einen neuen Warteschlangenmanager und legt ihn nach der Erstellung auf den automatischen Start fest. Sie müssen das ändern.)

## WS-Manager über eine Eingabeaufforderung erstellen

Führen Sie die folgenden Schritte aus, um einen WS-Manager über eine Eingabeaufforderung zu erstellen, der für MSCS verwendet werden soll:

1. Stellen Sie sicher, dass die Umgebungsvariable `MQSPREFIX` auf ein lokales Laufwerk (z. B. `C:\IBM MQ`) gesetzt ist. Wenn Sie dies ändern, führen Sie einen Warmstart der Maschine durch, damit das Systemkonto die Änderung aufnimmt. Wenn Sie die Variable nicht definieren, wird der Warteschlangenmanager im Standardverzeichnis von IBM MQ für Warteschlangenmanager erstellt.
2. Erstellen Sie den WS-Manager mit dem Befehl **crtmqm** . Wenn Sie beispielsweise einen WS-Manager mit dem Namen `mscs_test` im Standardverzeichnis erstellen möchten, verwenden Sie Folgendes:

   ```
   crtmqm mscs_test
   ```

3. Fahren Sie mit dem Abschnitt „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531 fort.

## Warteschlangenmanager mit dem IBM MQ Explorer erstellen

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager unter Verwendung des IBM MQ Explorers zu erstellen, der mit MSCS verwendet werden soll:

1. Starten Sie den IBM MQ Explorer über das Startmenü.
2. Erweitern Sie in der Navigatoransicht die Baumknoten, um den Baumknoten `Warteschlangenmanager` zu suchen.
3. Klicken Sie mit der rechten Maustaste den Baumknoten `WS-Manager` an, und wählen Sie **Neu** > **Warteschlangenmanager** aus. Das Fenster 'Create Queue Manager' (Warteschlangenmanager erstellen) wird angezeigt
4. Füllen Sie den Dialog aus (Schritt 1) und klicken Sie dann auf **Weiter >**.
5. Füllen Sie den Dialog aus (Schritt 2) und klicken Sie dann auf **Weiter >**.
6. Füllen Sie den Dialog (Schritt 3) aus, und stellen Sie sicher, dass `Start Queue Manager` und `Create Server Connection Channel` nicht ausgewählt sind. Klicken Sie anschließend auf **Weiter >**.
7. Füllen Sie den Dialog aus (Schritt 4) und klicken Sie dann auf **Fertig stellen** .
8. Fahren Sie mit dem Abschnitt „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531 fort.

## Windows *Warteschlangenmanager in MSCS-Speicher versetzen*

Mit dieser Prozedur wird ein vorhandener Warteschlangenmanager so konfiguriert, dass er für die Steuerung durch Microsoft Cluster Service (MSCS) geeignet ist.

Um dies zu erreichen, verschieben Sie die Protokolldateien und Datendateien auf gemeinsam genutzte Platten, um sie im Falle eines Fehlers für den anderen Computer verfügbar zu machen. Der vorhandene Warteschlangenmanager kann z. B. Pfade wie `C:\WebSphere MQ\log\`*`QMname`* und `C:\WebSphere MQ\qmgrs\`*`QMname`* haben.

**Achtung:** Versuchen Sie nicht, die Dateien manuell zu verschieben. Verwenden Sie das Dienstprogramm, das als Teil von IBM MQ MSCS Support bereitgestellt wird, wie in diesem Thema beschrieben.

Wenn der Warteschlangenmanager, der versetzt wird, TLS-Verbindungen verwendet und sich das TLS-Schlüsselrepository im Datenverzeichnis des Warteschlangenmanagers auf der lokalen Maschine befindet, wird das Schlüsselrepository mit dem Rest des Warteschlangenmanagers auf die gemeinsam genutzte Platte verschoben. Standardmäßig wird das Warteschlangenmanagerattribut, das die Position des TLS-Schlüsselrepositorys (SSLKEYR) angibt, auf *MQ_INSTALLATION_PATH*\qmgrs\QMGRNAME\ssl\keygesetzt, das sich unter dem Datenverzeichnis des Warteschlangenmanagers befindet. *MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist. Mit dem Befehl hamvmqm wird dieses WS-Manager-Attribut nicht geändert. In dieser Situation müssen Sie das WS-Managerattribut SSLKEYR mit dem IBM MQ Explorer oder mit dem MQSC-Befehl ALTER QMGR ändern, um auf die neue TLS-Schlüsselrepositorydatei zu verweisen.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Das Verfahren ist wie folgt:

1. Fahren Sie den Warteschlangenmanager herunter, und prüfen Sie, ob keine Fehler aufgetreten sind.
2. Wenn die Protokolldateien oder die Warteschlangendateien des Warteschlangenmanagers bereits auf einer gemeinsam genutzten Platte gespeichert sind, überspringen Sie den Rest dieser Prozedur und fahren Sie direkt mit „WS-Manager unter MSCS-Steuerung einschalten” auf Seite 533 fort.
3. Führen Sie ein vollständiges Backup aller Warteschlangen- und Protokolldateien durch und speichern Sie diese Sicherheitskopie in einem sicheren Bereich (in Abschnitt „Warteschlangenmanagerprotokolldateien” auf Seite 543 wird erläutert, warum dies wichtig ist).
4. Wenn Sie bereits über eine geeignete gemeinsam genutzte Plattenressource verfügen, fahren Sie mit Schritt 6 fort. Andernfalls verwenden Sie den MSCS-Cluster-Administrator, um eine Ressource des Typs *gemeinsam genutzte Platte* mit einer ausreichenden Kapazität zum Speichern der WS-Manager-Protokolldateien und der Daten (Warteschlangen-) Dateien zu erstellen.
5. Testen Sie die gemeinsam genutzte Platte, indem Sie den MSCS-Cluster-Administrator verwenden, um ihn von einem Clusterknoten in den anderen und wieder zurück zu versetzen.
6. Stellen Sie sicher, dass die gemeinsam genutzte Platte online auf dem Clusterknoten ist, auf dem das WS-Manager-Protokoll und die Datendateien lokal gespeichert werden.
7. Führen Sie das Dienstprogramm aus, um den WS-Manager wie folgt zu verschieben:

   ```
   hamvmqm /m qmname /dd " e: \
   IBM MQ " /ld " e: \
   IBM MQ \log"
   ```

   Ersetzen Sie *qmname* durch den Namen Ihres Warteschlangenmanagers, den Buchstaben des gemeinsam genutzten Plattenlaufwerks für *e* und das von Ihnen ausgewählte Verzeichnis für *IBM MQ*. Die Verzeichnisse werden erstellt, wenn sie noch nicht vorhanden sind.
8. Testen Sie den Warteschlangenmanager, um sicherzustellen, dass er funktioniert. Verwenden Sie dazu den IBM MQ Explorer. For example:
   a. Klicken Sie auf den Knoten des Warteschlangenmanagers, und wählen Sie dann **Starten** aus. Der WS-Manager wird gestartet.
   b. Klicken Sie mit der rechten Maustaste auf den Baumknoten Warteschlangen und wählen Sie dann **Neu** > **Lokale Warteschlange ...**aus. und geben Sie der Warteschlange einen Namen.
   c. Klicken Sie auf **Fertigstellen**.
   d. Klicken Sie auf die Warteschlange und wählen Sie dann **Testnachricht einreihen ...** aus. Die Anzeige 'Testnachricht einreihen' wird angezeigt.
   e. Geben Sie einen Nachrichtentext ein, klicken Sie dann auf **Testnachricht einreihen** , und schließen Sie die Anzeige.

f. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten durchsuchen ...** aus. Die Anzeige 'Nachrichten-Browser' wird angezeigt.

g. Stellen Sie sicher, dass sich Ihre Nachricht in der Warteschlange befindet, und klicken Sie anschließend auf **Schließen** . Die Anzeige 'Nachrichten-Browser' wird geschlossen.

h. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten löschen ...** aus. Die Nachrichten in der Warteschlange werden gelöscht.

i. Klicken Sie auf die Warteschlange und wählen Sie dann **Löschen ...** aus. Eine Bestätigungsanzeige wird angezeigt. Klicken Sie auf **OK** . Die Warteschlange wird gelöscht.

j. Klicken Sie auf den Knoten des Warteschlangenmanagers, und wählen Sie dann **Stoppen ...** aus. Die Anzeige 'End Queue Manager' wird angezeigt.

k. Klicken Sie auf **OK**. Der Warteschlangenmanager wird gestoppt.

9. Stellen Sie als IBM MQ-Administrator sicher, dass das Startattribut des Warteschlangenmanagers auf "Manuell" gesetzt ist. Geben Sie in der IBM MQ Explorer in der Eigenschaftenanzeige des Warteschlangenmanagers das Startfeld auf `manual` ein.
10. Fahren Sie mit dem Abschnitt „WS-Manager unter MSCS-Steuerung einschalten” auf Seite 533 fort.

## Windows *WS-Manager unter MSCS-Steuerung einschalten*

Vorgehensweise zum Platzieren eines Warteschlangenmanagers unter Microsoft Cluster Service -Steuerung (MSCS), einschließlich vorausgesetzter Tasks.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

### Bevor Sie einen Warteschlangenmanager unter MSCS/WSFC-Steuerung stellen

Führen Sie die folgenden Schritte aus, bevor Sie einen Warteschlangenmanager unter MSCS/WSFC-Steuerung stellen:

1. Stellen Sie sicher, dass IBM MQ und die zugehörige MSCS/WSFC-Unterstützung auf beiden Maschinen im Cluster installiert sind und dass die Software auf jedem Computer identisch ist, wie in „IBM MQ für MSCS-Clustering einrichten” auf Seite 528beschrieben.
2. Verwenden Sie das Dienstprogramm **haregtyp**, um IBM MQ als MSCS-Ressourcentyp auf allen Clusterknoten zu registrieren. Siehe „Unterstützung für MSCS-Dienstprogramme” auf Seite 545.
3. Falls noch nicht geschehen, erstellen Sie einen Warteschlangenmanager zur Verwendung mit MSCS/WSFC.
4. Wenn Sie den Warteschlangenmanager erstellt haben oder er bereits vorhanden ist, stellen Sie sicher, dass Sie die Prozedur in „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531 ausgeführt haben.
5. Wenn der Warteschlangenmanager aktiv ist, stoppen Sie ihn über eine Eingabeaufforderung oder den IBM MQ Explorer.
6. Testen Sie den MSCS/WSFC-Betrieb der gemeinsam genutzten Laufwerke, bevor Sie mit einer der folgenden Windows -Prozeduren in diesem Abschnitt fortfahren.

### Windows Server 2012, 2016, 2019 oder 2022

Gehen Sie wie folgt vor, um einen Warteschlangenmanager unter Windows Server 2012 oder höher unter MSCS/WSFC-Steuerung zu stellen:

1. Melden Sie sich am Cluster-Knotencomputer an, auf dem sich der Warteschlangenmanager befindet, oder melden Sie sich als Benutzer mit Clusterverwaltungsberechtigungen an einer fernen Workstation an und stellen Sie eine Verbindung zu dem Clusterknoten her, der als Host für den Warteschlangenmanager fungiert.
2. Starten Sie das Tool Failover Cluster Management.

3. Klicken Sie mit der rechten Maustaste auf **Failover Cluster Management > Connect Cluster ...** . , um eine Verbindung zum Cluster zu öffnen.
4. Im Gegensatz zu dem Gruppenschema, das im MSCS-Clusteradministrator in früheren Versionen von Windows verwendet wird, verwendet das Tool "Failover Cluster Management" das Konzept von Services und Anwendungen. Ein konfigurierter Service oder eine konfigurierte Anwendung enthält alle Ressourcen, die erforderlich sind, damit eine Anwendung in einem Cluster zusammengefasst wird. Sie können einen Warteschlangenmanager unter WSFC wie folgt konfigurieren:
   a. Klicken Sie auf den Cluster, und wählen Sie **Rolle konfigurieren** aus, um den Konfigurationsassistenten zu starten.
   b. Wählen Sie in der Anzeige "Service oder Anwendung auswählen" die Option **Anderer Server** aus.
   c. Wählen Sie eine geeignete IP-Adresse als Clientzugriffspunkt aus.

      Bei dieser Adresse muss es sich um eine nicht verwendete IP-Adresse handeln, die von Clients und anderen Warteschlangenmanagern verwendet werden soll, um eine Verbindung zum *virtuellen* Warteschlangenmanager herzustellen. Bei dieser IP-Adresse handelt es sich nicht um die normale (statische) Adresse eines der beiden Knoten. Es handelt sich um eine zusätzliche Adresse, die zwischen den beiden Knoten *floats* ist. Obwohl WSFC das Routing dieser Adresse handhabt, prüft es **nicht** , ob die Adresse erreichbar ist.
   d. Ordordnen Sie eine Speichereinheit für die exklusive Verwendung durch den Warteschlangenmanager zu. Diese Einheit muss als Ressourceninstanz erstellt werden, bevor sie zugeordnet werden kann.

      Sie können ein Laufwerk verwenden, um sowohl die Protokolle als auch die Warteschlangendateien zu speichern, oder Sie können sie über Laufwerke hinweg aufteilen. Wenn jeder WS-Manager über eine eigene gemeinsam genutzte Platte verfügt, stellen Sie sicher, dass alle Laufwerke, die von diesem Warteschlangenmanager verwendet werden, exklusiv für diesen Warteschlangenmanager sind, d. -E., dass nichts anderes auf den Laufwerken basiert. Stellen Sie außerdem sicher, dass Sie für jedes Laufwerk, das der Warteschlangenmanager verwendet, eine Ressourceninstanz erstellen.

      Der Ressourcentyp für ein Laufwerk hängt von der von Ihnen verwendeten SCSI-Unterstützung ab. Weitere Informationen finden Sie in den Anweisungen zum SCSI-Adapter. Es können bereits Gruppen und Ressourcen für die einzelnen gemeinsam genutzten Laufwerke vorhanden sein. Ist dies der Fall, müssen Sie die Ressourceninstanz für jedes Laufwerk nicht erstellen. Verschieben Sie ihn von seiner aktuellen Gruppe in die für den Warteschlangenmanager erstellte Gruppe.

      Geben Sie für jede Laufwerkressource die möglichen Eigner beider Knoten an. Setzen Sie abhängige Ressourcen auf "none".
   e. Wählen Sie die Ressource **MQSeries MSCS** in der Anzeige "Ressourcentyp auswählen" aus.
   f. Führen Sie die verbleibenden Schritte im Assistenten aus.
5. Bevor Sie die Ressource online setzen, benötigt die MSCS-Ressource von MQSeries zusätzliche Konfigurationsanforderungen:
   a. Wählen Sie den neu definierten Service aus, der eine Ressource mit dem Namen 'New MQSeries MSCS' enthält.
   b. Klicken Sie in der MQ-Ressource auf **Eigenschaften** .
   c. Konfigurieren Sie die Ressource:
      - `Name` ; Wählen Sie einen Namen aus, der die Identifizierung des Warteschlangenmanagers, für den er ausgeführt wird, erleichtert.
      - `Run in a separate Resource Monitor` ; für eine bessere Isolation
      - `Possible owners` ; beide Knoten festlegen
      - `Dependencies` ; Fügen Sie das Laufwerk und die IP-Adresse für diesen Warteschlangenmanager hinzu.

        **Warnung:** Wenn Sie diese Abhängigkeiten nicht hinzufügen, bedeutet dies, dass IBM MQ versucht, den Warteschlangenmanagerstatus während der Failover auf die falsche Clusterplatte zu

schreiben. Da viele Prozesse möglicherweise gleichzeitig versuchen, auf diese Platte zu schreiben, könnten einige IBM MQ-Prozesse von der Ausführung blockiert werden.

- `Parameters` ; wie folgt:
  - `QueueManagerName` (erforderlich); der Name des Warteschlangenmanagers, den diese Ressource steuern soll. Dieser WS-Manager muss auf dem lokalen Computer vorhanden sein.
  - `PostOnlineCommand` (optional); Sie können ein Programm angeben, das ausgeführt werden soll, wenn die WS-Manager-Ressource ihren Status von offline in online ändert. Nähere Informationen finden Sie unter „PostOnlineCommand und PreOfflineCommand in MSCS" auf Seite 544.
  - `PreOfflineCommand` (optional); Sie können ein Programm angeben, das ausgeführt werden soll, wenn die Warteschlangenmanagerressource ihren Status von "Online" in "Offline" ändert. Nähere Informationen finden Sie unter „PostOnlineCommand und PreOfflineCommand in MSCS" auf Seite 544.

    **Anmerkung:** Das Abfrageintervall für *LooksAlive* wird auf den Standardwert 5000 ms gesetzt. Das Abfrageintervall für *isAlive* wird auf den Standardwert 60000 ms gesetzt. Diese Standardwerte können nur geändert werden, nachdem die Ressourcendefinition abgeschlossen wurde. Weitere Informationen finden Sie im Abschnitt „looksAlive und isAlive-Polling für MSCS" auf Seite 540.

d. Legen Sie optional einen bevorzugten Knoten fest (aber beachten Sie die Kommentare in „Bevorzugte Knoten in MSCS verwenden" auf Seite 544).

e. Die *Funktionsübernahmerichtlinie* wird standardmäßig auf sinnvolle Werte gesetzt, Sie können jedoch die Schwellenwerte und Zeiträume, die *Resource Failover* und *Group Failover* steuern, so optimieren, dass sie mit den auf den Warteschlangenmanager platzierten Lasten übereinstimmen.

6. Testen Sie den Warteschlangenmanager, indem Sie ihn online in den MSCS-Clusteradministrator setzen und einer Testworkload unterziehen. Wenn Sie mit einem Testwarteschlangenmanager experimentieren, verwenden Sie den IBM MQ-Explorer. For example:

   a. Klicken Sie mit der rechten Maustaste auf den Baumknoten `Warteschlangen` und wählen Sie dann **Neu** > **Lokale Warteschlange ...**aus. und geben Sie der Warteschlange einen Namen.

   b. Klicken Sie auf **Fertigstellen**. Die Warteschlange wird erstellt und in der Inhaltsansicht angezeigt.

   c. Klicken Sie auf die Warteschlange und wählen Sie dann **Testnachricht einreihen ...** aus. Die Anzeige 'Testnachricht einreihen' wird angezeigt.

   d. Geben Sie einen Nachrichtentext ein, klicken Sie dann auf **Testnachricht einreihen** , und schließen Sie die Anzeige.

   e. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten durchsuchen ...** aus. Die Anzeige 'Nachrichten-Browser' wird angezeigt.

   f. Stellen Sie sicher, dass sich Ihre Nachricht in der Warteschlange befindet, und klicken Sie anschließend auf **Schließen** . Die Anzeige 'Nachrichten-Browser' wird geschlossen.

   g. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten löschen ...** aus. Die Nachrichten in der Warteschlange werden gelöscht.

   h. Klicken Sie auf die Warteschlange und wählen Sie dann **Löschen ...** aus. Eine Bestätigungsanzeige wird angezeigt. Klicken Sie auf **OK** . Die Warteschlange wird gelöscht.

7. Testen Sie, ob der Warteschlangenmanager offline und wieder online mit dem MSCS-Cluster-Administrator ausgeführt werden kann.

8. Simulieren Sie eine Funktionsübernahme.

   Klicken Sie im MSCS-Clusteradministrator auf die Gruppe, die den Warteschlangenmanager enthält, und wählen Sie `Move Group` aus. Dies kann einige Minuten dauern. (Wenn Sie einen Warteschlangenmanager zu einem anderen Zeitpunkt schnell in einen anderen Knoten verschieben möchten, führen Sie die Prozedur in „Warteschlangenmanager in MSCS-Speicher versetzen" auf Seite 531 aus.) Sie können auch mit der rechten Maustaste klicken und `Initiate Failure` auswählen. Die Aktion (lokaler Neustart oder Failover) hängt vom aktuellen Status und den Konfigurationseinstellungen ab.

## Windows Server 2008

Gehen Sie wie folgt vor, um einen Warteschlangenmanager unter MSCS-Steuerung auf Windows Server 2008 zu stellen:

1. Melden Sie sich am Cluster-Knotencomputer an, auf dem sich der Warteschlangenmanager befindet, oder melden Sie sich als Benutzer mit Clusterverwaltungsberechtigungen an einer fernen Workstation an und stellen Sie eine Verbindung zu dem Clusterknoten her, der als Host für den Warteschlangenmanager fungiert.
2. Starten Sie das Tool Failover Cluster Management.
3. Klicken Sie mit der rechten Maustaste auf **Failover-Cluster-Management > Cluster verwalten** . , um eine Verbindung zum Cluster zu öffnen.
4. Im Gegensatz zu dem Gruppenschema, das im MSCS-Clusteradministrator in früheren Versionen von Windows verwendet wird, verwendet das Tool "Failover Cluster Management" das Konzept von Services und Anwendungen. Ein konfigurierter Service oder eine konfigurierte Anwendung enthält alle Ressourcen, die erforderlich sind, damit eine Anwendung in einem Cluster zusammengefasst wird. Sie können einen WS-Manager wie folgt unter MSCS konfigurieren:
   a. Klicken Sie mit der rechten Maustaste auf **Dienste und Anwendungen > Service oder Anwendung konfigurieren ...** . , um den Konfigurationsassistenten zu starten.
   b. Wählen Sie **Anderer Server** in der Anzeige **Service oder Anwendung auswählen** aus.
   c. Wählen Sie eine geeignete IP-Adresse als Clientzugriffspunkt aus.

      Bei dieser Adresse muss es sich um eine nicht verwendete IP-Adresse handeln, die von Clients und anderen Warteschlangenmanagern verwendet werden soll, um eine Verbindung zum *virtuellen* Warteschlangenmanager herzustellen. Bei dieser IP-Adresse handelt es sich nicht um die normale (statische) Adresse eines der beiden Knoten. Es handelt sich um eine zusätzliche Adresse, die zwischen den beiden Knoten *floats* ist. Obwohl MSCS das Routing dieser Adresse handhabt, prüft es **nicht** , ob die Adresse erreicht werden kann.
   d. Ordordnen Sie eine Speichereinheit für die exklusive Verwendung durch den Warteschlangenmanager zu. Diese Einheit muss als Ressourceninstanz erstellt werden, bevor sie zugeordnet werden kann.

      Sie können ein Laufwerk verwenden, um sowohl die Protokolle als auch die Warteschlangendateien zu speichern, oder Sie können sie über Laufwerke hinweg aufteilen. Wenn jeder WS-Manager über eine eigene gemeinsam genutzte Platte verfügt, stellen Sie sicher, dass alle Laufwerke, die von diesem Warteschlangenmanager verwendet werden, exklusiv für diesen Warteschlangenmanager sind, d. -E., dass nichts anderes auf den Laufwerken basiert. Stellen Sie außerdem sicher, dass Sie für jedes Laufwerk, das der Warteschlangenmanager verwendet, eine Ressourceninstanz erstellen.

      Der Ressourcentyp für ein Laufwerk hängt von der von Ihnen verwendeten SCSI-Unterstützung ab. Weitere Informationen finden Sie in den Anweisungen zum SCSI-Adapter. Es können bereits Gruppen und Ressourcen für die einzelnen gemeinsam genutzten Laufwerke vorhanden sein. Ist dies der Fall, müssen Sie die Ressourceninstanz für jedes Laufwerk nicht erstellen. Verschieben Sie ihn von seiner aktuellen Gruppe in die für den Warteschlangenmanager erstellte Gruppe.

      Geben Sie für jede Laufwerkressource die möglichen Eigner beider Knoten an. Setzen Sie abhängige Ressourcen auf "none".
   e. Wählen Sie die Ressource **MQSeries MSCS** in der Anzeige **Ressourcentyp auswählen** aus.
   f. Führen Sie die verbleibenden Schritte im Assistenten aus.
5. Bevor Sie die Ressource online setzen, benötigt die MSCS-Ressource von MQSeries zusätzliche Konfigurationsanforderungen:
   a. Wählen Sie den neu definierten Service aus, der eine Ressource mit dem Namen 'New MQSeries MSCS' enthält.
   b. Klicken Sie in der MQ-Ressource auf **Eigenschaften** .
   c. Konfigurieren Sie die Ressource:

- `Name` ; Wählen Sie einen Namen aus, der die Identifizierung des Warteschlangenmanagers, für den er ausgeführt wird, erleichtert.
- `Run in a separate Resource Monitor` ; für eine bessere Isolation
- `Possible owners` ; beide Knoten festlegen
- `Dependencies` ; Fügen Sie das Laufwerk und die IP-Adresse für diesen Warteschlangenmanager hinzu.

  **Warnung:** Wenn Sie diese Abhängigkeiten nicht hinzufügen, bedeutet dies, dass IBM MQ versucht, den Warteschlangenmanagerstatus während der Failover auf die falsche Clusterplatte zu schreiben. Da viele Prozesse möglicherweise gleichzeitig versuchen, auf diese Platte zu schreiben, könnten einige IBM MQ-Prozesse von der Ausführung blockiert werden.
- `Parameters` ; wie folgt:
  - `QueueManagerName` (erforderlich); der Name des Warteschlangenmanagers, den diese Ressource steuern soll. Dieser WS-Manager muss auf dem lokalen Computer vorhanden sein.
  - `PostOnlineCommand` (optional); Sie können ein Programm angeben, das ausgeführt werden soll, wenn die WS-Manager-Ressource ihren Status von offline in online ändert. Nähere Informationen finden Sie unter „PostOnlineCommand und PreOfflineCommand in MSCS” auf Seite 544.
  - `PreOfflineCommand` (optional); Sie können ein Programm angeben, das ausgeführt werden soll, wenn die Warteschlangenmanagerressource ihren Status von "Online" in "Offline" ändert. Nähere Informationen finden Sie unter „PostOnlineCommand und PreOfflineCommand in MSCS” auf Seite 544.

    **Anmerkung:** Das Abfrageintervall für *LooksAlive* wird auf den Standardwert 5000 ms gesetzt. Das Abfrageintervall für *isAlive* wird auf den Standardwert 60000 ms gesetzt. Diese Standardwerte können nur geändert werden, nachdem die Ressourcendefinition abgeschlossen wurde. Weitere Informationen finden Sie im Abschnitt „looksAlive und isAlive-Polling für MSCS” auf Seite 540.

d. Legen Sie optional einen bevorzugten Knoten fest (aber beachten Sie die Kommentare in „Bevorzugte Knoten in MSCS verwenden” auf Seite 544).

e. Die *Funktionsübernahmerichtlinie* wird standardmäßig auf sinnvolle Werte gesetzt, Sie können jedoch die Schwellenwerte und Zeiträume, die *Resource Failover* und *Group Failover* steuern, so optimieren, dass sie mit den auf den Warteschlangenmanager platzierten Lasten übereinstimmen.

6. Testen Sie den Warteschlangenmanager, indem Sie ihn online in den MSCS-Clusteradministrator setzen und einer Testworkload unterziehen. Wenn Sie mit einem Testwarteschlangenmanager experimentieren, verwenden Sie den IBM MQ-Explorer. For example:

   a. Klicken Sie mit der rechten Maustaste auf den Baumknoten `Warteschlangen` und wählen Sie dann **Neu** > **Lokale Warteschlange ...**aus. und geben Sie der Warteschlange einen Namen.

   b. Klicken Sie auf **Fertigstellen**. Die Warteschlange wird erstellt und in der Inhaltsansicht angezeigt.

   c. Klicken Sie auf die Warteschlange und wählen Sie dann **Testnachricht einreihen ...** aus. Die Anzeige **Testnachricht einreihen** wird angezeigt.

   d. Geben Sie einen Nachrichtentext ein, klicken Sie dann auf **Testnachricht einreihen** , und schließen Sie die Anzeige.

   e. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten durchsuchen ...** aus. Die Anzeige **Nachrichtenbrowser** wird angezeigt.

   f. Stellen Sie sicher, dass sich Ihre Nachricht in der Warteschlange befindet, und klicken Sie anschließend auf **Schließen** . Die Anzeige **Nachrichtenbrowser** wird geschlossen.

   g. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten löschen ...** aus. Die Nachrichten in der Warteschlange werden gelöscht.

   h. Klicken Sie auf die Warteschlange und wählen Sie dann **Löschen ...** aus. Eine Bestätigungsanzeige wird angezeigt. Klicken Sie auf **OK** . Die Warteschlange wird gelöscht.

7. Testen Sie, ob der Warteschlangenmanager offline und wieder online mit dem MSCS-Cluster-Administrator ausgeführt werden kann.
8. Simulieren Sie eine Funktionsübernahme.

   Klicken Sie im MSCS-Clusteradministrator auf die Gruppe, die den Warteschlangenmanager enthält, und wählen Sie `Move Group` aus. Dies kann einige Minuten dauern. (Wenn Sie einen Warteschlangenmanager zu einem anderen Zeitpunkt schnell in einen anderen Knoten verschieben möchten, führen Sie die Prozedur in „Warteschlangenmanager in MSCS-Speicher versetzen" auf Seite 531 aus.) Sie können auch mit der rechten Maustaste klicken und `Initiate Failure` auswählen. Die Aktion (lokaler Neustart oder Failover) hängt vom aktuellen Status und den Konfigurationseinstellungen ab.

## Windows 2003

Gehen Sie wie folgt vor, um einen Warteschlangenmanager unter MSCS-Steuerung auf Windows 2003 zu stellen:

1. Melden Sie sich am Cluster-Knotencomputer an, auf dem sich der Warteschlangenmanager befindet, oder melden Sie sich als Benutzer mit Clusterverwaltungsberechtigungen an einer fernen Workstation an und stellen Sie eine Verbindung zu dem Clusterknoten her, der als Host für den Warteschlangenmanager fungiert.
2. Starten Sie den MSCS-Clusteradministrator.
3. Öffnen Sie eine Verbindung zum Cluster.
4. Erstellen Sie eine MSCS-Gruppe, die verwendet werden soll, um die Ressourcen für den Warteschlangenmanager zu enthalten. Benennen Sie die Gruppe so, dass es ersichtlich ist, auf welche Warteschlangenmanager sie sich bezieht. Jede Gruppe kann mehrere Warteschlangenmanager enthalten, wie in „Mehrere WS-Manager mit MSCS verwenden" auf Seite 529 beschrieben.

   Verwenden Sie die Gruppe für alle verbleibenden Schritte.
5. Erstellen Sie eine Ressourceninstanz für jedes der logischen SCSI-Laufwerke, die vom Warteschlangenmanager verwendet werden.

   Sie können ein Laufwerk verwenden, um sowohl die Protokolle als auch die Warteschlangendateien zu speichern, oder Sie können sie über Laufwerke hinweg aufteilen. Wenn jeder WS-Manager über eine eigene gemeinsam genutzte Platte verfügt, stellen Sie sicher, dass alle Laufwerke, die von diesem Warteschlangenmanager verwendet werden, exklusiv für diesen Warteschlangenmanager sind, d. -E., dass nichts anderes auf den Laufwerken basiert. Stellen Sie außerdem sicher, dass Sie für jedes Laufwerk, das der Warteschlangenmanager verwendet, eine Ressourceninstanz erstellen.

   Der Ressourcentyp für ein Laufwerk hängt von der von Ihnen verwendeten SCSI-Unterstützung ab. Weitere Informationen finden Sie in den Anweisungen zum SCSI-Adapter. Es können bereits Gruppen und Ressourcen für die einzelnen gemeinsam genutzten Laufwerke vorhanden sein. Ist dies der Fall, müssen Sie die Ressourceninstanz für jedes Laufwerk nicht erstellen. Verschieben Sie ihn von seiner aktuellen Gruppe in die für den Warteschlangenmanager erstellte Gruppe.

   Geben Sie für jede Laufwerkressource die möglichen Eigner beider Knoten an. Setzen Sie abhängige Ressourcen auf "none".
6. Erstellen Sie eine Ressourceninstanz für die IP-Adresse.

   Erstellen Sie eine IP-Adressenressource (Ressourcentyp *IP-Adresse* ). Bei dieser Adresse muss es sich um eine nicht verwendete IP-Adresse handeln, die von Clients und anderen Warteschlangenmanagern verwendet werden soll, um eine Verbindung zum *virtuellen* Warteschlangenmanager herzustellen. Bei dieser IP-Adresse handelt es sich nicht um die normale (statische) Adresse eines der beiden Knoten. Es handelt sich um eine zusätzliche Adresse, die zwischen den beiden Knoten *floats* ist. Obwohl MSCS das Routing dieser Adresse handhabt, prüft es **nicht** , ob die Adresse erreicht werden kann.
7. Erstellen Sie eine Ressourceninstanz für den WS-Manager.

   Erstellen Sie eine Ressource vom Typ *IBM MQ MSCS*. Der Assistent fordert Sie zur Eingabe von verschiedenen Elementen auf, einschließlich der folgenden:

- `Name` ; Wählen Sie einen Namen aus, der die Identifizierung des Warteschlangenmanagers, für den er ausgeführt wird, erleichtert.
- `Add to group` ; verwenden Sie die von Ihnen erstellte Gruppe.
- `Run in a separate Resource Monitor` ; für eine bessere Isolation
- `Possible owners` ; beide Knoten festlegen
- `Dependencies` ; Fügen Sie das Laufwerk und die IP-Adresse für diesen Warteschlangenmanager hinzu.

  **Warnung:** Wenn Sie diese Abhängigkeiten nicht hinzufügen, bedeutet dies, dass IBM MQ versucht, den Warteschlangenmanagerstatus während der Failover auf die falsche Clusterplatte zu schreiben. Da viele Prozesse möglicherweise gleichzeitig versuchen, auf diese Platte zu schreiben, könnten einige IBM MQ-Prozesse von der Ausführung blockiert werden.
- `Parameters` ; wie folgt:
  - `QueueManagerName` (erforderlich); der Name des Warteschlangenmanagers, den diese Ressource steuern soll. Dieser WS-Manager muss auf dem lokalen Computer vorhanden sein.
  - `PostOnlineCommand` (optional); Sie können ein Programm angeben, das ausgeführt werden soll, wenn die WS-Manager-Ressource ihren Status von offline in online ändert. Nähere Informationen finden Sie unter „PostOnlineCommand und PreOfflineCommand in MSCS” auf Seite 544.
  - `PreOfflineCommand` (optional); Sie können ein Programm angeben, das ausgeführt werden soll, wenn die Warteschlangenmanagerressource ihren Status von "Online" in "Offline" ändert. Nähere Informationen finden Sie unter „PostOnlineCommand und PreOfflineCommand in MSCS” auf Seite 544.

    **Anmerkung:** Das Abfrageintervall für *LooksAlive* wird auf den Standardwert 5000 ms gesetzt. Das Abfrageintervall für *isAlive* wird auf den Standardwert 30000 ms gesetzt. Diese Standardwerte können nur geändert werden, nachdem die Ressourcendefinition abgeschlossen wurde. Weitere Informationen finden Sie im Abschnitt „looksAlive und isAlive-Polling für MSCS” auf Seite 540.

8. Legen Sie optional einen bevorzugten Knoten fest (aber beachten Sie die Kommentare in „Bevorzugte Knoten in MSCS verwenden” auf Seite 544).
9. Die *Funktionsübernahmerichtlinie* (wie in den Eigenschaften für die Gruppe definiert) wird standardmäßig auf sinnvolle Werte gesetzt, Sie können jedoch die Schwellenwerte und Zeiträume, die *Resource Failover* und *Group Failover* steuern, so optimieren, dass sie mit den auf den Warteschlangenmanager platzierten Lasten übereinstimmen.
10. Testen Sie den Warteschlangenmanager, indem Sie ihn online in den MSCS-Clusteradministrator setzen und einer Testworkload unterziehen. Wenn Sie mit einem Testwarteschlangenmanager experimentieren, verwenden Sie den IBM MQ-Explorer. For example:
    a. Klicken Sie mit der rechten Maustaste auf den Baumknoten `Warteschlangen` und wählen Sie dann **Neu** > **Lokale Warteschlange ...**aus. und geben Sie der Warteschlange einen Namen.
    b. Klicken Sie auf **Fertigstellen**. Die Warteschlange wird erstellt und in der Inhaltsansicht angezeigt.
    c. Klicken Sie auf die Warteschlange und wählen Sie dann **Testnachricht einreihen ...** aus. Die Anzeige **Testnachricht einreihen** wird angezeigt.
    d. Geben Sie einen Nachrichtentext ein, klicken Sie dann auf **Testnachricht einreihen** , und schließen Sie die Anzeige.
    e. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten durchsuchen ...** aus. Die Anzeige **Nachrichtenbrowser** wird angezeigt.
    f. Stellen Sie sicher, dass sich Ihre Nachricht in der Warteschlange befindet, und klicken Sie anschließend auf **Schließen** . Die Anzeige **Nachrichtenbrowser** wird geschlossen.
    g. Klicken Sie auf die Warteschlange, und wählen Sie dann **Nachrichten löschen ...** aus. Die Nachrichten in der Warteschlange werden gelöscht.
    h. Klicken Sie auf die Warteschlange und wählen Sie dann **Löschen ...** aus. Eine Bestätigungsanzeige wird angezeigt. Klicken Sie auf **OK** . Die Warteschlange wird gelöscht.

11. Testen Sie, ob der Warteschlangenmanager offline und wieder online mit dem MSCS-Cluster-Administrator ausgeführt werden kann.
12. Simulieren Sie eine Funktionsübernahme.

    Klicken Sie im MSCS-Clusteradministrator auf die Gruppe, die den Warteschlangenmanager enthält, und wählen Sie `Move Group` aus. Dies kann einige Minuten dauern. (Wenn Sie einen Warteschlangenmanager zu einem anderen Zeitpunkt schnell in einen anderen Knoten verschieben möchten, führen Sie die Prozedur in „Warteschlangenmanager in MSCS-Speicher versetzen" auf Seite 531 aus.) Sie können auch mit der rechten Maustaste klicken und `Initiate Failure` auswählen. Die Aktion (lokaler Neustart oder Failover) hängt vom aktuellen Status und den Konfigurationseinstellungen ab.

## Windows *looksAlive und isAlive-Polling für MSCS*

*looksAlive* und *isAlive* sind Intervalle, in denen Microsoft Cluster Service (MSCS) Rückrufe an den von den Ressourcentypen bereitgestellten Bibliothekscode und Anforderungen, die die Ressource ausführt, durchführt, um den Betriebsstatus von sich selbst zu ermitteln. Dies bestimmt schließlich, ob MSCS über die Ressource fehlschlagen soll.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Bei jedem Ablaufen des *looksAlive* -Intervalls (Standardwert: 5000 ms) wird die WS-Manager-Ressource aufgerufen, um eine eigene Prüfung durchzuführen, um festzustellen, ob der Status zufriedenstellend ist.

Bei jeder Ausführung des *isAlive* -Intervalls (Standardwert: 30000 ms) wird ein weiterer Aufruf an die WS-Manager-Ressource vorgenommen, um eine weitere Überprüfung durchzuführen, um zu ermitteln, ob die Ressource ordnungsgemäß funktioniert. Auf diese Weise können zwei Ebenen der Ressourcentypüberprüfung aktiviert werden

1. Eine *looksAlive* -Statusprüfung, um festzustellen, ob die Ressource funktionsfähig ist.
2. Eine größere *isAlive* Prüfung, die festlegt, ob die WS-Manager-Ressource aktiv ist.

Wenn festgestellt wird, dass die Warteschlangenmanagerressource nicht aktiv ist, löst MSCS basierend auf anderen erweiterten MSCS-Optionen eine Übernahme für die Ressource und zugehörige abhängige Ressourcen auf einem anderen Knoten im Cluster aus. Weitere Informationen finden Sie in der MSCS-Dokumentation .

## Windows *Entfernen eines Warteschlangenmanagers aus MSCS-Steuerung*

Sie können Warteschlangenmanager aus der Microsoft Cluster Service -Steuerung (MSCS) entfernen und an die manuelle Verwaltung zurückgeben.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Sie müssen Warteschlangenmanager nicht von der MSCS-Steuerung für Wartungsoperationen entfernen. Sie können das tun, indem Sie einen Warteschlangenmanager vorübergehend offline schalten, indem Sie den MSCS-Clusteradministrator verwenden. Das Entfernen eines WS-Managers von der MSCS-Steuerung ist eine permanente Änderung; nur dann, wenn Sie entscheiden, dass MSCS keine weitere Steuerung des Warteschlangenmanagers mehr haben soll.

Wenn der WS-Manager, der entfernt wird, TSL-Verbindungen verwendet, muss das WS-Managerattribut SSLKEYR mit dem IBM MQ-Explorer oder mit dem MQSC-Befehl `ALTER QMGR` geändert werden, um auf die TLS-Schlüsselrepositorydatei im lokalen Verzeichnis zu verweisen.

Das Verfahren ist wie folgt:

1. Schalten Sie den Warteschlangenmanager über MSCS Cluster Administrator offline (siehe „WS-Manager offline von MSCS übernehmen" auf Seite 541).
2. Löschen Sie die Ressourceninstanz. Der WS-Manager wird dadurch nicht gelöscht.
3. Optional können Sie die WS-Manager-Dateien von gemeinsam genutzten Laufwerken zurück auf lokale Laufwerke migrieren. Informationen dazu finden Sie unter „WS-Manager aus MSCS-Speicher zurückgeben" auf Seite 541.

4. Testen Sie den WS-Manager.

## WS-Manager offline von MSCS übernehmen

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager offline von MSCS zu nehmen:

1. Starten Sie den MSCS-Clusteradministrator.
2. Öffnen Sie eine Verbindung zum Cluster.
3. Wählen Sie `Groups` oder `Role` aus, wenn Sie Windows 2012 verwenden, und öffnen Sie die Gruppe, die den Warteschlangenmanager enthält, der verschoben werden soll.
4. Wählen Sie die WS-Manager-Ressource aus.
5. Klicken Sie auf die Schaltfläche und wählen Sie `Offline` aus.
6. Warten Sie auf den Abschluss.

## WS-Manager aus MSCS-Speicher zurückgeben

Mit dieser Prozedur wird der Warteschlangenmanager so konfiguriert, dass er auf dem lokalen Laufwerk seines Computers zurückbleibt, d. h., er wird zu einem *normalen* IBM MQ-Warteschlangenmanager. Um dies zu erreichen, verschieben Sie die Protokolldateien und Datendateien von den gemeinsam genutzten Platten. Der vorhandene Warteschlangenmanager kann z. B. Pfade wie `E:\WebSphere MQ\log\`*QMname* und `E:\WebSphere MQ\qmgrs\`*QMname* haben. Versuchen Sie nicht, die Dateien manuell zu verschieben; verwenden Sie das Dienstprogramm **hamvmqm**, das als Teil von IBM MQ MSCS Support bereitgestellt wird:

1. Führen Sie ein vollständiges Backup aller Warteschlangen- und Protokolldateien durch und speichern Sie diese Sicherheitskopie in einem sicheren Bereich (in Abschnitt „Warteschlangenmanagerprotokolldateien” auf Seite 543 wird erläutert, warum dies wichtig ist).
2. Entscheiden Sie, welches lokale Laufwerk verwendet werden soll, und stellen Sie sicher, dass es über ausreichende Kapazität zum Speichern der Protokolldateien und Datendateien (Warteschlangendateien) des Warteschlangenmanagers verfügt.
3. Stellen Sie sicher, dass sich die gemeinsam genutzte Platte, auf der sich die Dateien befinden, online auf dem Clusterknoten befindet, in den das WS-Manager-Protokoll und die Datendateien verschoben werden sollen.
4. Führen Sie das Dienstprogramm aus, um den WS-Manager wie folgt zu verschieben:

   ```
   hamvmqm /m qmname /dd " c:\
   IBM MQ " /ld "c:\
   IBM MQ \log"
   ```

   Ersetzen Sie *qmname* durch Ihren Warteschlangenmanagernamen, *c* durch den lokalen Laufwerkbuchstaben und *IBM MQ* durch das ausgewählte Verzeichnis (die Verzeichnisse werden erstellt, wenn sie noch nicht vorhanden sind).
5. Testen Sie den Warteschlangenmanager, um sicherzustellen, dass er funktioniert (wie in „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531 beschrieben).

## Windows *Hinweise und Tipps zur Verwendung von MSCS*

Dieser Abschnitt enthält einige allgemeine Informationen, die Sie bei der effektiven Verwendung der IBM MQ -Unterstützung für Microsoft Cluster Service (MSCS) unterstützen.

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

Wie lange dauert es, bis ein WS-Manager von einer Maschine zum anderen fehlschlägt? Dies hängt stark von der Auslastung des Warteschlangenmanagers und der Kombination des Datenverkehrs ab, z. B., wie viel von dem Warteschlangenmanager persistent ist, innerhalb des Synchronisationspunkts und wie viel vor dem Fehlschlag festgeschrieben wurde. IBM-Tests haben Failover- und Failback-Zeiten von etwa einer

Minute angegeben. Dies war auf einem sehr gering belasteten WS-Manager, und die tatsächlichen Zeiten werden je nach Ladezeit sehr unterschiedlich sein.

Windows *Sicherstellen, dass MSCS funktioniert*
Führen Sie die folgenden Schritte aus, um sicherzustellen, dass Sie über einen aktiven MSCS-Cluster verfügen.

Bei den Taskbeschreibungen, die mit „Warteschlangenmanager für die Verwendung mit MSCS erstellen” auf Seite 530 beginnen, wird davon ausgegangen, dass Sie über einen aktiven MSCS-Cluster verfügen, in dem Sie Ressourcen erstellen, migrieren und löschen können. Wenn Sie sicherstellen möchten, dass Sie über einen solchen Cluster verfügen:

1. Erstellen Sie mit dem MSCS-Clusteradministrator eine Gruppe.
2. Erstellen Sie in dieser Gruppe eine Instanz einer generischen Anwendungsressource und geben Sie dabei die Systemuhr an (Pfadname `C:\winnt\system32\clock.exe` und Arbeitsverzeichnis von `C:\`).
3. Stellen Sie sicher, dass Sie die Ressource in den Onlinemodus versetzen können, dass Sie die Gruppe, die sie enthält, in den anderen Knoten verschieben können und dass Sie die Ressource offline schalten können.

Windows *Manueller Start und MSCS*
Für einen Warteschlangenmanager, der von MSCS verwaltet wird, müssen Sie das Startattribut auf 'Manuell' setzen. Dadurch wird sichergestellt, dass die MSCS-Unterstützung von IBM MQ den MQSeries-Service ohne sofortigen Start des Warteschlangenmanagers erneut starten kann.

Die MSCS-Unterstützung von IBM MQ muss in der Lage sein, den Service erneut zu starten, damit er die Überwachung und Steuerung ausführen kann. Er muss jedoch selbst die Kontrolle darüber behalten, welche Warteschlangenmanager ausgeführt werden und auf welchen Maschinen. Weitere Informationen finden Sie unter „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531.

Windows *MSCS-und Warteschlangenmanager*
Hinweise zu Warteschlangenmanagern bei der Verwendung von MSCS.

## Erstellen eines übereinstimmenden Warteschlangenmanagers auf dem anderen Knoten

Damit das Clustering mit IBM MQ funktioniert, benötigen Sie für jeden auf Knoten A einen identischen Warteschlangenmanager auf Knoten B. Es ist jedoch nicht erforderlich, die zweite explizit zu erstellen. Sie können einen Warteschlangenmanager auf einem Knoten erstellen oder vorbereiten, ihn wie in „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531 beschrieben in den anderen Knoten versetzen und auf diesem Knoten vollständig duplizieren.

## Standardwarteschlangenmanager

Verwenden Sie keinen Standard-WS-Manager unter MSCS-Steuerung. Ein Warteschlangenmanager verfügt über keine Eigenschaft, die ihn zum Standard macht. IBM MQ behält seinen eigenen separaten Datensatz bei. Wenn Sie einen Warteschlangenmanager so einstellen, dass er als Standardwert für den anderen Computer bei der Funktionsübernahme verwendet wird, wird er nicht zum Standardwert dort. Stellen Sie sicher, dass alle Anwendungen auf bestimmte WS-Manager nach Namen verweisen.

## Löschen eines Warteschlangenmanagers

Sobald ein WS-Manager einen Knoten verschoben hat, sind seine Details in der Registry auf beiden Computern vorhanden. Wenn Sie den Warteschlangenmanager löschen möchten, tun Sie dies wie üblich auf einem Computer und führen Sie dann das in „Unterstützung für MSCS-Dienstprogramme” auf Seite 545 beschriebene Dienstprogramm aus, um die Registry auf dem anderen Computer zu bereinigen.

## Unterstützung für vorhandene WS-Manager

Sie können einen vorhandenen Warteschlangenmanager unter MSCS-Steuerung stellen, vorausgesetzt, Sie können die Protokolldateien und die Warteschlangendateien Ihres Warteschlangenmanagers auf einer Platte, die sich auf dem gemeinsam genutzten SCSI-Bus befindet, zwischen den beiden Maschinen (siehe Abbildung 71 auf Seite 528) setzen. Sie müssen den WS-Manager kurz offline schalten, während die MSCS-Ressource erstellt wird.

Wenn Sie einen neuen Warteschlangenmanager erstellen möchten, erstellen Sie ihn unabhängig von MSCS, testen Sie ihn, und legen Sie ihn dann unter MSCS-Steuerung. Unter

- „Warteschlangenmanager für die Verwendung mit MSCS erstellen” auf Seite 530
- „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531
- „WS-Manager unter MSCS-Steuerung einschalten” auf Seite 533

## MSCS-Warteschlangenmanager für die Verwaltung von Telling

Sie können auswählen, welche Warteschlangenmanager unter MSCS-Steuerung platziert werden, indem Sie den MSCS-Clusteradministrator verwenden, um eine Ressourceninstanz für jeden dieser Warteschlangenmanager zu erstellen. In diesem Prozess wird eine Liste der Ressourcen angezeigt, aus denen Sie den Warteschlangenmanager auswählen können, den diese Instanz verwalten soll.

## Warteschlangenmanagerprotokolldateien

Wenn Sie einen Warteschlangenmanager in den MSCS-Speicher versetzen, versetzen Sie seine Protokoll- und Datendateien auf eine gemeinsam genutzte Platte (siehe „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531).

Es empfiehlt sich, vor dem Verschieben den Warteschlangenmanager ordnungsgemäß zu beenden und eine vollständige Sicherung der Datendateien und Protokolldateien zu erstellen.

## Mehrere Warteschlangenmanager

Mit der MSCS-Unterstützung von IBM MQ können Sie mehrere Warteschlangenmanager auf jeder Maschine ausführen und einzelne Warteschlangenmanager unter MSCS-Steuerung stellen.

Windows *Immer MSCS zum Verwalten von Clustern verwenden*

Versuchen Sie nicht, mit Hilfe der Steuerbefehle oder des IBM MQ Explorers die Operationen zum Starten und Stoppen direkt auf einem beliebigen Warteschlangenmanager unter der Kontrolle von MSCS auszuführen. Verwenden Sie stattdessen MSCS Cluster Administrator, um den WS-Manager online zu schalten oder offline zu schalten.

Die Verwendung des MSCS-Clusteradministrators ist zum Teil die Vermeidung möglicher Verwechslungen durch MSCS, die gemeldet werden, dass der Warteschlangenmanager offline ist, wenn Sie ihn tatsächlich außerhalb der Steuerung von MSCS gestartet haben. Ein schwerwiegerer Fehler beim Stoppen eines Warteschlangenmanagers ohne MSCS wird von MSCS als Fehler erkannt und die Übernahme durch den anderen Knoten eingeleitet.

Windows *Arbeiten im Aktiv/Aktiv-Modus in MSCS*

Beide Computer im MSCS-Cluster können WS-Manager im Aktiv/Aktiv-Modus ausführen. Es ist nicht erforderlich, eine vollständig inaktive Maschine als Standby-Server zu verwenden (aber Sie können, wenn Sie möchten, im Modus 'Aktiv/Passiv') ausgeführt werden.

Wenn Sie beide Maschinen zur Ausführung der Workload verwenden möchten, stellen Sie jeder Maschine ausreichend Kapazität (Prozessor, Speicher, Sekundärspeicher) zur Verfügung, um die gesamte Clusterauslastung mit einem zufriedenstellenden Leistungsniveau zu führen.

**Anmerkung:** Wenn Sie MSCS zusammen mit Microsoft Transaction Server (COM +) verwenden, können Sie **nicht** den Modus 'Aktiv/Aktiv' verwenden. Dies liegt daran, dass IBM MQ mit MSCS und COM + verwendet wird:

- Anwendungskomponenten, die die COM+-Unterstützung von IBM MQ verwenden, müssen auf demselben Computer ausgeführt werden wie der Distributed Transaction Coordinator (DTC), ein Teil von COM+.
- Der WS-Manager muss auch auf demselben Computer ausgeführt werden.
- Die DTC muss als MSCS-Ressource konfiguriert werden und kann daher zu einem beliebigen Zeitpunkt auf einem der Computer im Cluster ausgeführt werden.

Windows *PostOnlineCommand und PreOfflineCommand in MSCS*

Verwenden Sie diese Befehle, um die MSCS-Unterstützung von IBM MQ bei anderen Systemen zu integrieren. Sie können sie zum Absetzen von IBM MQ-Befehlen verwenden, d. h. mit einigen Einschränkungen.

Geben Sie diese Befehle in den Parametern für eine Ressource des Typs `IBM MQ MSCS` an. Sie können sie verwenden, um die MSCS-Unterstützung von IBM MQ mit anderen Systemen oder Prozeduren zu integrieren. Sie können z. B. den Namen eines Programms angeben, das eine E-Mail-Nachricht sendet, einen Pager aktiviert oder eine andere Form von Alert generiert, die von einem anderen Überwachungssystem erfasst werden soll.

Der Befehl 'PostOnlineCommand' wird aufgerufen, wenn sich die Ressource von offline in online ändert. Der Befehl 'PreOfflineCommand' wird für einen Wechsel von online in offline aufgerufen. Wenn diese Befehle aufgerufen werden, werden sie standardmäßig aus dem Windows-Systemverzeichnis ausgeführt. Da IBM MQ einen 32-Bit-Ressourcenüberwachungsprozess auf Windows 64-Bit-Systemen verwendet, ist dies das `\Windows\SysWOW64`-Verzeichnis und nicht das `\Windows\system32`-Verzeichnis. Weitere Informationen finden Sie in der Microsoft-Dokumentation zur Dateiumleitung in einer Windows x64-Umgebung. Beide Befehle werden unter dem Benutzeraccount ausgeführt, der für die Ausführung des MSCS-Clusterservice verwendet wird. Sie werden asynchron aufgerufen; IBM MQ MSCS-Unterstützung wartet nicht darauf, dass sie abgeschlossen werden, bevor sie fortgesetzt werden. Dadurch wird das Risiko ausgeschlossen, dass weitere Clusteroperationen blockiert oder verzögert werden.

Sie können diese Befehle auch zum Absetzen von IBM MQ-Befehlen verwenden, z. B. um Requesterkanäle erneut zu starten. Die Befehle werden jedoch zu dem Zeitpunkt ausgeführt, zu dem sich die Statusänderungen des Warteschlangenmanagers ändern, so dass sie nicht für die Ausführung von Funktionen mit langer Laufzeit gedacht sind und keine Annahmen über den aktuellen Status des Warteschlangenmanagers treffen müssen. Es ist durchaus möglich, dass ein Administrator unmittelbar nach dem Online-Betrieb des Warteschlangenmanagers einen Offline-Befehl abgesetzt hat.

Wenn Sie Programme ausführen wollen, die vom Status des Warteschlangenmanagers abhängig sind, sollten Sie die Erstellung von Instanzen des Ressourcentyps `MSCS Generic Application` in der gleichen MSCS-Gruppe wie die Warteschlangenmanager-Ressource in Betracht ziehen und sie abhängig von der WS-Manager-Ressource machen.

Windows *Bevorzugte Knoten in MSCS verwenden*

Es kann nützlich sein, wenn Sie den Active/Active-Modus in MSCS verwenden, um einen *bevorzugten Knoten* für jeden Warteschlangenmanager zu konfigurieren. Im Allgemeinen ist es jedoch besser, keinen bevorzugten Knoten festzulegen, sondern sich auf einen manuellen Failback zu verlassen.

Im Gegensatz zu anderen relativ statusunabhängigen Ressourcen kann ein Warteschlangenmanager eine Weile dauern, bis er (oder zurück) von einem Knoten zum anderen fehlschlägt. Um unnötige Ausfälle zu vermeiden, testen Sie den wiederhergestellten Knoten, bevor Sie einen WS-Manager wieder in den Knoten zurückschlagen. Dies schließt die Verwendung der `immediate` -Failback-Einstellung aus. Sie können die Failback-Funktion zwischen bestimmten Tageszeiten konfigurieren.

Die sicherste Route ist wahrscheinlich, dass der WS-Manager manuell auf den erforderlichen Knoten zurückversetzt wird, wenn Sie sicher sind, dass der Knoten vollständig wiederhergestellt ist. Dies schließt die Verwendung der Option `preferred node` aus.

Windows *COM + -Fehler bei der Installation von auf MSCS*

Wenn Sie IBM MQ auf einem neu installierten MSCS-Cluster installieren, finden Sie möglicherweise einen Fehler bei Source COM+ und der Ereignis-ID 4691, die im Anwendungsereignisprotokoll aufgelistet werden.

Dies bedeutet, dass Sie versuchen, IBM MQ in einer MSCS-Umgebung (MSCS = Microsoft Cluster Server) auszuführen, wenn der Microsoft Distributed Transaction Coordinator (MSDTC) nicht für die Ausführung in einer solchen Umgebung konfiguriert wurde. Informationen zum Konfigurieren von MSDTC in einer Clusterumgebung finden Sie in der Microsoft-Dokumentation.

## Windows *Unterstützung für MSCS-Dienstprogramme*

Eine Liste der MSCS-Dienstprogramme ( IBM MQ Support for Microsoft Cluster Service ), die Sie an einer Eingabeaufforderung ausführen können

**Anmerkung:** Ab Windows Server 2016lautet der neue Name für Microsoft Cluster Service (MSCS) Windows Server Failover Clustering (WSFC).

IBM MQ-Unterstützung für MSCS umfasst die folgenden Dienstprogrammprogramme:

**Ressourcentyp registrieren/deregistrieren**
`haregtyp.exe`

Nachdem Sie die Registrierung des IBM MQ MSCS-Ressourcentyps *aufgehoben* haben, können Sie keine Ressourcen dieses Typs mehr erstellen. MSCS lässt die Deregistrierung eines Ressourcentyps nicht zu, wenn noch Instanzen dieses Typs in dem Cluster vorhanden sind:

1. Stoppen Sie mit Hilfe des MSCS-Clusteradministrators alle Warteschlangenmanager, die unter MSCS-Steuerung ausgeführt werden, indem Sie sie offline schalten, wie in „WS-Manager offline von MSCS übernehmen” auf Seite 541 beschrieben.
2. Löschen Sie die Ressourceninstanzen mit Hilfe des MSCS-Clusteradministrators.
3. Heben Sie an einer Eingabeaufforderung die Registrierung des Ressourcentyps auf, indem Sie den folgenden Befehl eingeben:

```
haregtyp /u
```

Wenn Sie den Typ *registrieren* (oder zu einem späteren Zeitpunkt erneut registrieren) möchten, geben Sie an einer Eingabeaufforderung den folgenden Befehl ein:

```
haregtyp /r
```

Nachdem Sie die MSCS-Bibliotheken erfolgreich registriert haben, müssen Sie das System neu starten, wenn Sie dies seit der Installation von IBM MQ noch nicht getan haben.

**WS-Manager in MSCS-Speicher verschieben**
`hamvmqm.exe`

Weitere Informationen finden Sie unter „Warteschlangenmanager in MSCS-Speicher versetzen” auf Seite 531.

**WS-Manager aus einem Knoten löschen**
`hadltmqm.exe`

Angenommen, Sie haben einen Warteschlangenmanager in Ihrem Cluster, der von einem Knoten in einen anderen verschoben wurde, und jetzt möchten Sie ihn zerstören. Verwenden Sie den IBM MQ-Explorer, um ihn auf dem Knoten zu löschen, auf dem er sich derzeit befindet. Die Registry-Einträge für diese Datei sind noch auf dem anderen Computer vorhanden. Um diese zu löschen, geben Sie den folgenden Befehl an einer Eingabeaufforderung auf diesem Computer ein:

```
hadltmqm /m qmname
```

Dabei steht `qmname` für den Namen des zu entfernenden Warteschlangenmanager.

**Setup-Details prüfen und speichern**
`amqmsysn.exe`

Dieses Dienstprogramm stellt einen Dialog mit vollständigen Details Ihrer Einrichtung von IBM MQ-MSCS-Unterstützung dar, z. B. wenn Sie die Unterstützung von IBM aufrufen. Es gibt eine Option zum Speichern der Details in einer Datei.

## Multi Warteschlangenmanager mit mehreren Instanzen

Multi-Instanz-Warteschlangenmanager sind Instanzen desselben Warteschlangenmanagers, die auf verschiedenen Servern konfiguriert sind. Eine Instanz des Warteschlangenmanagers ist als aktive Instanz definiert, die andere ist eine Standby-Instanz. Wenn die aktive Instanz ausfällt, wird der Multi-Instanz-Warteschlangenmanager automatisch auf dem Standby-Server gestartet.

### Beispiel einer WS-Manager-Konfiguration mit mehreren Instanzen

Abbildung 72 auf Seite 546 zeigt ein Beispiel für eine Konfiguration mit mehreren Instanzen für WS-Manager QM1. IBM MQ ist auf zwei Servern installiert, von denen einer als Ersatz dient. Es wurde ein Warteschlangenmanager (QM1) erstellt. Eine Instanz von QM1 ist aktiv und wird auf einem einzigen Server ausgeführt. Die andere Instanz von QM1 wird in der Bereitschaftsdatenbank auf dem anderen Server ausgeführt, ohne aktive Verarbeitung zu tun, aber bereit ist, von der aktiven Instanz von QM1 zu übernehmen, wenn die aktive Instanz fehlschlägt. (In einer Konfiguration mit mehreren Instanzen kann nur eine aktive Instanz und eine Standby-Instanz des Warteschlangenmanagers vorhanden sein.)

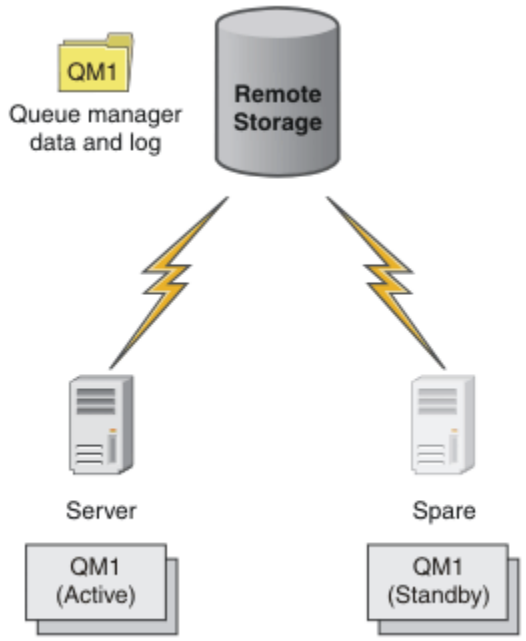



*Abbildung 72. Multi-Instanz-Warteschlangenmanager*

Wenn Sie einen Warteschlangenmanager als Multi-Instanz-Warteschlangenmanager verwenden wollen, erstellen Sie einen einzelnen Warteschlangenmanager auf einem der Server mit dem Befehl **crtmqm** , indem Sie seine Warteschlangenmanagerdaten und -protokolle in den gemeinsam genutzten Netzspeicher stellen. Verwenden Sie auf dem anderen Server den Befehl **addmqinf** , um einen Verweis auf die Warteschlangenmanagerdaten und -protokolle im Netzspeicher zu erstellen, anstatt den Warteschlangenmanager erneut zu erstellen.

Sie können nun den Warteschlangenmanager von einem der beiden Server aus ausführen. Jeder der Server verweist auf dieselben WS-Manager-Daten und -Protokolle. Es gibt nur einen Warteschlangenmanager, und er ist auf nur einem Server gleichzeitig aktiv.

Der Warteschlangenmanager kann entweder als Einzelinstanzwarteschlangenmanager oder als Warteschlangenmanager mit mehreren Instanzen ausgeführt werden. In beiden Fällen wird nur eine Instanz des Warteschlangenmanagers ausgeführt, Verarbeitungsanforderungen werden verarbeitet. Der Unterschied besteht darin, dass der Server, der die aktive Instanz des Warteschlangenmanagers nicht ausführt,

als Standby-Instanz ausgeführt wird, wenn er als Standby-Warteschlangenmanager ausgeführt wird und bereit ist, automatisch von der aktiven Instanz zu übernehmen, wenn der aktive Server ausfällt.

Das einzige Steuerelement, über das Sie über die Instanz verfügen, ist die Reihenfolge, in der Sie den WS-Manager auf den beiden Servern starten. Die erste Instanz, die Lese-/Schreibsperren für die WS-Manager-Daten erhält, wird zum aktiven Exemplar.

Sie können die aktive Instanz, nachdem sie gestartet wurde, in den anderen Server eintauschen, indem Sie die aktive Instanz mit der Umschaltoption stoppen, um die Steuerung an den Standby-Server zu übertragen.

Die aktive Instanz von QM1 hat exklusiven Zugriff auf die Daten des gemeinsam genutzten Warteschlangenmanagers und protokolliert Ordner, wenn sie ausgeführt wird. Die Standby-Instanz von QM1 erkennt, wenn die aktive Instanz ausgefallen ist, und wird zur aktiven Instanz. Sie übernimmt die QM1-Daten und -Protokolle in dem Status, den sie von der aktiven Instanz hinterlassen haben, und akzeptiert die Verbindungen von Clients und Kanälen.

Die aktive Instanz kann aus verschiedenen Gründen fehlschlagen, die dazu führen, dass die Bereitschaftsdatenbank die folgenden Schritte übernimmt:

- Fehler des Servers, auf dem sich die aktive WS-Manager-Instanz befindet.
- Fehler bei der Verbindung zwischen dem Server, auf dem sich die aktive WS-Manager-Instanz und das Dateisystem befinden.
- IBM MQ erkennt keine Reaktion der Prozesse des Warteschlangenmanagers und beendet daraufhin den Warteschlangenmanager.

Sie können die WS-Manager-Konfigurationsinformationen zu mehreren Servern hinzufügen und die beiden Server als Aktiv/Standby-Paar ausführen. Es gibt eine Begrenzung von insgesamt zwei Instanzen. Es können keine zwei Standby-Instanzen und eine aktive Instanz vorhanden sein.

## Zusätzliche Komponenten, die zum Erstellen einer Hochverfügbarkeitslösung benötigt werden

Ein WS-Manager mit mehreren Instanzen ist Teil einer Hochverfügbarkeitslösung. Sie benötigen einige zusätzliche Komponenten, um eine nützliche Hochverfügbarkeitslösung zu erstellen.

- Verbindungswiederholung von Client und Kanal zur Übertragung von IBM MQ-Verbindungen auf den Computer, der die Ausführung der aktiven Instanz des Warteschlangenmanagers übernimmt.
- Ein gemeinsam genutzes Network File System (NFS) mit hoher Leistung, das die Sperren korrekt verwaltet und den Schutz vor einem Datenträger-und Dateiserverfehler bietet.

  **Wichtig:** Sie müssen alle WS-Manager-Instanzen mit mehreren Instanzen stoppen, die in Ihrer Umgebung ausgeführt werden, bevor Sie die Wartung auf dem NFS-Laufwerk ausführen können. Stellen Sie sicher, dass die Konfigurationssicherungen des Warteschlangenmanagers im Falle eines NFS-Fehlers wiederhergestellt werden müssen.
- Resiliente Netze und Netzteile, um Single Points of Failure in der Basisinfrastruktur zu eliminieren.
- Anwendungen, die Failover tolerieren. Sie sollten insbesondere auf das Verhalten von Transaktionsanwendungen und von Anwendungen achten, die IBM MQ-Warteschlangen durchsuchen.
- Überwachung und Verwaltung der aktiven und Standby-Instanzen, um sicherzustellen, dass sie aktiv sind, und um aktive Instanzen erneut zu starten, die fehlgeschlagen sind. Obwohl Warteschlangenmanager mit mehreren Instanzen automatisch erneut gestartet werden, müssen Sie sicher sein, dass Ihre Standby-Instanzen aktiv sind, bereit sind, um zu übernehmen, und dass fehlgeschlagene Instanzen wieder als neue Standby-Instanzen wieder online sind.

IBM MQ MQI clients und Kanäle verbinden sich automatisch wieder mit dem Standby-WS-Manager, wenn er aktiv wird. Weitere Informationen zur erneuten Verbindung und zu den anderen Komponenten in einer Hochverfügbarkeitslösung finden Sie in den zugehörigen Themen. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt.

## Unterstützte Plattformen

Sie können einen Warteschlangenmanager mit mehreren Instanzen auf einer beliebigen Nicht-z/OS -Plattform erstellen.

Die automatische Clientverbindungswiederholung wird für MQI-Clients unterstützt.

### *Erstellen eines Warteschlangenmanagers mit mehreren Instanzen*

Erstellen Sie einen Multi-Instanz-Warteschlangenmanager, indem Sie den Warteschlangenmanager auf einem Server erstellen und IBM MQ auf einem anderen Server konfigurieren. Warteschlangenmanager mit mehreren Instanzen nutzen die Daten und Protokolle des Warteschlangenmanagers gemeinsam.

Der größte Aufwand bei der Erstellung eines Multi-Instanz-WS-Managers ist die Aufgabe, die Daten und Protokolldateien des gemeinsam genutzten Warteschlangenmanagers zu konfigurieren. Sie müssen gemeinsam genutzte Verzeichnisse im Netzspeicher erstellen und die Verzeichnisse für andere Server mit Netzfreigaben verfügbar machen. Diese Aufgaben müssen von einer Person mit Administratorberechtigung, z. B. *Root* auf AIX and Linux-Systemen, durchgeführt werden. Die Schritte sind wie folgt:

1. Erstellen Sie die Freigaben für die Daten-und Protokolldateien.
2. Erstellen Sie den WS-Manager auf einem einzigen Server.
3. Führen Sie den Befehl **dspmqinf** auf dem ersten Server aus, um die Konfigurationsdaten des Warteschlangenmanagers zu erfassen und in die Zwischenablage zu kopieren.
4. Führen Sie den Befehl **addmqinf** mit den kopierten Daten aus, um die Warteschlangenmanagerkonfiguration auf dem zweiten Server zu erstellen.

Sie führen **crtmqm** nicht aus, um den WS-Manager erneut auf dem zweiten Server zu erstellen.

## Dateizugriffssteuerung

Sie müssen darauf achten, dass der Benutzer und die Gruppe `mqm` auf allen anderen Servern die Berechtigung zum Zugriff auf die Freigaben haben.

Unter AIX and Linux müssen Sie `uid` und `gid` für `mqm` auf allen Systemen gleich festlegen. Möglicherweise müssen Sie `/etc/passwd` auf jedem System bearbeiten, um eine gemeinsame `uid` und `gid` für `mqm` festzulegen, und anschließend einen Warmstart Ihres Systems.

Unter Microsoft Windows muss die Benutzer-ID, die die Warteschlangenmanagerprozesse ausführt, über die vollständige Steuerungsberechtigung für die Verzeichnisse verfügen, die die Daten und Protokolldateien des Warteschlangenmanagers enthalten. Sie können die Berechtigung auf zwei Arten konfigurieren:

1. Erstellen Sie einen Warteschlangenmanager mit einer globalen Gruppe als alternativen Sicherheitprincipal. Autorisieren Sie die globale Gruppe, um vollständigen Steuerungszugriff auf die Verzeichnisse zu haben, die Warteschlangenmanagerdaten und Protokolldateien enthalten. Weitere Informationen finden Sie in „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern” auf Seite 579. Erstellen Sie die Benutzer-ID, unter der der Warteschlangenmanager ausgeführt wird, ein Mitglied der globalen Gruppe. Sie können keinen lokalen Benutzer zu einem Mitglied einer globalen Gruppe machen, daher müssen die WS-Manager-Prozesse unter einer Domänenbenutzer-ID ausgeführt werden. Die Domänenbenutzer-ID muss ein Mitglied der lokalen Gruppe `mqm`sein. Die Task „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen” auf Seite 551 veranschaulicht, wie ein WS-Manager mit mehreren Instanzen unter Verwendung von auf diese Weise gesicherten Dateien eingerichtet wird.
2. Erstellen Sie einen Warteschlangenmanager auf dem Domänencontroller, so dass die lokale `mqm`-Gruppe einen Domänenbereich hat, "lokale Domäne". Sichern Sie die Dateifreigabe mit den lokalen `mqm`-Domänen und führen Sie Warteschlangenmanager-Prozesse für alle Instanzen eines Warteschlangenmanagers unter derselben lokalen `mqm`-Gruppe aus. Die Task „WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen” auf Seite 568 veranschaulicht, wie ein WS-Manager mit mehreren Instanzen unter Verwendung von auf diese Weise gesicherten Dateien eingerichtet wird.

## Konfigurationsdaten

Konfigurieren Sie so viele Warteschlangenmanagerinstanzen, wie Sie benötigen, indem Sie die Konfigurationsdaten des IBM MQ-Warteschlangenmanagers für jeden Server ändern. Auf jedem Server muss die gleiche Version von IBM MQ auf einer kompatiblen Fixversion installiert sein. Die Befehle **dspmqinf** und **addmqinf** unterstützen Sie bei der Konfiguration der zusätzlichen Warteschlangenmanagerinstanzen. Alternativ können Sie die `mqs.ini`- und `qm.ini`-Dateien direkt bearbeiten. Die Themen „Warteschlangenmanager mit mehreren Instanzen unter Linux erstellen" auf Seite 592, „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen" auf Seite 551 und „WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen" auf Seite 568 sind Beispiele für die Konfiguration eines Multi-Instanz-Warteschlangenmanagers.

Auf AIX, Linux, and Windows-Systemen können Sie eine einzelne `mqs.ini`-Datei gemeinsam nutzen, indem Sie sie auf dem gemeinsam verwendeten Netzbereich platzieren und die Umgebungsvariable **AMQ_MQS_INI_LOCATION** so festlegen, dass sie auf sie verweist.

## Einschränkungen

1. Konfigurieren Sie mehrere Instanzen desselben Warteschlangenmanagers nur auf Servern, die dasselbe Betriebssystem, dieselbe Architektur und Endian-Format aufweisen. Beide Maschinen müssen zum Beispiel entweder 32-Bit oder 64-Bit sein.
2. Alle IBM MQ-Installationen müssen Release-Level 7.0.1 oder höher aufweisen.
3. In der Regel werden aktive Installationen und Standby-Installationen auf derselben Wartungsstufe verwaltet. Lesen Sie die Wartungsanweisungen für jedes Upgrade, um zu überprüfen, ob Sie alle Installationen gemeinsam aktualisieren müssen.

   Beachten Sie, dass die Wartungsstufen für die aktiven und passiven WS-Manager identisch sein müssen.
4. Teilen Sie Warteschlangenmanagerdaten und -protokolle nur zwischen Warteschlangenmanagern, die mit demselben IBM MQ-Benutzer, derselben Gruppe und demselben Zugriffssteuerungsmechanismus konfiguriert sind. IBM i Die Netzfreigabe auf einem Linux-Server könnte beispielsweise separate Warteschlangenmanagerdaten und -protokolle für AIX and Linux-Warteschlangenmanager enthalten, aber die von IBM i verwendeten Warteschlangenmanagerdaten nicht enthalten.

   IBM i Sie können mehrere Freigaben in demselben vernetzten Speicher für IBM i und für AIX and Linux-Systeme erstellen, solange die Freigaben unterschiedlich sind. Sie können verschiedene Eigentümern verschiedene Freigaben geben. Die Einschränkung ist eine Folge der unterschiedlichen Namen, die für die IBM MQ-Benutzer und -Gruppen zwischen AIX and Linux und IBM iverwendet werden. Die Tatsache, dass der Benutzer und die Gruppe denselben `uid` und `gid` haben können, lockt die Einschränkung nicht ab.
5. Auf AIX and Linux-Systemen konfigurieren Sie das gemeinsam genutzte Dateisystem auf dem vernetzten Speicher am besten mit einem `unbedingten`, unterbrechbaren Mount, statt mit einem `bedingten` Mount. Eine unterbrechungsfreie Halterung erzwingt die Blockierung des Warteschlangenmanagers, bis die Unterbrechung durch einen Systemaufruf unterbrochen wird. Weiche Mounts stellen keine Datenkonsistenz nach einem Serverausfall sicher.
6. Das gemeinsam genutzte Protokoll und die Datenverzeichnisse können nicht auf einem FAT oder einem NFSv3-Dateisystem gespeichert werden. Bei Warteschlangenmanagern mit mehreren Instanzen unter Windows muss auf den Netzspeicher durch das Common Internet File System (CIFS) zugegriffen werden, das von Windows-Netzen verwendet wird.
7. z/OS z/OS unterstützt keine Multi-Instanz-Warteschlangenmanager. Verwenden Sie Gruppen mit gemeinsamer Warteschlange.

   Wiederverbindungsfähige Clients arbeiten mit z/OS-Warteschlangenmanagern.

Windows *Windows-Domänen und Multi-Instanz-Warteschlangenmanager*

Ein Multi-Instanz-Warteschlangenmanager unter Windows erfordert die gemeinsame Nutzung seiner Daten und Protokolle. Die Freigabe muss für alle Instanzen des Warteschlangenmanagers, die auf verschiedenen Servern oder Workstations ausgeführt werden, zugänglich sein. Konfigurieren Sie die Warteschlangenmanager und nutzen Sie sie als Teil einer Windows-Domäne gemeinsam. Der Warteschlangenmanager kann auf einer Domänenworkstation oder einem Server oder auf dem Domänencontroller ausgeführt werden.

**Wichtig:** Computer, die mit Windows 10 Version 1607 und Windows Server 2016 beginnen, sind standardmäßig restriktiver als frühere Versionen von Windows.

Durch diese Änderung können Clients, die remote Aufrufe an den Sicherheitskontenmanager (SAM) vornehmen dürfen, eingeschränkt werden, und sie können sich auf IBM MQ auswirken, wenn die Warteschlangenmanager nicht gestartet werden. Der Zugriff auf SAM ist für das Funktionieren von IBM MQ kritisch, wenn IBM MQ als Domänenkonto konfiguriert ist.

Lesen Sie vor der Konfiguration eines Multi-Instanz-Warteschlangenmanagers die Abschnitte „Nicht gemeinsam genutzte WS-Manager-Daten und -Protokollverzeichnisse und -Dateien unter Windows schützen" auf Seite 582 und „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern" auf Seite 579, um zu erfahren, wie der Zugriff auf Warteschlangenmanagerdaten und Protokolldateien gesteuert werden kann. Diese Abschnitte enthalten lehrreiche Informationen. Wenn Sie direkt mit der Einrichtung gemeinsam genutzter Verzeichnisse für einen Multi-Instanz-Warteschlangenmanager in einer Windows-Domäne beginnen möchten, beachten Sie die Informationen im Abschnitt „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen" auf Seite 551.

## Multi-Instanz-WS-Manager auf Domänenworkstations oder Servern ausführen

Ab IBM WebSphere MQ 7.1 werden Multi-Instanz-Warteschlangenmanager auf einer Workstation oder einem Server ausgeführt, die/der Mitglied einer Domäne ist. Um einen Multi-Instanz-Warteschlangenmanager unter Windows auszuführen, benötigen Sie einen Domänencontroller, einen Dateiserver sowie zwei Workstations oder Server, auf denen derselbe Warteschlangenmanager ausgeführt wird, der mit derselben Domäne verbunden ist.

Die Änderung, die die Ausführung eines Warteschlangenmanagers mit mehreren Instanzen auf einem beliebigen Server oder einer Workstation in einer Domäne ermöglicht, besteht darin, dass Sie jetzt einen Warteschlangenmanager mit einer zusätzlichen Sicherheitsgruppe erstellen können. Die zusätzliche Sicherheitsgruppe wird mit dem Befehl **crtmqm** im Parameter -a übergeben. Sie sichern die Verzeichnisse, die die Daten des Warteschlangenmanagers enthalten, und Protokolle mit der Gruppe. Die Benutzer-ID, die WS-Manager-Prozesse ausführt, muss ein Mitglied dieser Gruppe sein. Wenn der Warteschlangenmanager auf die Verzeichnisse zugreift, überprüft Windows die Berechtigungen, über die die Benutzer-ID für den Zugriff auf die Verzeichnisse verfügt. Wenn sowohl die Gruppe als auch der Benutzer-ID-Domänenbereich angegeben wird, verfügt die Benutzer-ID, die die WS-Manager-Prozesse ausführt, über Berechtigungsnachweise aus der globalen Gruppe. Wenn der WS-Manager auf einem anderen Server ausgeführt wird, kann die Benutzer-ID, die die WS-Manager-Prozesse ausführt, dieselben Berechtigungsnachweise haben. Die Benutzer-ID muss nicht identisch sein. Es muss ein Mitglied der alternativen Sicherheitsgruppe sowie ein Mitglied der lokalen `mqm` -Gruppe sein.

Details zum Erstellen eines Multi-Instanz-Warteschlangenmanagers finden Sie unter „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen" auf Seite 551 .

Es sind mehrere Schritte erforderlich, um die Domäne und die Domänenserver und Workstations zu konfigurieren. Sie müssen sich darüber im Klaren sein, wie Windows den Zugriff eines Warteschlangenmanagers auf seine Daten- und Protokollverzeichnisse autorisiert. Wenn Sie nicht sicher sind, wie Warteschlangenmanagerprozesse berechtigt werden, auf ihre Protokoll- und Datendateien zuzugreifen, lesen Sie den Abschnitt „Nicht gemeinsam genutzte WS-Manager-Daten und -Protokollverzeichnisse und -Dateien unter Windows schützen" auf Seite 582. Das Thema enthält zwei Tasks, die Ihnen helfen, die Schritte zu verstehen, die erforderlich sind. Es handelt sich um folgende Tasks: „Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind" auf Seite 584 und „Lesen und Schreiben von Daten-und Protokolldateien, die von einer alternativen lokalen Sicherheitsgruppe

autorisiert sind” auf Seite 588. In einem weiteren Abschnitt, „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern” auf Seite 579, wird erläutert, wie gemeinsam genutzte Verzeichnisse mit Warteschlangenmanagerdaten und Protokolldateien mit der alternativen Sicherheitsgruppe gesichert werden. Der Abschnitt umfasst vier Tasks, die Einrichtung einer Windows-Domäne, die Erstellung einer Dateifreigabe, die Installation von IBM MQ for Windows und die Konfiguration eines Warteschlangenmanagers für die Verwendung der Freigabe. Die Aufgaben lauten wie folgt:

1. „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555.
2. „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren” auf Seite 558.
3. „Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows” auf Seite 562.
4. „Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind” auf Seite 565.

Anschließend können Sie die Task „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen” auf Seite 551 unter Verwendung der Domäne ausführen. Führen Sie diese Tasks aus, um die Konfiguration eines Warteschlangenmanagers mit mehreren Instanzen zu untersuchen, bevor Sie Ihre Kenntnisse in eine Produktionsdomäne übertragen.

## Multi-Instanz-WS-Manager auf Domänencontrollern ausführen

Die WS-Manager-Daten konnten mit der Domäne `mqm` gesichert werden. Wie im Abschnitt „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern” auf Seite 579 erläutert, können Sie Verzeichnisse, die mit der lokalen `mqm`-Gruppe auf Workstations oder Servern geschützt sind, nicht gemeinsam nutzen. Auf Domänencontrollern haben jedoch alle Gruppen-und Principals einen Domänenbereich. Wenn Sie IBM MQ for Windows auf einem Domänencontroller installieren, werden die Warteschlangenmanagerdaten und Protokolldateien mit der `mqm`-Gruppe der Domäne geschützt, die gemeinsam genutzt werden kann. Führen Sie die Schritte in der Task „WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen” auf Seite 568 aus, um einen Multi-Instanz-Warteschlangenmanager auf Domänencontrollern zu konfigurieren.

**Zugehörige Informationen**

Berechtigungs-und Zugriffssteuerung verwalten

Verwendung von Windows Server-Clusterknoten als Domänencontroller

Windows *WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen*

Anhand eines Beispiels wird veranschaulicht, wie ein Multi-Instanz-Warteschlangenmanager auf einer Workstation oder einem Server unter Windows eingerichtet wird, der Teil einer Windows-Domäne ist. Der Server muss kein Domänencontroller sein. In der Konfiguration werden die verwendeten Konzepte und nicht die Produktionsmaßstab, sondern die Konzepte veranschaulicht. Das Beispiel basiert auf Windows Server 2008. Möglicherweise weichen die Schritte auf anderen Windows Server-Versionen im Einzelnen ab.

In einer Produktionsmaßstabskonfiguration müssen Sie die Konfiguration möglicherweise an eine vorhandene Domäne anpassen. Sie können z. B. verschiedene Domänengruppen definieren, um unterschiedliche Freigaben zu berechtigen und die Benutzer-IDs zu gruppieren, die Warteschlangenmanager ausführen.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
: Ein Domänencontroller unter Windows Server 2008. Sie ist Eigner der *wmq.example.com*-Domäne, die *Sun*, *mars* und *venus*enthält. Für die Zwecke der Veranschaulichung wird sie auch als Dateiserver verwendet.

***mars***
: Ein Windows Server 2008, der als erster IBM MQ-Server verwendet wird. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***venus***
Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

Unter Windows muss das Dateisystem, auf dem die Warteschlangenmanager-Daten und Protokolldateien gespeichert werden sollen, nicht überprüft werden. Die Prüfprozedur Verhalten des gemeinsam genutzten Dateisystems überprüfen ist auf AIX and Linux anwendbar. Unter Windows werden die Prüfungen immer mit positivem Ergebnis abgeschlossen.

Führen Sie die Schritte in den folgenden Tasks aus. Die Tasks erstellen den Domänencontroller und die Domäne, installieren IBM MQ for Windows auf einem Server und erstellen den Dateifreigabewert für Daten und Protokolldateien. Wenn Sie einen bereits vorhandenen Domänencontroller konfigurieren möchten, ist es unter Umständen hilfreich, die durchzuführenden Schritte an einem neuen Windows Server2008 zu testen. Sie können die Schritte an Ihre Domäne anpassen.

1. „Active Directory- und DNS-Domäne unter Windows erstellen" auf Seite 555.
2. „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren" auf Seite 558.
3. „Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows" auf Seite 562.
4. „Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind" auf Seite 565.

## Informationen zu diesem Vorgang

Diese Task ist eine der Tasks, die zum Konfigurieren eines Domänencontrollers und zwei Servern in der Domäne ausgeführt werden, um Instanzen eines Warteschlangenmanagers auszuführen. In dieser Task konfigurieren Sie einen zweiten Server, *venus*, um eine andere Instanz des Warteschlangenmanagers *QMGR* auszuführen. Führen Sie die Schritte in dieser Task aus, um die zweite Instanz des Warteschlangenmanagers *QMGR* zu erstellen und zu testen, ob sie funktioniert.

Diese Task ist von den vier Tasks im vorherigen Abschnitt getrennt. Sie enthält die Schritte, mit denen ein einzelner Instanz-WS-Manager in einen Warteschlangenmanager mit mehreren Instanzen konvertiert wird. Alle anderen Schritte sind für einzelne oder mehrere Instanzen von Warteschlangenmanagern mit mehreren Instanzen gemeinsam.

## Vorgehensweise

1. Konfigurieren Sie einen zweiten Server für die Ausführung von IBM MQ for Windows.
   a) Führen Sie die Schritte der Task „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren" auf Seite 558 aus, um einen zweiten Domänenserver zu erstellen. In dieser Folge von Tasks wird der zweite Server als *venus* bezeichnet.

      **Tipp:** Erstellen Sie die zweite Installation. Achten Sie dabei darauf, dass Sie IBM MQ auf beiden Servern mit denselben Einstellungen installieren. Wenn die Standardwerte abweichen, müssen Sie unter Umständen die Variablen `Präfix` und `Installationsname` in der *QMGR* **QueueManager**-Zeilengruppe der IBM MQ-Konfigurationsdatei `mqs.ini` anpassen. Die Variablen beziehen sich auf Pfade, die sich für jede Installation und jeden Warteschlangenmanager auf jedem Server unterscheiden können. Wenn die Pfade auf jedem Server gleich bleiben, ist es einfacher, einen WS-Manager mit mehreren Instanzen zu konfigurieren.
2. Erstellen Sie eine zweite Instanz von *QMGR* unter *venus*.
   a) Wenn *QMGR* auf *mars* nicht vorhanden ist, führen Sie die Task „Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind" auf Seite 565 aus, um sie zu erstellen.

b) Überprüfen Sie, ob die Werte der Präfix- und Installationsname-Parameter für *venus* korrekt sind.

Führen Sie unter *mars*den Befehl **dspmqinf** aus:

```
dspmqinf QMGR
```

Systemantwort:

```
QueueManager:
Name=QMGR
Directory=QMGR
Prefix=C:\ProgramData\IBM\MQ
DataPath=\\sun\wmq\data\QMGR
InstallationName=Installation1
```

c) Kopieren Sie die maschinenlesbare Form der Zeilengruppe **QueueManager** in die Zwischenablage.

Führen Sie unter *mars* den Befehl **dspmqinf** erneut mit dem Parameter `-o command` aus.

```
dspmqinf -o command QMGR
```

Systemantwort:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

d) Führen Sie unter *venus* den Befehl **addmqinf** aus der Zwischenablage heraus aus, um eine Instanz des Warteschlangenmanagers unter *venus*zu erstellen.

Passen Sie den Befehl bei Bedarf an, um die Unterschiede in den Parametern `Präfix` oder `InstallationName` zu berücksichtigen.

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

```
IBM MQ configuration information added.
```

3. Starten Sie den Warteschlangenmanager *QMGR* unter *venus*, um Standby-Instanzen zu ermöglichen.

a) Überprüfen Sie, ob *QMGR* auf *mars* gestoppt wurde.

Führen Sie unter *mars*den Befehl **dspmq** aus:

```
dspmq -m QMGR
```

Die Systemantwort hängt davon ab, wie der WS-Manager gestoppt wurde. Beispiel:

```
C:\Users\Administrator>dspmq -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

b) Führen Sie unter *venus* den Befehl **strmqm** aus, damit *QMGR* die Bereitschaftsdatenbanken zulässt:

```
strmqm -x QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

## Ergebnisse

Führen Sie die folgenden Schritte aus, um den WS-Manager mit mehreren Instanzen zu testen:

1. Führen Sie unter *mars*den Befehl **strmqm** aus, um *QMGR* zu starten, das Bereitschaftsdatenbanken zulässt:

   ```
   strmqm -x QMGR
   ```

   Systemantwort:

   ```
   IBM MQ queue manager 'QMGR' starting.
   The queue manager is associated with installation 'Installation1'.
   A standby instance of queue manager 'QMGR' has been started.
   The active instance is running elsewhere.
   ```

2. Führen Sie unter *venus* den Befehl **endmqm** aus:

   ```
   endmqm -r -s -i QMGR
   ```

   Die Systemantwort auf *venus*:

   ```
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ended, permitting switchover to
   a standby instance.
   ```

   Und unter *mars*:

   ```
   dspmq
   QMNAME(QMGR) STATUS(Running as standby)
   C:\Users\wmquser2>dspmq
   QMNAME(QMGR) STATUS(Running as standby)
   C:\Users\wmquser2>dspmq
   QMNAME(QMGR)  STATUS(Running)
   ```

## Nächste Schritte

Ein Mehrinstanz-WS-Manager kann auch mit Musterprogrammen überprüft werden. Informationen hierzu finden Sie im Abschnitt „Multi-Instanz-Warteschlangenmanager unter Windows überprüfen” auf Seite 576.

Windows *Active Directory- und DNS-Domäne unter Windows erstellen*

Mit dieser Task wird die Domäne *wmq.example.com* auf einem Domänencontroller von Windows 2008 mit dem Namen *sun*erstellt. Er konfiguriert die globale Gruppe `Domain mqm` in der Domäne mit den korrekten Berechtigungen und mit einem Benutzer.

In einer Produktionsmaßstabskonfiguration müssen Sie die Konfiguration möglicherweise an eine vorhandene Domäne anpassen. Sie können z. B. verschiedene Domänengruppen definieren, um unterschiedliche Freigaben zu berechtigen und die Benutzer-IDs zu gruppieren, die Warteschlangenmanager ausführen.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
: Ein Domänencontroller unter Windows Server 2008. Sie ist Eigner der *wmq.example.com*-Domäne, die *Sun*, *mars* und *venus*enthält. Für die Zwecke der Veranschaulichung wird sie auch als Dateiserver verwendet.

***mars***
: Ein Windows Server 2008, der als erster IBM MQ-Server verwendet wird. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***venus***
: Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Die Taskschritte sind konsistent mit einem Windows Server 2008, der installiert ist, jedoch nicht mit Rollen konfiguriert ist. Wenn Sie einen bereits vorhandenen Domänencontroller konfigurieren möchten, ist es unter Umständen hilfreich, die durchzuführenden Schritte an einem neuen Windows Server2008 zu testen. Sie können die Schritte an Ihre Domäne anpassen.

## Informationen zu diesem Vorgang

In dieser Task erstellen Sie eine Active Directory- und DNS-Domäne auf einem neuen Domänencontroller. Anschließend nehmen Sie die erforderliche Konfiguration für die Installation von IBM MQ auf anderen Servern und Workstations vor, die derselben Domäne beitreten. Befolgen Sie die hier beschriebene Vorgehensweise, wenn Sie noch keine Erfahrung mit der Installation und Konfiguration von Active Directory zum Erstellen einer Windows-Domäne haben. Für die Konfiguration eines Multi-Instanz-Warteschlangenmanagers muss eine Windows-Domäne erstellt werden. Ziel dieser Task ist es nicht, eine optimale Konfiguration einer Windows-Domäne zu erzielen. Für die Bereitstellung von Multi-Instanz-Warteschlangenmanagern in einer Produktionsumgebung ist die Windows-Dokumentation zu konsultieren.

Während der Task führen Sie die folgenden Schritte aus:

1. Installieren Sie Active Directory.
2. Fügen Sie eine Domäne hinzu.
3. Fügen Sie die Domäne dem DNS hinzu.
4. Erstellen Sie die globale Gruppe `Domain mqm` und geben Sie ihr die richtigen Berechtigungen.
5. Fügen Sie einen Benutzer hinzu und machen Sie ihn zu einem Mitglied der globalen Gruppe `Domain mqm`.

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550.

Für den Zweck der Aufgabe ist der Hostname des Domänencontrollers *sun* und die beiden IBM MQ-Server werden als *mars* und *venus* bezeichnet. Die Domäne wird als *wmq.example.com* bezeichnet. Sie können alle in Kursivdruck angezeigten Namen in der Task durch Namen Ihrer Wahl ersetzen.

## Vorgehensweise

1. Melden Sie sich als lokaler Administrator oder `Workgroup`-Administrator am Domänencontroller ( *sun*) an.

   Wenn der Server bereits als Domänencontroller konfiguriert ist, müssen Sie sich als Domänenadministrator anmelden.
2. Führen Sie den Assistenten für die Active Directory-Domänenservices aus.

   a) Klicken Sie **Start** > **Ausführen ...** Geben Sie `dcpromo` ein und klicken Sie auf **OK**.

   Wenn die Binärdateien von Active Directory noch nicht installiert sind, installiert Windows die Dateien automatisch.
3. Lassen Sie im ersten Fenster des Assistenten das Kontrollkästchen **Erweiterte Modusinstallation verwenden** leer. Klicken Sie auf **Weiter** > **Weiter** und anschließend auf **Create a new domain in a new forest** > **Weiter** (Neue Domäne in neuer Gesamtstruktur erstellen > Weiter).
4. Geben Sie *wmq.example.com* in das Feld **FQDN of the forest root domain** ein. Klicken Sie auf **Weiter (Next)**.
5. Wählen Sie im Fenster "Set Forest Functional Level" (Funktionsebene der Gesamtstruktur festlegen) **Windows Server 2003** oder höher in der Liste **Forest functional levels** > **Weiter** (Funktionsebenen der Gesamtstruktur > Weiter) aus.

   Die älteste Version von Windows Server, die von IBM MQ unterstützt wird, ist Windows Server 2003.
6. Optional: Wählen Sie im Fenster "Set Domain Functional Level" (Funktionsebene der Domäne festlegen) **Windows Server 2003** oder höher in der Liste **Domain functional levels** > **Weiter** (Funktionsebenen der Domäne > Weiter) aus.

   Dieser Schritt ist nur erforderlich, wenn Sie als Funktionsebene der Gesamtstruktur **Windows Server 2003** festgelegt haben.
7. Das Fenster "Zusätzliche Domänencontrolleroptionen" wird geöffnet, wobei **DNS-Server** als zusätzliche Option ausgewählt ist. Klicken Sie auf **Weiter** und **Ja** , um das Warnfenster zu löschen.

   **Tipp:** Wenn bereits ein DNS-Server installiert ist, wird diese Option nicht angezeigt. Wenn Sie diese Task genau verfolgen möchten, entfernen Sie alle Rollen von diesem Domänencontroller und starten Sie sie erneut.
8. Lassen Sie die Verzeichnisse `Database`, `Log Files` und SYSVOL unverändert. Klicken Sie auf **Weiter**.
9. Geben Sie ein Kennwort in die Felder **Kennwort** und **Kennwort bestätigen** im Fenster 'Administratorkennwort für Directory Services Restore Mode' ein. Klicken Sie auf **Weiter** > **Weiter**. Wählen Sie **Reboot on completion** im letzten Assistentenfenster aus.
10. Wenn der Domänencontroller erneut bootet, melden Sie sich als *wmq*`\Adminstrator` an.

    Der Servermanager wird automatisch gestartet.
11. Öffnen Sie den Ordner *wmq.example.com*`\Users`.

    a) Öffnen Sie **Server Manager** > **Roles** > **Active Directory Domain Services** > ***wmq.example.com*** > **Users** (Servermanager -> Rollen -> Active Directory Domain Services -> wmq.example.com -> Benutzer).
12. Klicken Sie mit der rechten Maustaste auf **Benutzer** > **Neu** > **Gruppe**.

    a) Geben Sie einen Gruppennamen in das Feld **Gruppenname** ein.

    **Anmerkung:** Der bevorzugte Gruppenname ist `Domain mqm`. Geben Sie ihn genau wie angezeigt ein.

    - Durch das Aufrufen der Gruppe `Domain mqm` ändert sich das Verhalten des Prepare IBM MQ Wizarden auf einer Workstation oder einem Server der Domäne. Sie bewirkt, dass Prepare IBM

MQ Wizard die Gruppe Domain mqm in jeder neuen Installation von IBM MQ in der Domäne automatisch zur lokalen mqm -Gruppe hinzufügt.

- Sie können Workstations oder Server in einer Domäne ohne globale Domain mqm-Gruppe installieren. Wenn dies der Fall ist, müssen Sie eine Gruppe mit den gleichen Eigenschaften wie die Gruppe Domain mqm definieren. Sie müssen diese Gruppe oder die Benutzer, die Mitglieder der Gruppe sind, zu Mitgliedern der lokalen mqm-Gruppe machen, je nachdem, wo IBM MQ in einer Domäne installiert ist. Sie können Domänenbenutzer in mehrere Gruppen stellen. Erstellen Sie mehrere Domänengruppen, wobei jede Gruppe einer Gruppe von Installationen entspricht, die Sie separat verwalten möchten. Teilen Sie Domänenbenutzer entsprechend den Installationen, die sie verwalten, in verschiedene Domänengruppen auf. Fügen Sie jede Domänengruppe oder jede Gruppe der lokalen mqm-Gruppe mit unterschiedlichen IBM MQ-Installationen hinzu. Nur Domänenbenutzer in den Domänengruppen, die Mitglieder einer bestimmten lokalen mqm-Gruppe sind, können Warteschlangenmanager für diese Installation erstellen, verwalten und ausführen.
- Der Domänenbenutzer, den Sie bei der Installation von IBM MQ auf einer Workstation oder einem Server in einer Domäne nominieren, muss Mitglied der Gruppe Domain mqm oder einer alternativen Gruppe sein, für die Sie dieselben Eigenschaften wie für die Gruppe Domain mqm definiert haben.

b) Lassen Sie **Global** als **Gruppenbereich** geklickt oder ändern Sie ihn in **Universal** . Lassen Sie **Sicherheit** als **Gruppentyp** geklickt. Klicken Sie auf **OK**.

13. Fügen Sie die Berechtigungen **Zulassen Gruppenzugehörigkeit lesen** und **Allow Read groupMembershipSAM** zu den Rechten der globalen Gruppe Domain mqm hinzu.
    a) Klicken Sie in der Aktionsleiste des Servermanagers auf **View** > **Advanced features** (Ansicht > Erweiterte Funktionen).
    b) Klicken Sie in der Navigationsstruktur des Servermanagers auf **Benutzer**.
    c) Klicken Sie im Fenster "Benutzer" mit der rechten Maustaste auf **Domain mqm** > **Eigenschaften**.
    d) Klicken Sie auf **Sicherheit** > **Erweitert** > **Hinzufügen...**. Geben Sie Domain mqm ein und klicken Sie auf **Check names** > **OK** (Namen prüfen > OK).

       Das Feld **Name** wird mit der Zeichenfolge Domain mqm (*domain name*\Domain mqm)ausgefüllt.
    e) Klicken Sie auf **Eigenschaften** . Wählen Sie in der Liste **Apply to** die Option **Descendant User Objects** aus.
    f) Markieren Sie in der Liste **Permissions** (Berechtigungen) die Kontrollkästchen **Read group membership** (Gruppenzugehörigkeit lesen) und **Read groupMembershipSAM Zulassen**; klicken Sie auf **OK** > **Anwenden** > **OK** > **OK**.
14. Fügen Sie zwei oder mehr Benutzer zur globalen Gruppe Domain mqm hinzu.

    Ein Benutzer, im Beispiel *wmquser1* , führt den IBM MQ -Service aus und der andere Benutzer, *wmquser2* , wird interaktiv verwendet.
    Ein Domänenbenutzer ist für die Erstellung eines Warteschlangenmanagers erforderlich, der die alternative Sicherheitsgruppe in einer Domänenkonfiguration verwendet. Es reicht nicht aus, wenn die Benutzer-ID ein Administrator ist, obwohl ein Administrator über die Berechtigung zum Ausführen des Befehls **crtmqm** verfügt. Der Domänenbenutzer, der ein Administrator sein kann, muss ein Mitglied der lokalen mqm -Gruppe sowie der alternativen Sicherheitsgruppe sein.
    In diesem Beispiel nehmen Sie *wmquser1* -und *wmquser2* -Member der globalen Gruppe Domain mqm ab. Der Prepare IBM MQ Wizard konfiguriert Domain mqm automatisch als Mitglied der lokalen Gruppe mqm, unabhängig davon, wo der Assistent ausgeführt wird.
    Für jede Installation von IBM MQ auf einem einzelnen Computer müssen Sie für die Ausführung des IBM MQ-Service einen anderen Benutzer angeben. Sie können dieselben Benutzer auf verschiedenen Computern verwenden.

    a) Klicken Sie in der Navigationsstruktur des Servermanagers auf **Benutzer** > **Neu** > **Benutzer**.

b) Geben Sie im Fenster 'Neues Objekt-Benutzer' *wmquser1* in das Feld **Benutzeranmeldename** ein. Geben Sie *WebSphere* in das Feld **Vorname** und *MQ1* in das Feld **Nachname** ein. Klicken Sie auf **Weiter**.

c) Geben Sie ein Kennwort in die Felder **Kennwort** und **Kennwort bestätigen** ein, und inaktivieren Sie das Kontrollkästchen **Benutzer muss Kennwort bei der nächsten Anmeldung ändern** . Klicken Sie auf **Weiter** > **Fertigstellen**.

d) Klicken Sie im Fenster "Benutzer" mit der rechten Maustaste auf ***WebSphere MQ*** > **Zu Gruppe hinzufügen...**. Geben Sie `Domain mqm` ein und klicken Sie auf **Namen prüfen** > **OK** > **OK**.

e) Wiederholen Sie die Schritte a bis d , um *WebSphere MQ2* als *wmquser2* hinzuzufügen.

15. Ausführung von IBM MQ als Service.

Wenn Sie IBM MQ als Service ausführen und dem Domänenbenutzer, den Sie von Ihrem Domänenadministrator erhalten haben, danach Zugriff für die Ausführung als Service erteilen müssen, führen Sie die folgenden Schritte aus:

a) Klicken Sie auf **Start> Ausführen...**.

Geben Sie den Befehl `secpol.msc` ein und klicken Sie auf **OK** .

b) Öffnen Sie **Sicherheitseinstellungen > Local Policies** (Lokale Richtlinien) **> User Rights Assignments** (Zuweisen von Benutzerrechten).

Klicken Sie in der Liste der Richtlinien mit der rechten Maustaste auf **Log on as a service** (Anmelden als Service) **> Eigenschaften**.

c) Klicken Sie auf **Add User or Group** (Benutzer oder Gruppe hinzufügen).

Geben Sie den Namen des Benutzers ein, den Sie von Ihrem Domänenadministrator erhalten haben, und klicken Sie auf **Check Names** (Namen überprüfen).

d) Wenn Sie in einem Windows-Sicherheitsfenster dazu aufgefordert werden, geben Sie den Benutzernamen und das Kennwort eines Accountbenutzers oder eines Administrators mit ausreichender Berechtigung ein, und klicken Sie auf **OK > Anwenden > OK**.

Schließen Sie das Fenster "Lokale Sicherheitsrichtlinie".

**Anmerkung:** Unter Windows Server 2008 und Windows Server 2012 ist die Benutzerkontosteuerung (UAC) standardmäßig aktiviert.

Diese Funktion schränkt die Operationen ein, die Benutzer (selbst wenn sie zur Gruppe 'Administratoren' gehören) für bestimmte Komponenten des Betriebssystems ausführen können. Zur Umgehung dieser Einschränkung müssen Sie bestimmte Schritte ausführen.

## Nächste Schritte

Fahren Sie mit der nächsten Task, „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren” auf Seite 558, fort.

**Zugehörige Tasks**

Windows IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren

Windows Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows

Windows Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind

Windows *IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren*

In dieser Task installieren und konfigurieren Sie IBM MQ auf einem Server oder einer Workstation in der *wmq.example.com* Windows-Domäne.

In einer Produktionsmaßstabskonfiguration müssen Sie die Konfiguration möglicherweise an eine vorhandene Domäne anpassen. Sie können z. B. verschiedene Domänengruppen definieren, um unterschiedliche Freigaben zu berechtigen und die Benutzer-IDs zu gruppieren, die Warteschlangenmanager ausführen.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
Ein Domänencontroller unter Windows Server 2008. Sie ist Eigner der *wmq.example.com*-Domäne, die *Sun*, *mars* und *venus*enthält. Für die Zwecke der Veranschaulichung wird sie auch als Dateiserver verwendet.

***mars***
Ein Windows Server 2008, der als erster IBM MQ-Server verwendet wird. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***venus***
Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

**Wichtig:** Computer, die mit Windows 10 Version 1607 und Windows Server 2016 beginnen, sind standardmäßig restriktiver als frühere Versionen von Windows.

Durch diese Änderung können Clients, die remote Aufrufe an den Sicherheitskontenmanager (SAM) vornehmen dürfen, eingeschränkt werden, und sie können sich auf IBM MQ auswirken, wenn die Warteschlangenmanager nicht gestartet werden. Der Zugriff auf SAM ist für das Funktionieren von IBM MQ kritisch, wenn IBM MQ als Domänenkonto konfiguriert ist.

1. Führen Sie die Schritte in „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 aus, um einen Domänencontroller, *sun*, für die Domäne *wmq.example.com* zu erstellen. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.
2. Im Abschnitt Hardware- und Softwarevoraussetzungen auf Windows-Systemen finden Sie Informationen zu weiteren Windows-Versionen, auf denen IBM MQ ausgeführt werden kann.

## Informationen zu diesem Vorgang

In dieser Task konfigurieren Sie einen Windows Server 2008 mit dem Namen *mars* als Mitglied der *wmq.example.com*-Domäne. Sie installieren IBM MQ und konfigurieren die Installation so, dass sie als Mitglied der *wmq.example.com*-Domäne ausgeführt wird.

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550.

Für den Zweck der Aufgabe ist der Hostname des Domänencontrollers *sun* und die beiden IBM MQ-Server werden als *mars* und *venus* bezeichnet. Die Domäne wird als *wmq.example.com* bezeichnet. Sie können alle in Kursivdruck angezeigten Namen in der Task durch Namen Ihrer Wahl ersetzen.

## Vorgehensweise

1. Fügen Sie den Domänencontroller *sun.wmq.example.com* zu *mars* als DNS-Server hinzu.
   a) Melden Sie sich unter *mars* als *mars*\Administrator an und klicken Sie auf **Start**.
   b) Klicken Sie mit der rechten Maustaste auf **Network** > **Properties** > **Manage network connections** (Netz > Eigenschaften > Netzverbindungen verwalten).
   c) Klicken Sie auf den Netzadapter, und klicken Sie auf **Eigenschaften** .

      Das System antwortet mit dem Fenster "Merkmale des lokalen Bereichs", in dem die von der Verbindung verwendeten Elemente aufgelistet werden.
   d) Wählen Sie **Internet Protocol Version 4** oder **Internet Protocol IBM WebSphere MQ 6** aus der Liste der Elemente im Fenster "Local Area Connection Properties" (Eigenschaften der Verbindung

im lokalen Netz) aus. Klicken Sie auf **Eigenschaften** > **Erweitert ...** und klicken Sie auf die Registerkarte **DNS** .

e) Klicken Sie unter den DNS-Serveradressen auf **Hinzufügen ...** .

f) Geben Sie die IP-Adresse des Domänencontrollers ein, bei dem es sich auch um den DNS-Server handelt, und klicken Sie auf **Hinzufügen** .

g) Klicken Sie auf **Diese DNS-Suffixe anhängen** > **Hinzufügen ...**.

h) Geben Sie *wmq.example.com* ein und klicken Sie auf **Hinzufügen**

i) Geben Sie *wmq.example.com* in das Feld **DNS-Suffix für diese Verbindung** ein.

j) Wählen Sie die Option **Diese Verbindungsadresse in DNS registrieren** und **Das Suffix dieser Verbindung in der DNS-Registrierung verwenden** aus. Klicken Sie auf **OK** > **OK** > **Schließen**.

k) Öffnen Sie ein Befehlsfenster und geben Sie den Befehl **ipconfig /all** ein, um die TCP/IP-Einstellungen zu überprüfen.

2. Fügen Sie unter *mars* den Computer zur Domäne *wmq.example.com* hinzu.

a) Klicken Sie auf **Start**.

b) Klicken Sie mit der rechten Maustaste auf **Computer** > **Eigenschaften**. Klicken Sie im Bereich "Computername", "Domäne" und "Arbeitsgruppeneinstellungen" auf **Einstellungen ändern** .

c) Klicken Sie in den Fenstern "Systemeigenschaften" auf **Ändern ...** .

d) Klicken Sie auf Domäne, geben Sie *wmq.example.com* ein und klicken Sie auf **OK**.

e) Geben Sie den **Benutzernamen** und das **Kennwort** des Domänencontrolleradministrators ein, der über die Berechtigung verfügt, dass der Computer der Domäne beitreten kann, und klicken Sie auf **OK** .

f) Klicken Sie auf **OK** > **OK** > **Schließen** > **Jetzt erneut starten** als Antwort auf die "Willkommen bei der *wmq.example.com*-Domäne"-Nachricht.

3. Überprüfen Sie, ob der Computer Mitglied der *wmq.example.com*-Domäne ist.

a) Melden Sie sich unter *sun* an dem Domänencontroller als *wmq*\Administrator an.

b) Öffnen Sie **Servermanager** > **Active Directory-Domänenservices** > ***wmq.example.com*** > **Computer** und überprüfen Sie, ob *mars* im Fenster "Computer" korrekt aufgelistet ist.

4. Installieren Sie IBM MQ for Windows unter *mars*.

Weitere Informationen zur Ausführung des IBM MQ for Windows-Installationsassistenten finden Sie im Abschnitt zur Installation von IBM MQ-Server unter Windows.

a) Melden Sie sich unter *mars* als lokaler Administrator an, *mars*\Administrator.

b) Führen Sie den Befehl **Setup** von den IBM MQ for Windows-Installationsmedien aus.

Die Launchpadanwendung von IBM MQ wird gestartet.

c) Klicken Sie auf **Softwarevoraussetzungen** , um zu überprüfen, ob die vorausgesetzte Software installiert ist.

d) Klicken Sie auf **Netzwerkkonfiguration** > **Ja**, um eine Domänenbenutzer-ID zu konfigurieren.

Bei der Task „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 wird für diese Gruppe von Tasks eine Domänenbenutzer-ID konfiguriert.

e) Klicken Sie auf **IBM MQ-Installation**, wählen Sie eine Installationssprache aus und klicken Sie auf "Launch IBM MQ Installer" (IBM MQ-Installationsprogramm starten).

f) Bestätigen Sie die Lizenzvereinbarung und klicken Sie auf **Weiter** > **Weiter** > **Installieren**, um die Standardkonfiguration zu akzeptieren. Warten Sie, bis die Installation abgeschlossen ist, und klicken Sie auf **Fertig stellen** .

Möglicherweise möchten Sie den Namen der Installation ändern, verschiedene Komponenten installieren, ein anderes Verzeichnis für WS-Manager-Daten und -Protokolle konfigurieren oder in einem anderen Verzeichnis installieren. Wenn dies der Fall ist, klicken Sie auf **Angepasst** und nicht auf **Standard** .

IBM MQ wird installiert und der Prepare IBM MQ Wizard wird vom Installationsprogramm gestartet.

**Wichtig:** Führen Sie den Assistenten noch nicht aus.

5. Konfigurieren Sie den Benutzer, der den IBM MQ-Service ausführen wird, mit dem Recht **Run as a service** (Als Service ausführen).

   Wählen Sie aus, ob die lokale Gruppe `mqm`, die Gruppe `Domain mqm` oder der Benutzer konfiguriert werden soll, der den IBM MQ-Service mit dem Recht ausführen wird. In dem Beispiel geben Sie dem Benutzer das Recht.

   a) Klicken Sie auf **Starten** > **Ausführen**. Geben Sie den Befehl **`secpol.msc`** ein und klicken Sie auf **OK**.

   b) Öffnen Sie **Sicherheitseinstellungen** > **Lokale Richtlinien** > **Zuweisen von Benutzerrechten**. Klicken Sie in der Liste der Richtlinien mit der rechten Maustaste auf **Log on as a service** > **Properties** (Als Dienst anmelden > Eigenschaften).

   c) Klicken Sie auf **Add User or Group** (Benutzer oder Gruppe hinzufügen). Geben Sie *`wmquser1`* ein und klicken Sie auf **Namen überprüfen** .

   d) Geben Sie den Benutzernamen und das Kennwort eines Domänenadministrators, *`wmq`*`\Administrator`, ein und klicken Sie auf **OK** > **Anwenden** > **OK**. Schließen Sie das Fenster "Lokale Sicherheitsrichtlinie".

6. Führen Sie die Prepare IBM MQ Wizard aus.

   Weitere Informationen finden Sie im Abschnitt IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren.

   a) Das IBM MQ-Installationsprogramm führt den Prepare IBM MQ Wizard automatisch aus.

   Für den manuellen Start des Assistenten suchen Sie den Direktaufruf für den Prepare IBM MQ Wizard im Ordner **Start** > **Alle Programme** > **IBM MQ**. Wählen Sie in einer Konfiguration mit mehreren Installationen den entsprechenden Direktaufruf für die IBM MQ-Installation aus.

   b) Klicken Sie auf **Weiter** und lassen Sie das **Ja** als Antwort auf die Frage "Identify if there is a Windows 2000 or later domain controller in the network?" (Ermitteln, ob ein Domänencontroller von Windows 2000 oder höher im Netz vorhanden ist?) angeklickt.

   c) Klicken Sie im ersten Fenster " IBM MQ for Windows für Windows -Domänenbenutzer konfigurieren" auf **Ja** > **Weiter** .

   d) Geben Sie im zweiten Fenster " IBM MQ for Windows für Windows -Domänenbenutzer konfigurieren" im Feld **Domäne** *`wmq`* ein. Geben Sie *`wmquser1`* in das Feld **Benutzername** und das Kennwort (falls Sie eines festgelegt haben) in das Feld **Kennwort** ein. Klicken Sie auf **Weiter**.

   Der Assistent konfiguriert und startet IBM MQ mit *wmquser1*.

   e) Wählen Sie auf der letzten Seite des Assistenten die Markierungsfelder aus, die Sie benötigen, und klicken Sie auf **Fertig stellen** .

## Nächste Schritte

1. Führen Sie die Task „Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind” auf Seite 584 aus, um zu prüfen, ob die Installation und Konfiguration ordnungsgemäß funktionieren.
2. Führen Sie die Task „Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows” auf Seite 562 aus, um eine Dateifreigabe zum Speichern der Daten- und Protokolldateien eines Multi-Instanz-Warteschlangenmanagers zu konfigurieren.

**Zugehörige Tasks**

Windows Active Directory- und DNS-Domäne unter Windows erstellen

Windows Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows

Windows Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind

**Zugehörige Verweise**

Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service

Windows *Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows*

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind.

In einer Produktionsmaßstabskonfiguration müssen Sie die Konfiguration möglicherweise an eine vorhandene Domäne anpassen. Sie können z. B. verschiedene Domänengruppen definieren, um unterschiedliche Freigaben zu berechtigen und die Benutzer-IDs zu gruppieren, die Warteschlangenmanager ausführen.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
Ein Domänencontroller unter Windows Server 2008. Sie ist Eigner der *wmq.example.com*-Domäne, die *Sun*, *mars* und *venus*enthält. Für die Zwecke der Veranschaulichung wird sie auch als Dateiserver verwendet.

***mars***
Ein Windows Server 2008, der als erster IBM MQ-Server verwendet wird. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***venus***
Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Damit diese Task genau wie dokumentiert ausgeführt wird, führen Sie die Schritte in der Task „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 aus, um die Domäne *sun.wmq.example.com* auf dem Domänencontroller *sun* zu erstellen. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.

## Informationen zu diesem Vorgang

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550.

In der Task erstellen Sie eine Freigabe, die ein Daten-und Protokollverzeichnis enthält, und eine globale Gruppe, um den Zugriff auf die Freigabe zu berechtigen. Sie übergeben den Namen der globalen Gruppe, die die Freigabe für den Befehl **crtmqm** im Parameter `-a` berechtigt. Die globale Gruppe bietet Ihnen die Flexibilität, die Benutzer dieses Anteils von den Benutzern anderer Aktien zu trennen. Wenn Sie diese Flexibilität nicht benötigen, genehmigen Sie die gemeinsame Nutzung mit der Gruppe `Domain mqm`, anstatt eine neue globale Gruppe zu erstellen.

Die globale Gruppe, die für die gemeinsame Nutzung in dieser Task verwendet wird, wird als *wmqha* bezeichnet, und die gemeinsam genutzte Gruppe wird als *wmq* bezeichnet. Sie werden auf dem Domänencontroller *sun* in der Windows-Domäne *wmq.example.com* definiert. Die gemeinsam verwendete Gruppe verfügt über vollständige Steuerberechtigungen für die globale Gruppe *wmqha*. Ersetzen Sie die italicisierten Namen in der Task durch die Namen Ihrer Wahl.

Für die Zwecke dieser Task ist der Domänencontroller derselbe Server wie der Dateiserver. Teilen Sie in praktischen Anwendungen die Verzeichnis-und Dateiservices zwischen verschiedenen Servern auf Leistung und Verfügbarkeit auf.

Sie müssen die Benutzer-ID, unter der der WS-Manager ausgeführt wird, als Mitglied von zwei Gruppen konfigurieren. Es muss ein Mitglied der lokalen mqm-Gruppe auf einem IBM MQ-Server und der globalen Gruppe von *wmqha* sein.

Wenn in dieser Gruppe von Tasks der Warteschlangenmanager als Service ausgeführt wird, wird er unter der Benutzer-ID *wmquser1* ausgeführt, daher muss *wmquser1* ein Mitglied von *wmqha* sein. Wenn der Warteschlangenmanager interaktiv ausgeführt wird, wird er unter der Benutzer-ID *wmquser2* ausgeführt, daher muss *wmquser2* ein Mitglied von *wmqha* sein. Sowohl *wmquser1* als auch *wmquser2* sind Mitglieder der globalen Gruppe `Domain mqm`. `Domain mqm` ist ein Mitglied der lokalen `mqm` -Gruppe auf den *mars* -und *venus* IBM MQ -Servern. Daher sind *wmquser1* und *wmquser2* Mitglieder der lokalen `mqm`-Gruppe auf beiden IBM MQ-Servern.

## Vorgehensweise

1. Melden Sie sich bei dem Domänencontroller *sun.wmq.example.com* als Domänenadministrator an.
2. Erstellen Sie die globale Gruppe *wmqha*.
   a) Öffnen Sie **Server Manager** > **Roles** > **Active Directory Domain Services** > ***wmq.example.com*** > **Users** (Servermanager -> Rollen -> Active Directory Domain Services -> wmq.example.com -> Benutzer).
   b) Öffnen Sie den Ordner *wmq.example.com*\Users.
   c) Klicken Sie mit der rechten Maustaste auf **Benutzer** > **Neu** > **Gruppe**.
   d) Geben Sie *wmqha* in das Feld **Gruppenname** ein.
   e) Lassen Sie **Global** als **Gruppenbereich** und **Sicherheit** als **Gruppentyp** geklickt. Klicken Sie auf **OK**.
3. Fügen Sie die Domänenbenutzer *wmquser1* und *wmquser2* zur globalen Gruppe *wmqha* hinzu.
   a) Klicken Sie in der Navigationsstruktur des Servermanagers auf **Benutzer** und klicken Sie mit der rechten Maustaste auf ***wmqha*** > **Eigenschaften** in der Liste der Benutzer.
   b) Klicken Sie im Fenster *wmqha* Eigenschaften auf die Registerkarte "Mitglieder".
   c) Klicken Sie auf **Hinzufügen ...** . *wmquser1 ; wmquser2* eingeben und auf **Namen prüfen** > **OK** > **Anwenden** > **OK**klicken.
4. Erstellen Sie die Verzeichnisstruktur, in der die Daten und Protokolldateien des Warteschlangenmanagers enthalten sind.
   a) Öffnen Sie eine Eingabeaufforderung.
   b) Geben Sie den folgenden Befehl ein:

```
md c:\wmq\data, c:\wmq\logs
```

5. Berechtigen Sie die globale Gruppe *wmqha*, die vollständige Steuerberechtigung für die *c:\wmq*-Verzeichnisse und die gemeinsame Nutzung zu erhalten.
   a) Klicken Sie in Windows Explorer mit der rechten Maustaste auf ***c:\wmq*** > **Eigenschaften**.
   b) Klicken Sie auf die Registerkarte **Sicherheit** und klicken Sie auf **Erweitert** > **Bearbeiten...**.
   c) Wählen Sie das Kontrollkästchen **Vererbbare Berechtigungen für diesen Objekteigner einschließen** ab. Klicken Sie im Fenster " Windows -Sicherheit" auf **Kopieren** .
   d) Wählen Sie die Zeilen für Benutzer in der Liste der **Berechtigungseinträge** aus und klicken Sie auf **Entfernen** . Übernehmen Sie die Zeilen für SYSTEM, Administratoren und CREATOR OWNER in der Liste der **Berechtigungseinträge** .
   e) Klicken Sie auf **Hinzufügen ...**. und geben Sie den Namen der globalen Gruppe *wmqha*ein. Klicken Sie auf **Namen überprüfen** > **OK**.
   f) Wählen Sie im Fenster "Berechtigungseintrag für wmq" die Option **Vollständige Kontrolle** in der Liste der **Berechtigungen** aus.
   g) Klicken Sie auf **OK** > **Anwenden** > **OK** > **OK** > **OK**

h) Klicken Sie in Windows Explorer mit der rechten Maustaste auf ***c:\wmq*** > **Share...** (Gemeinsam nutzen).

i) Klicken Sie auf **Erweiterte gemeinsame Nutzung** . und wählen Sie das Kontrollkästchen **Diesen Ordner freigeben** aus. Belassen Sie den gemeinsam verwendeten Namen auf *wmq*.

j) Klicken Sie auf **Berechtigungen** > **Hinzufügen ...**, und geben Sie den Namen der globalen Gruppe *wmqha*ein. Klicken Sie auf **Namen überprüfen** > **OK**.

k) Wählen Sie *wmqha* in der Liste **Gruppen- oder Benutzernamen** aus. Wählen Sie das Kontrollkästchen **Vollständige Steuerung** in der Liste der **Berechtigungen für *wmqha*** aus. Klicken Sie auf **Anwenden**.

l) Wählen Sie *Administrators* in der Liste **Gruppen- oder Benutzernamen** aus. Wählen Sie das Kontrollkästchen **Vollständige Steuerung** in der Liste der **Berechtigungen für *Administrators*** aus. Klicken Sie auf **Anwenden** > **OK** > **OK** > **Schließen**.

## Nächste Schritte

Überprüfen Sie, ob Sie Dateien von jedem der IBM MQ-Server lesen und in die gemeinsam genutzten Verzeichnisse schreiben können. Überprüfen Sie die IBM MQ-Servicebenutzer-ID, *wmquser1* und die interaktive Benutzer-ID *wmquser2*.

1. Wenn Sie Remote Desktop verwenden, müssen Sie *wmq**wmquser1* und *wmquser2* der lokalen Gruppe `Remote Desktop Users` unter *mars*hinzufügen.
   a. Melden Sie sich bei *mars* als *wmq*\`Administrator` an.
   b. Führen Sie den Befehl **`lusrmgr.msc`** aus, um das Fenster "Lokale Benutzer und Gruppen" zu öffnen.
   c. Klicken Sie auf **Gruppen** . Klicken Sie mit der rechten Maustaste auf **Remote Desktop Users** > **Properties** > **Add...** (Remote-Desktop-Benutzer -> Eigenschaften -> Hinzufügen...). Geben Sie *wmquser1 ; wmquser2* ein und klicken Sie auf **Namen überprüfen**.
   d. Geben Sie den Benutzernamen und das Kennwort des Domänenadministrators *wmq*\`Administrator`ein und klicken Sie auf **OK** > **Anwenden** > **OK**.
   e. Schließen Sie das Fenster 'Lokale Benutzer und Gruppen'.
2. Melden Sie sich bei *mars* als *wmq**wmquser1* an.
   a. Öffnen Sie ein Windows-Explorer-Fenster und geben Sie *sun**wmq* ein.

      Das System antwortet, indem es den gemeinsam verwendeten *wmq*-Anteil auf *sun.wmq.example.com* öffnet, und listet die Daten- und Protokollverzeichnisse auf.
   b. Überprüfen Sie die Berechtigungen von *wmquser1*, indem Sie eine Datei im Datenunterverzeichnis erstellen, einige Inhalte hinzufügen, lesen und anschließend löschen.
3. Melden Sie sich bei *mars* als *wmq**wmquser2* an, und wiederholen Sie die Prüfungen.
4. Führen Sie die nächste Task aus, um einen Warteschlangenmanager zu erstellen, der die gemeinsam genutzten Daten- und Protokollverzeichnisse verwendet (siehe „Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind” auf Seite 565).

**Zugehörige Tasks**

Windows Active Directory- und DNS-Domäne unter Windows erstellen

Windows IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren

Windows Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind

Windows *Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind*

Diese Task zeigt, wie das Flag -a im **crtmqm** -Befehl verwendet wird. Das Flag -a gibt dem Warteschlangenmanager Zugriff auf seine Protokoll-und Datendateien in einer fernen Dateifreigabe unter Verwendung der alternativen Sicherheitsgruppe.

In einer Produktionsmaßstabskonfiguration müssen Sie die Konfiguration möglicherweise an eine vorhandene Domäne anpassen. Sie können z. B. verschiedene Domänengruppen definieren, um unterschiedliche Freigaben zu berechtigen und die Benutzer-IDs zu gruppieren, die Warteschlangenmanager ausführen.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
Ein Domänencontroller unter Windows Server 2008. Sie ist Eigner der *wmq.example.com*-Domäne, die *Sun*, *mars* und *venus*enthält. Für die Zwecke der Veranschaulichung wird sie auch als Dateiserver verwendet.

***mars***
Ein Windows Server 2008, der als erster IBM MQ-Server verwendet wird. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***venus***
Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

Führen Sie die Schritte in den folgenden Tasks aus. Die Tasks erstellen den Domänencontroller und die Domäne, installieren IBM MQ for Windows auf einem Server und erstellen den Dateifreigabewert für Daten und Protokolldateien. Wenn Sie einen bereits vorhandenen Domänencontroller konfigurieren möchten, ist es unter Umständen hilfreich, die durchzuführenden Schritte an einem neuen Windows Server2008 zu testen. Sie können die Schritte an Ihre Domäne anpassen.

1. „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555.
2. „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren” auf Seite 558.
3. „Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows” auf Seite 562.

## Informationen zu diesem Vorgang

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550.

In dieser Task erstellen Sie einen Warteschlangenmanager, der seine Daten speichert und sich in einem fernen Verzeichnis auf einem Dateiserver anmeldet. Für die Zwecke dieses Beispiels ist der Dateiserver der gleiche Server wie der Domänencontroller. Das Verzeichnis, das die Daten-und Protokollordner enthält, wird mit der vollständigen Steuerberechtigung für die globale Gruppe wmqha gemeinsam genutzt.

## Vorgehensweise

1. Melden Sie sich am Domänenserver *mars* als lokaler Administrator *mars*\Administrator an.
2. Öffnen Sie ein Befehlsfenster.
3. Starten Sie den IBM MQ-Service erneut.

Sie müssen den Service erneut starten, damit die Benutzer-ID, unter der er ausgeführt wird, die zusätzlichen Sicherheitsberechtigungsnachweise erhält, die Sie für die Benutzer-ID konfiguriert haben.

Geben Sie die Befehle ein:

```
endmqsvc
strmqsvc
```

Die Systemantworten:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' ended successfully.
```

Und:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' started successfully.
```

4. Erstellen Sie den Warteschlangenmanager.

```
crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs
QMGR
```

Sie müssen die Domäne *wmq* der alternativen Sicherheitsgruppe *wmqha* angeben, indem Sie den vollständigen Domänennamen der globalen Gruppe *"wmq\wmqha"* angeben.

Sie müssen den UNC-Namen (Universal Naming Convention) für den gemeinsam verwendeten *sun**wmq* festlegen und keine zugeordnete Laufwerkreferenz verwenden.

Systemantwort:

```
IBM MQ queue manager created.
Directory '\\sun\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

## Nächste Schritte

Testen Sie den Warteschlangenmanager, indem Sie eine Nachricht in eine Warteschlange einreihen und eine Nachricht erhalten.

1. Starten Sie den Warteschlangenmanager.

```
strmqm QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Erstellen Sie eine Testwarteschlange.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Systemantwort:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. Reihen Sie eine Testnachricht mit dem Beispielprogramm **amqsput**ein.

```
echo 'A test message' | amqsput QTEST QMGR
```

Systemantwort:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. Rufen Sie die Testnachricht mit dem Beispielprogramm **amqsget**ab.

```
amqsget QTEST QMGR
```

Systemantwort:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. Stoppen Sie den Warteschlangenmanager.

```
endmqm -i QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. Löschen Sie den Warteschlangenmanager.

```
dltmqm QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Löschen Sie die von Ihnen erstellten Verzeichnisse.

**Tipp:** Fügen Sie die Option /Q zu den Befehlen hinzu, um zu verhindern, dass der Befehl zum Löschen jeder Datei oder jedes Verzeichnisses auffordert.

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

**Zugehörige Tasks**

Windows Active Directory- und DNS-Domäne unter Windows erstellen

Windows IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren

Windows Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows

Windows *WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen*

Im folgenden Beispiel wird gezeigt, wie ein Warteschlangenmanager mit mehreren Instanzen unter Windows auf Domänencontrollern konfiguriert wird. In der Konfiguration werden die verwendeten Konzepte und nicht die Produktionsmaßstab, sondern die Konzepte veranschaulicht. Das Beispiel basiert auf Windows Server 2008. Möglicherweise weichen die Schritte auf anderen Windows Server-Versionen im Einzelnen ab.

Die Konfiguration verwendet das Konzept einer Mini-Domäne oder "domainlet"; siehe Windows 2000-, Windows Server 2003- und Windows Server 2008-Clusterknoten als Domänencontroller. Informationen zum Hinzufügen von Warteschlangenmanagern mit mehreren Instanzen zu einer vorhandenen Domäne finden Sie unter „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen” auf Seite 551.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
Ein Server von Windows Server 2008, der als erster Domänencontroller verwendet wird. Sie definiert die *wmq.example.com*-Domäne, die *sun*, *earth* und *mars*enthält. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***earth***
Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***mars***
Ein Windows Server 2008, der als Dateiserver verwendet wird.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Unter Windows muss das Dateisystem, auf dem die Warteschlangenmanager-Daten und Protokolldateien gespeichert werden sollen, nicht überprüft werden. Die Prüfprozedur Verhalten des gemeinsam genutzten Dateisystems überprüfen ist auf AIX and Linux anwendbar. Unter Windows werden die Prüfungen immer mit positivem Ergebnis abgeschlossen.
2. Führen Sie die Schritte in „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 aus, um den ersten Domänencontroller zu erstellen.
3. Führen Sie die Schritte in „Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne” auf Seite 572 aus, um einen zweiten Domänencontroller hinzuzufügen, IBM MQ for Windows auf beiden Domänencontrollern zu installieren und die Installationen zu überprüfen.
4. Führen Sie die Schritte in „IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren” auf Seite 574 aus, um IBM MQ auf den beiden Domänencontrollern zu installieren.

## Informationen zu diesem Vorgang

Erstellen Sie auf einem Dateiserver in derselben Domäne eine Freigabe für die Protokoll-und Datenverzeichnisse des Warteschlangenmanagers. Erstellen Sie als Nächstes die erste Instanz eines Multi-Instanz-Warteschlangenmanagers, der die Dateifreigabe auf einem der Domänencontroller verwendet. Erstellen Sie die andere Instanz auf dem anderen Domänencontroller, und überprüfen Sie abschließend die Konfiguration. Sie können die Dateifreigabe auf einem Domänencontroller erstellen.

In der Stichprobe ist *sun* der erste Domänencontroller, *earth* der zweite und *mars* der Dateiserver.

## Vorgehensweise

1. Erstellen Sie die Verzeichnisse, die die WS-Manager-Daten und -Protokolldateien enthalten sollen.
   a) Geben Sie unter *mars* den folgenden Befehl ein:

   ```
   md c:\wmq\data , c:\wmq\logs
   ```

2. Geben Sie die Verzeichnisse frei, die die Daten des Warteschlangenmanagers und die Protokolldateien enthalten sollen.

   Sie müssen den vollständigen Steuerzugriff auf die lokale Domänengruppe `mqm` und die Benutzer-ID, die Sie zum Erstellen des Warteschlangenmanagers verwenden, zulassen. In dem Beispiel haben Benutzer-IDs, die Mitglieder von `Domain Administrators` sind, die Berechtigung zum Erstellen von Warteschlangenmanagern.

   Die Dateifreigabe muss sich auf einem Server befinden, der sich in derselben Domäne wie die Domänencontroller befindet. In dem Beispiel befindet sich der Server *mars* in derselben Domäne wie die Domänencontroller.

   a) Klicken Sie in Windows Explorer mit der rechten Maustaste auf ***c:\wmq*** > **Eigenschaften**.
   b) Klicken Sie auf die Registerkarte **Sicherheit** und klicken Sie auf **Erweitert** > **Bearbeiten...**.
   c) Wählen Sie das Kontrollkästchen **Vererbbare Berechtigungen für diesen Objekteigner einschließen** ab. Klicken Sie im Fenster " Windows -Sicherheit" auf **Kopieren** .
   d) Wählen Sie die Zeilen für Benutzer in der Liste der **Berechtigungseinträge** aus und klicken Sie auf **Entfernen** . Übernehmen Sie die Zeilen für SYSTEM, Administratoren und CREATOR OWNER in der Liste der **Berechtigungseinträge** .
   e) Klicken Sie auf **Hinzufügen ...**. und geben Sie den Namen der lokalen Domänengruppe *mqm*ein. Klicken Sie auf **Namen überprüfen**.
   f) Geben Sie als Antwort auf ein Sicherheitsfenster von Windows den Namen und das Kennwort des `Domain Administrator` ein und klicken Sie auf **OK** > **OK**.
   g) Wählen Sie im Fenster "Berechtigungseintrag für `wmq`" die Option **Vollständige Kontrolle** in der Liste der **Berechtigungen** aus.
   h) Klicken Sie auf **OK** > **Anwenden** > **OK** > **OK** > **OK**
   i) Wiederholen Sie die Schritte e bis h, um `Domain Administrators`hinzuzufügen.
   j) Klicken Sie in Windows Explorer mit der rechten Maustaste auf ***c:\wmq*** > **Share...** (Gemeinsam nutzen).
   k) Klicken Sie auf **Erweiterte gemeinsame Nutzung** . und wählen Sie das Kontrollkästchen **Diesen Ordner freigeben** aus. Belassen Sie den gemeinsam verwendeten Namen auf *wmq*.
   l) Klicken Sie auf **Berechtigungen** > **Hinzufügen ...**, und geben Sie den Namen der lokalen Domänengruppe *mqm* `; Domain Administrators`ein. Klicken Sie auf **Namen überprüfen** .
   m) Geben Sie als Antwort auf ein Sicherheitsfenster von Windows den Namen und das Kennwort des `Domain Administrator` ein und klicken Sie auf **OK** > **OK**.

3. Erstellen Sie den Warteschlangenmanager *QMGR* auf dem ersten Domänencontroller, *sun*.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

Systemantwort:

```
IBM MQ queue manager created.
Directory '\\mars\wmq\data\QMGR' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMGR'.
Default objects statistics : 74 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. Starten Sie den Warteschlangenmanager unter *sun*, um eine Standby-Instanz zu ermöglichen.

```
strmqm -x QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. Erstellen Sie eine zweite Instanz von *QMGR* unter *earth*.
   a) Überprüfen Sie, ob die Werte der Präfix- und Installationsname-Parameter für *earth* korrekt sind.

   Führen Sie unter *sun*den Befehl **dspmqinf** aus:

   ```
   dspmqinf QMGR
   ```

   Systemantwort:

   ```
   QueueManager:
   Name=QMGR
   Directory=QMGR
   Prefix=C:\ProgramData\IBM\MQ
   DataPath=\\mars\wmq\data\QMGR
   InstallationName=Installation1
   ```

   b) Kopieren Sie die maschinenlesbare Form der Zeilengruppe **QueueManager** in die Zwischenablage.

   Führen Sie unter *sun* den Befehl **dspmqinf** erneut mit dem Parameter `-o command` aus.

   ```
   dspmqinf -o command QMGR
   ```

   Systemantwort:

   ```
   addmqinf -s QueueManager -v Name=QMGR
   -v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
   -v DataPath=\\mars\wmq\data\QMGR
   ```

c) Führen Sie unter *earth* den Befehl **addmqinf** aus der Zwischenablage heraus aus, um eine Instanz des Warteschlangenmanagers unter *earth*zu erstellen.

Passen Sie den Befehl bei Bedarf an, um die Unterschiede in den Parametern `Präfix` oder `InstallationName` zu berücksichtigen.

```
addmqinf -s QueueManager -v Name= QMGR
-v Directory= QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

```
IBM MQ configuration information added.
```

6. Starten Sie die Standby-Instanz des Warteschlangenmanagers unter *earth*.

```
strmqm -x QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

## Ergebnisse

Stellen Sie sicher, dass der Warteschlangenmanager von *sun* auf *earth* umschaltet:

1. Führen Sie unter *sun* den folgenden Befehl aus:

```
endmqm -i -r -s QMGR
```

Die Systemantwort auf *sun*:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

2. Geben Sie in *earth* den folgenden Befehl wiederholt ein:

```
dspmq
```

Die Systemantworten:

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

## Nächste Schritte

Ein Mehrinstanz-WS-Manager kann auch mit Musterprogrammen überprüft werden. Informationen hierzu finden Sie im Abschnitt „Multi-Instanz-Warteschlangenmanager unter Windows überprüfen” auf Seite 576.

**Zugehörige Tasks**

„Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne” auf Seite 572

„IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren” auf Seite 574

**Zugehörige Informationen**

Clusterknoten vonWindows 2000, Windows Server 2003 und Windows Server 2008 als Domänencontroller

Windows *Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne*

Fügen Sie der *wmq.example.com*-Domäne einen zweiten Domänencontroller hinzu, um eine Windows-Domäne zu erstellen, in der Multi-Instanz-Warteschlangenmanager auf Domänencontrollern und Dateiservern ausgeführt werden.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
: Ein Server von Windows Server 2008, der als erster Domänencontroller verwendet wird. Sie definiert die *wmq.example.com*-Domäne, die *sun*, *earth* und *mars*enthält. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***earth***
: Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***mars***
: Ein Windows Server 2008, der als Dateiserver verwendet wird.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Führen Sie die Schritte in „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 aus, um einen Domänencontroller, *sun*, für die Domäne *wmq.example.com* zu erstellen. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.
2. Installieren Sie Windows Server 2008 auf einem Server in der Standardarbeitsgruppe WORKGROUP. Für das Beispiel wird der Server *earth* genannt.

## Informationen zu diesem Vorgang

In dieser Task konfigurieren Sie einen Windows Server 2008 mit dem Namen *earth* als zweiten Domänencontroller in der *wmq.example.com*-Domäne.

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550.

## Vorgehensweise

1. Fügen Sie den Domänencontroller *sun.wmq.example.com* zu *earth* als DNS-Server hinzu.
   a) Melden Sie sich unter *earth* als *earth*\Administrator an und klicken Sie auf **Start**.
   b) Klicken Sie mit der rechten Maustaste auf **Network** > **Properties** > **Manage network connections** (Netz > Eigenschaften > Netzverbindungen verwalten).
   c) Klicken Sie auf den Netzadapter, und klicken Sie auf **Eigenschaften** .

      Das System antwortet mit dem Fenster "Merkmale des lokalen Bereichs", in dem die von der Verbindung verwendeten Elemente aufgelistet werden.
   d) Wählen Sie **Internet Protocol Version 4** oder **Internet Protocol IBM WebSphere MQ 6** aus der Liste der Elemente im Fenster "Local Area Connection Properties" (Eigenschaften der Verbindung

im lokalen Netz) aus. Klicken Sie auf **Eigenschaften** > **Erweitert ...** und klicken Sie auf die Registerkarte **DNS** .

e) Klicken Sie unter den DNS-Serveradressen auf **Hinzufügen ...** .

f) Geben Sie die IP-Adresse des Domänencontrollers ein, bei dem es sich auch um den DNS-Server handelt, und klicken Sie auf **Hinzufügen** .

g) Klicken Sie auf **Diese DNS-Suffixe anhängen** > **Hinzufügen ...**.

h) Geben Sie *wmq.example.com* ein und klicken Sie auf **Hinzufügen**

i) Geben Sie *wmq.example.com* in das Feld **DNS-Suffix für diese Verbindung** ein.

j) Wählen Sie die Option **Diese Verbindungsadresse in DNS registrieren** und **Das Suffix dieser Verbindung in der DNS-Registrierung verwenden** aus. Klicken Sie auf **OK** > **OK** > **Schließen**.

k) Öffnen Sie ein Befehlsfenster und geben Sie den Befehl **ipconfig /all** ein, um die TCP/IP-Einstellungen zu überprüfen.

2. Melden Sie sich als lokaler Administrator oder `Workgroup`-Administrator am Domänencontroller ( *sun*) an.

   Wenn der Server bereits als Domänencontroller konfiguriert ist, müssen Sie sich als Domänenadministrator anmelden.

3. Führen Sie den Assistenten für die Active Directory-Domänenservices aus.

   a) Klicken Sie **Start** > **Ausführen ...** Geben Sie `dcpromo` ein und klicken Sie auf **OK**.

   Wenn die Binärdateien von Active Directory noch nicht installiert sind, installiert Windows die Dateien automatisch.

4. Konfigurieren Sie *earth* als zweiten Domänencontroller in der *wmq.example.com*-Domäne.

   a) Lassen Sie im ersten Fenster des Assistenten das Kontrollkästchen **Erweiterte Modusinstallation verwenden** leer. Klicken Sie auf **Next** > **Next** und klicken Sie auf **Create Add a domain controller to an existing domain** > **Next**.

   b) Geben Sie *wmq* in das Feld **Geben Sie den Namen einer beliebigen Domäne in dieser Gesamtstruktur ein ...** ein Feld angegeben haben. Klicken Sie auf das Optionsfeld **Alternative Berechtigungsnachweise** , und klicken Sie auf **Festlegen ...** . Geben Sie den Namen und das Kennwort des Domänenadministrators ein und klicken Sie auf **OK** > **Weiter** > **Weiter** > **Weiter**.

   c) Akzeptieren Sie im Fenster "Zusätzliche Domänencontrolleroptionen" die Optionen **DNS-Server** und **Globaler Katalog**, die ausgewählt sind. Klicken Sie dann auf **Weiter** > **Weiter**.

   d) Geben Sie in "Administratorkennwort für den Verzeichnisdienst-Restore-Modus" ein **Kennwort** und ein **Bestätigungskennwort** ein und klicken Sie auf **Weiter** > **Weiter**.

   e) Geben Sie das Kennwort des Domänenadministrators ein, wenn Sie dazu aufgefordert werden, **Network Credentials** einzugeben. Wählen Sie **Reboot on completion** im letzten Assistentenfenster aus.

   f) Nach einer Weile wird möglicherweise ein Fenster mit einem **DCPromo** -Fehler bezüglich der DNS-Delegierung geöffnet. Klicken Sie auf **OK**. Der Server wird neu gestartet.

## Ergebnisse

Wenn *earth* erneut gestartet wurde, melden Sie sich als Domänenadministrator an. Überprüfen Sie, ob die `wmq.example.com`-Domäne in *earth* repliziert wurde.

## Nächste Schritte

Fahren Sie mit der Installation von IBM MQ fort; siehe „IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren” auf Seite 574.

**Zugehörige Tasks**

Windows IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren

„Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555

Windows *IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren*

Installieren und konfigurieren Sie Installationen von IBM MQ auf beiden Domänencontrollern in der *wmq.example.com*-Domäne.

Die Beispielkonfiguration besteht aus drei Servern:

***sun***
Ein Server von Windows Server 2008, der als erster Domänencontroller verwendet wird. Sie definiert die *wmq.example.com*-Domäne, die *sun*, *earth* und *mars*enthält. Sie enthält eine Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***earth***
Ein Windows Server 2008, der als der zweite IBM MQ-Server verwendet wird. Sie enthält die zweite Instanz des Multi-Instanz-Warteschlangenmanagers mit dem Namen *QMGR*.

***mars***
Ein Windows Server 2008, der als Dateiserver verwendet wird.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Führen Sie die Schritte in „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 aus, um einen Domänencontroller, *sun*, für die Domäne *wmq.example.com* zu erstellen. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.
2. Führen Sie die Schritte in „Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne” auf Seite 572 aus, um einen zweiten Domänencontroller ( *earth*) für die Domäne *wmq.example.com* zu erstellen. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.
3. Im Abschnitt Hardware- und Softwarevoraussetzungen auf Windows-Systemen finden Sie Informationen zu weiteren Windows-Versionen, auf denen IBM MQ ausgeführt werden kann.

## Informationen zu diesem Vorgang

Installieren und konfigurieren Sie Installationen von IBM MQ auf beiden Domänencontrollern in der *wmq.example.com*-Domäne.

## Vorgehensweise

1. Installieren Sie IBM MQ auf *sun* und *earth*.

   Weitere Informationen finden Sie unter IBM MQ-Server unter Windows installieren.

   a) Melden Sie sich sowohl auf *sun* als auch auf *earth* als Domänenadministrator an, *wmq*\Administrator.

   b) Führen Sie den Befehl **Setup** von den IBM MQ for Windows-Installationsmedien aus.

   Die Launchpadanwendung von IBM MQ wird gestartet.

   c) Klicken Sie auf **Softwarevoraussetzungen** , um zu überprüfen, ob die vorausgesetzte Software installiert ist.

   d) Klicken Sie auf **Netzwerkkonfiguration** > **Nein**.

   Sie können entweder eine Domänenbenutzer-ID oder nicht für diese Installation konfigurieren. Die Benutzer-ID, die erstellt wird, ist eine lokale Benutzer-ID.

   e) Klicken Sie auf **IBM MQ-Installation**, wählen Sie eine Installationssprache aus und klicken Sie auf "Launch IBM MQ Installer" (IBM MQ-Installationsprogramm starten).

   f) Bestätigen Sie die Lizenzvereinbarung und klicken Sie auf **Weiter** > **Weiter** > **Installieren**, um die Standardkonfiguration zu akzeptieren. Warten Sie, bis die Installation abgeschlossen ist, und klicken Sie auf **Fertig stellen** .

Wenn Sie den Namen der Installation ändern möchten, verschiedene Komponenten installieren, ein anderes Verzeichnis für WS-Manager-Daten und -Protokolle konfigurieren oder in einem anderen Verzeichnis installieren möchten, klicken Sie auf **Angepasst** und nicht auf **Typisch** .

IBM MQ wird installiert und der Prepare IBM MQ Wizard wird vom Installationsprogramm gestartet.

Die Installation von IBM MQ for Windows konfiguriert eine lokale Domänengruppe `mqm` und eine Domänengruppe `Domain mqm`. Sie macht `Domain mqm` zu einem Mitglied von `mqm`. Nachfolgende Domänencontroller in derselben Domäne verwenden die Gruppen `mqm` und `Domain mqm`.

2. Führen Sie in *earth* und *sun* den Prepare IBM MQ Wizard aus.

   Weitere Informationen finden Sie im Abschnitt IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren.

   a) Das IBM MQ-Installationsprogramm führt den Prepare IBM MQ Wizard automatisch aus.

   Für den manuellen Start des Assistenten suchen Sie den Direktaufruf für den Prepare IBM MQ Wizard im Ordner **Start** > **Alle Programme** > **IBM MQ**. Wählen Sie in einer Konfiguration mit mehreren Installationen den entsprechenden Direktaufruf für die IBM MQ-Installation aus.

   b) Klicken Sie auf **Weiter** und lassen Sie **Nein** als Antwort auf die Frage "Geben Sie an, ob ein Domänencontroller von Windows 2000 oder höher im Netz vorhanden ist" stehen.[1].

   c) Wählen Sie auf der letzten Seite des Assistenten die Markierungsfelder aus, die Sie benötigen, und klicken Sie auf **Fertig stellen** .

   Der Prepare IBM MQ Wizard erstellt einen lokalen Domänenbenutzer `MUSR_MQADMIN` auf dem ersten Domänencontroller und einen anderen lokalen Domänenbenutzer `MUSR_MQADMIN1` auf dem zweiten Domänencontroller. Der Assistent erstellt den IBM MQ-Service auf jedem Controller, wobei die Anmeldung des Service als Benutzer `MUSR_MQADMIN` oder `MUSR_MQADMIN1` erfolgt.

3. Definieren Sie einen Benutzer, der über die Berechtigung zum Erstellen eines Warteschlangenmanagers verfügt.

   Der Benutzer muss das Recht haben, sich lokal anzumelden und ein Mitglied der lokalen `mqm`-Gruppe zu sein. Domänenbenutzer haben auf Domänencontrollern nicht das Recht, sich lokal anzumelden, aber Administratoren tun dies. Standardmäßig verfügt kein Benutzer über beide Attribute. Fügen Sie in dieser Task Domänenadministratoren zur lokalen `mqm`-Gruppe hinzu.

   a) Öffnen Sie **Server Manager** > **Roles** > **Active Directory Domain Services** > ***wmq.example.com*** > **Users** (Servermanager -> Rollen -> Active Directory Domain Services -> wmq.example.com -> Benutzer).

   b) Klicken Sie mit der rechten Maustaste auf **Domänenadministratoren** > **Zu Gruppe hinzufügen ...** und geben Sie `mqm` ein; Klicken Sie auf **Namen prüfen (Check names)** > **OK** > **OK**

## Ergebnisse

1. Stellen Sie sicher, dass der Prepare IBM MQ Wizard den Domänenbenutzer `MUSR_MQADMIN` erstellt hat:

   a. Öffnen Sie **Server Manager** > **Roles** > **Active Directory Domain Services** > ***wmq.example.com*** > **Users** (Servermanager -> Rollen -> Active Directory Domain Services -> wmq.example.com -> Benutzer).

   b. Klicken Sie mit der rechten Maustaste auf **MUSR_MQADMIN** > **Eigenschaften ...** > **Mitglied von**und sehen, dass es ein Mitglied von `Domain users` und `mqm`ist

2. Stellen Sie sicher, dass `MUSR_MQADMIN` das Recht hat, als Dienst ausgeführt zu werden:

   a. Klicken Sie auf **Starten** > **Ausführen**. Geben Sie den Befehl **`secpol.msc`** ein und klicken Sie auf **OK**. anklicken

[1] Sie können die Installation für die Domäne konfigurieren. Da alle Benutzer und Gruppen auf einem Domänencontroller einen Domänenbereich haben, macht dies keinen Unterschied. Es ist einfacher, IBM MQ so zu installieren, als ob es nicht in der Domäne vorhanden ist.

b. Öffnen Sie **Sicherheitseinstellungen** > **Lokale Richtlinien** > **Zuweisen von Benutzerrechten**. Klicken Sie in der Liste der Richtlinien mit der rechten Maustaste auf **Log on as a service** > **Properties** (Als Dienst anmelden > Eigenschaften) und stellen Sie fest, ob MUSR_MQADMIN mit dem Recht, sich als Service anzumelden, aufgeführt ist. Klicken Sie auf **OK**.

## Nächste Schritte

1. Führen Sie die Task „Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind" auf Seite 584 aus, um zu prüfen, ob die Installation und Konfiguration ordnungsgemäß funktionieren.
2. Kehren Sie zur Task „WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen" auf Seite 568 zurück, um die Task zur Konfiguration eines Multi-Instanz-Warteschlangenmanagers auf Domänencontrollern auszuführen.

**Zugehörige Tasks**

Windows Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne

**Zugehörige Verweise**

Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service

Windows *Multi-Instanz-Warteschlangenmanager unter Windows überprüfen*

Verwenden Sie die Beispielprogramme **amqsghac**, **amqsphac** und **amqsmhac** , um die Konfiguration eines Multi-Instanz-Warteschlangenmanagers zu überprüfen. Dieser Abschnitt enthält eine Beispielkonfiguration für die Überprüfung der Konfiguration eines Multi-Instanz-Warteschlangenmanagers unter Windows Server 2003.

Die Beispielprogramme für hohe Verfügbarkeit verwenden die automatische Clientwiederverbindung. Wenn der verbundene Warteschlangenmanager ausfällt, versucht der Client, die Verbindung zu einem WS-Manager in derselben Warteschlangenmanager-Gruppe wieder herzustellen. Die Beschreibung der Beispiele Hochverfügbarkeits-Musterprogramme veranschaulicht die Clientwiederverbindung unter Verwendung eines einzigen Instanz-WS-Managers für die Einfachheit. Sie können dieselben Muster mit Warteschlangenmanagern mit mehreren Instanzen verwenden, um eine Konfiguration mit mehreren Instanzen des Warteschlangenmanagers zu überprüfen.

In diesem Beispiel wird die Konfiguration mit mehreren Instanzen verwendet, die in „WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen" auf Seite 568 beschrieben wird. Überprüfen Sie mit der Konfiguration, ob der Warteschlangenmanager mit mehreren Instanzen in die Standby-Instanz umschaltet. Stoppen Sie den Warteschlangenmanager mit dem Befehl **endmqm** und verwenden Sie die Option -s(Switchover). Die Clientprogramme stellen die Verbindung zur neuen Warteschlangenmanagerinstanz wieder her und arbeiten nach einer geringfügigen Verzögerung weiterhin mit der neuen Instanz.

Der Client wird in einem VMware-Image von 400 MB installiert, auf dem Windows 7 Service Pack 1 ausgeführt wird. Aus Sicherheitsgründen ist sie auf demselben nur-Host-Netz von VMWare wie die Domänenserver, auf denen der Multi-Instanz-Warteschlangenmanager ausgeführt wird, verbunden. Sie teilt den Ordner /MQHA, der die Clientverbindungstabelle enthält, um die Konfiguration zu vereinfachen.

### Funktionsübernahme mit IBM MQ Explorer überprüfen

Bevor Sie die Beispielanwendungen verwenden, um die Funktionsübernahme zu überprüfen, führen Sie den IBM MQ Explorer auf jedem Server aus. Fügen Sie jedem Explorer die beiden WS-Manager-Instanzen hinzu, indem Sie den Assistenten **Remote Queue Manager hinzufügen > Direkt mit einem Multi-Instanz-Warteschlangenmanager verbinden** verwenden. Stellen Sie sicher, dass beide Instanzen aktiv sind und den Standby-Modus zulassen. Schließen Sie das Fenster, in dem das VMware-Image mit der aktiven Instanz ausgeführt wird, schließen Sie den Server virtuell ab, oder stoppen Sie die aktive Instanz, indem Sie die Umschaltung auf die Standby-Instanz zulassen und Clients erneut verbinden können, um die Verbindung herzustellen.

**Achtung:** Wenn Sie den Server ausschalten, stellen Sie sicher, dass es sich nicht um den Server handelt, der den Ordner MQHA hostet!

**Anmerkung:** Die Option **Umschalten auf eine Standby-Instanz zulassen** ist unter Umständen nicht im Dialog **Warteschlangenmanager stoppen** verfügbar. Die Option fehlt, weil der WS-Manager als einzelner Instanz-Warteschlangenmanager ausgeführt wird. Sie müssen diese Option ohne die Option **Standby-Instanz zulassen** gestartet haben. Wenn Ihre Anforderung zum Stoppen des Warteschlangenmanagers zurückgewiesen wird, sehen Sie sich das Fenster **Details** an. Möglicherweise ist keine Standby-Instanz aktiv.

### Funktionsübernahme mit den Musterprogrammen überprüfen

**Wählen Sie einen Server zum Ausführen der aktiven Instanz aus.**

Möglicherweise haben Sie einen der Server ausgewählt, um das MQHA-Verzeichnis oder das Dateisystem zu hosten. Wenn Sie die Funktionsübernahme testen möchten, indem Sie das VMware-Fenster, auf dem der aktive Server ausgeführt wird, schließen, stellen Sie sicher, dass es sich nicht um den handelt, der MQHA hostet!

**Auf dem Server, auf dem die aktive WS-Manager-Instanz ausgeführt**

1. Ändern Sie *ipaddr1* und *ipaddr2* und speichern Sie die folgenden Befehle in `N:\hasample.tst`.

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN)  TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2 (1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

   **Anmerkung:** Wenn Sie den Parameter **MCAUSER** leer lassen, wird die Clientbenutzer-ID an den Server gesendet. Die Clientbenutzer-ID muss über die korrekten Berechtigungen für die Server verfügen. Alternativ können Sie den Parameter **MCAUSER** im Kanal SVRCONN auf die Benutzer-ID setzen, die Sie auf dem Server konfiguriert haben.

2. Öffnen Sie eine Eingabeaufforderung mit dem Pfad `N:\`, und führen Sie den folgenden Befehl aus:

```
runmqsc -m QM1 < hasample.tst
```

3. Überprüfen Sie, ob der Listener aktiv ist und die Steuerung des Warteschlangenmanagers hat, indem Sie die Ausgabe des Befehls **runmqsc** überprüfen.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

   Oder verwenden Sie den IBM MQ Explorer, der vom TCPIP-Listener ausgeführt wird, und auf dem `Control = Queue Manager` eingestellt ist.

**Auf dem Client**

1. Ordnen Sie das gemeinsam genutzte Verzeichnis `C:\MQHA` auf dem Server `N:\` auf dem Client zu.
2. Öffnen Sie eine Eingabeaufforderung mit dem Pfad `N:\`. Setzen Sie die Umgebungsvariable `MQCHLLIB` so, dass sie auf die Clientkanaldefinitionstabelle (CCDT) auf dem Server verweist:

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. Geben Sie an der Eingabeaufforderung die folgenden Befehle ein:

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

**Anmerkung:** Wenn Sie Probleme haben, starten Sie die Anwendungen an einer Eingabeaufforderung, sodass der Ursachencode auf der Konsole ausgegeben wird, oder sehen Sie sich die AMQERR01.LOG-Datei im Ordner `N:\data\QM1\errors` an.

**Auf dem Server, auf dem die aktive WS-Manager-Instanz ausgeführt**

1. Entweder:
   - Schließen Sie das Fenster, in dem das VMware-Image mit der aktiven Serverinstanz ausgeführt wird.
   - Stoppen Sie mit dem IBM MQ Explorer die aktive WS-Manager-Instanz und lassen Sie dabei das Umschalten auf die Standby-Instanz zu und weisen Sie erneut verbindbare Clients an, die Verbindung erneut herzustellen.
2. Die drei Clients erkennen schließlich, dass die Verbindung unterbrochen ist, und stellen Sie dann die Verbindung wieder her. Wenn Sie in dieser Konfiguration das Serverfenster schließen, wird für alle drei Verbindungen, die neu aufgebaut werden sollen, ungefähr sieben Minuten Zeit. Einige Verbindungen werden vor anderen wieder hergestellt.

**Ergebnisse**

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Windows *Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern*

In diesem Thema wird beschrieben, wie Sie eine gemeinsam genutzte Position für WS-Manager-Daten und -Protokolldateien mit einer globalen alternativen Sicherheitsgruppe sichern können. Sie können die Position zwischen verschiedenen Instanzen eines Warteschlangenmanagers, der auf verschiedenen Servern ausgeführt wird, gemeinsam nutzen.

In der Regel legen Sie keine gemeinsam genutzte Position für WS-Manager-Daten und Protokolldateien fest. Wenn Sie IBM MQ for Windows installieren, erstellt das Installationsprogramm ein Ausgangsverzeichnis Ihrer Wahl für alle WS-Manager, die auf diesem Server erstellt werden. Er sichert die Verzeichnisse mit der lokalen `mqm`-Gruppe und konfiguriert eine Benutzer-ID für den IBM MQ-Service, um auf die Verzeichnisse zuzugreifen.

Wenn Sie einen gemeinsam genutzten Ordner mit einer Sicherheitsgruppe sichern, muss ein Benutzer, der für den Zugriff auf den Ordner berechtigt ist, über die Berechtigungsnachweise der Gruppe verfügen. Angenommen, ein Ordner auf einem fernen Dateiserver ist mit der lokalen `mqm` -Gruppe auf einem Server mit dem Namen *`mars`* geschützt. Stellen Sie sicher, dass der Benutzer, der Warteschlangenmanager ausführt, ein Mitglied der lokalen `mqm` -Gruppe in *`mars`* verarbeitet. Der Benutzer verfügt über die Berechtigungsnachweise, die mit den Berechtigungsnachweisen des Ordners auf dem fernen Dateiserver übereinstimmen. Mit diesen Berechtigungsnachweisen ist der Warteschlangenmanager in der Lage, auf seine Daten zuzugreifen und die Dateien im Ordner zu Protokolldateien zu verwenden. Der Benutzer, der WS-Manager-Prozesse auf einem anderen Server ausführt, ist Mitglied einer anderen lokalen `mqm` -Gruppe, die keine übereinstimmenden Berechtigungsnachweise hat. Wenn der Warteschlangenmanager auf einem anderen Server als *`mars`* ausgeführt wird, kann er nicht auf die Daten und Protokolldateien zugreifen, die er erstellt hat, als er unter *`mars`* ausgeführt wurde. Selbst wenn Sie den Benutzer zu einem Domänenbenutzer machen, verfügt er über unterschiedliche Berechtigungsnachweise, da er die Berechtigungsnachweise aus der lokalen `mqm` -Gruppe auf *`mars`* erwerben muss und dies nicht von einem anderen Server aus ausgeführt werden kann.

Die Bereitstellung des Warteschlangenmanagers mit einer globalen alternativen Sicherheitsgruppe löst das Problem (siehe Abbildung 73 auf Seite 580). Sichern Sie einen fernen Ordner mit einer globalen Gruppe. Übergeben Sie den Namen der globalen Gruppe an den Warteschlangenmanager, wenn Sie ihn unter *`mars`* erstellen. Übergeben Sie den globalen Gruppennamen als alternative Sicherheitsgruppe mit dem Parameter `-a [r]` im Befehl **`crtmqm`** . Wenn Sie den Warteschlangenmanager zur Ausführung auf einem anderen Server übertragen, wird der Name der Sicherheitsgruppe mit dem Namen der Sicherheitsgruppe übertragen. Der Name wird in der Zeilengruppe **AccessMode** in der Datei `qm.ini` als `SecurityGroup` übertragen, z. B.:

```
AccessMode:
SecurityGroup=wmq\wmq
```

Die Zeilengruppe **AccessMode** in der `qm.ini` enthält auch die Datei `RemoveMQMAccess` , z. B.:

```
AccessMode:
RemoveMQMAccess=true|false
```

Wenn dieses Attribut mit dem Wert `true` angegeben wird und auch eine Zugriffsgruppe angegeben wurde, erhält die lokale Gruppe 'mqm' keinen Zugriff auf die Datendateien des Warteschlangenmanagers.

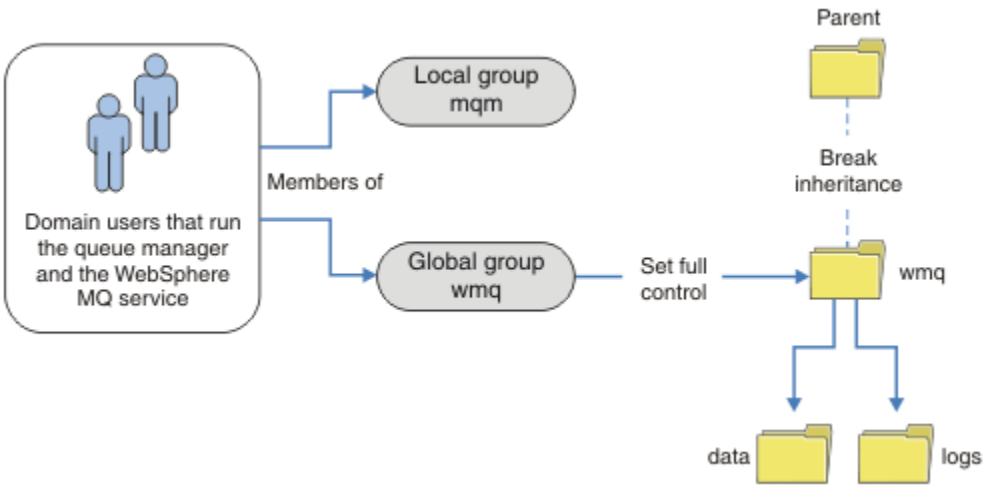



*Abbildung 73. Sichern von Warteschlangenmanagerdaten und -protokollen mithilfe einer alternativen globalen Sicherheitsgruppe (1)*

Für die Benutzer-ID, mit der WS-Manager-Prozesse ausgeführt werden sollen, um die übereinstimmenden Berechtigungsnachweise der globalen Sicherheitsgruppe zu verwenden, muss die Benutzer-ID auch über einen globalen Geltungsbereich verfügen. Sie können keine lokale Gruppe oder Principal als Mitglied einer globalen Gruppe erstellen. In Abbildung 73 auf Seite 580 werden die Benutzer, die die WS-Manager-Prozesse ausführen, als Domänenbenutzer angezeigt.

Wenn Sie viele IBM MQ-Server implementieren, ist die Gruppierung von Benutzern in Abbildung 73 auf Seite 580 nicht praktisch. Sie müssen den Prozess zum Hinzufügen von Benutzern zu lokalen Gruppen für jeden IBM MQ-Server wiederholen. Erstellen Sie stattdessen eine globale `Domain mqm`-Gruppe auf dem Domänencontroller und machen Sie die Benutzer, die IBM MQ ausführen zu Mitgliedern der `Domain mqm`-Gruppe. Informationen hierzu finden Sie im Artikel Abbildung 74 auf Seite 581. Wenn Sie IBM MQ als Domäneninstallation installieren, macht Prepare IBM MQ Wizard die `Domain mqm`-Gruppe automatisch zu einem Mitglied der lokalen `mqm`-Gruppe. Dieselben Benutzer sind sowohl in den globalen Gruppen `Domain mqm` als auch in `wmq`enthalten.

**Tipp:** Dieselben Benutzer können IBM MQ auf verschiedenen Servern ausführen,aber auf einem einzelnen Server müssen Sie über unterschiedliche Benutzer verfügen, um IBM MQ als Service und interaktiv auszuführen. Sie müssen auch für jede Installation auf einem Server unterschiedliche Benutzer haben. In der Regel enthält `Domain mqm` daher eine Reihe von Benutzern.

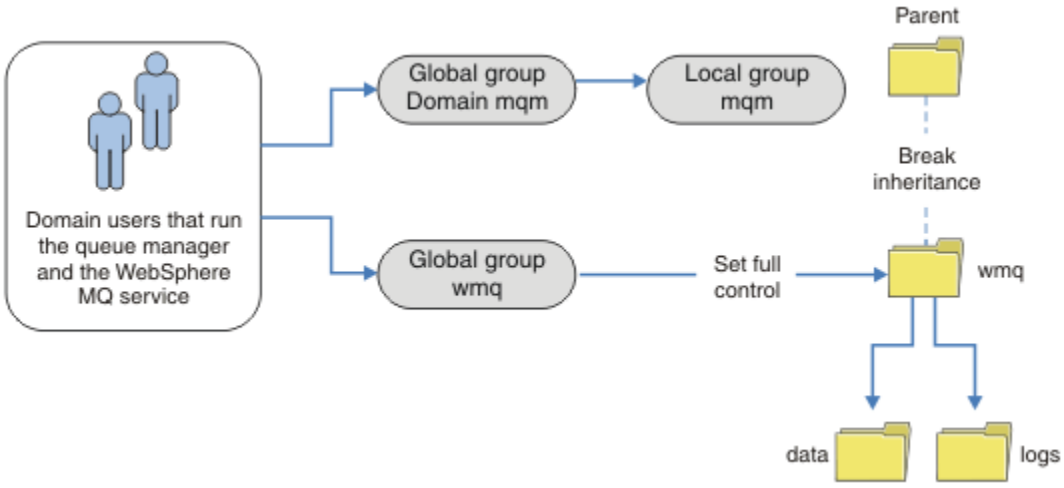



*Abbildung 74. Sichern von Warteschlangenmanagerdaten und -protokollen mithilfe einer alternativen globalen Sicherheitsgruppe (2)*

Die Organisation in Abbildung 74 auf Seite 581 ist nach dem Stand der Dinge unnötig kompliziert. Die Anordnung hat zwei globale Gruppen mit identischen Membern. Sie können die Organisation vereinfachen und nur eine globale Gruppe definieren (siehe Abbildung 75 auf Seite 581).

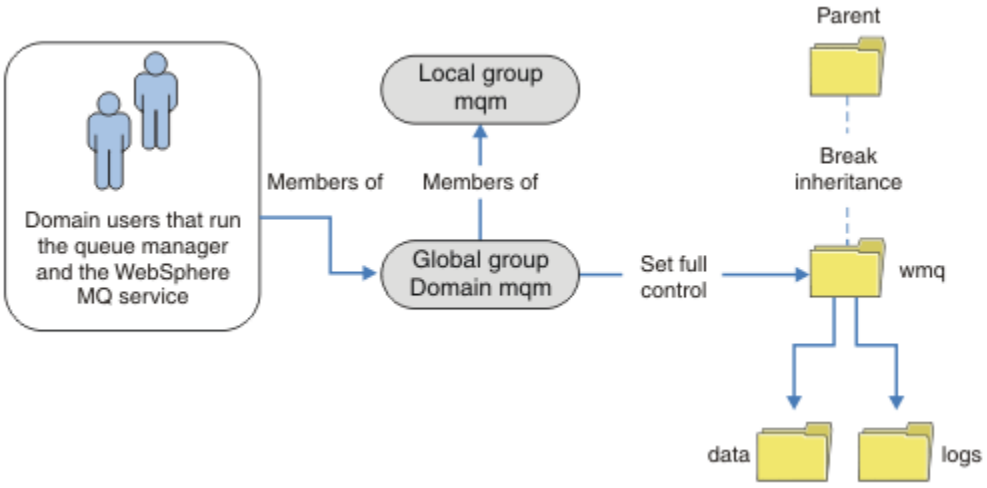



*Abbildung 75. Sichern von WS-Manager-Daten und -Protokollen mithilfe einer alternativen globalen Sicherheitsgruppe (3)*

Alternativ hierzu benötigen Sie möglicherweise eine feinere Zugriffssteuerung, wobei die verschiedenen Warteschlangenmanager auf verschiedene Ordner zugreifen können. Weitere Informationen finden Sie im Abschnitt Abbildung 76 auf Seite 582. In Abbildung 76 auf Seite 582 sind zwei Gruppen von Domänenbenutzern definiert, in separaten globalen Gruppen, um verschiedene WS-Manager-Protokoll-und Datendateien zu sichern. Es werden zwei verschiedene lokale `mqm`-Gruppen angezeigt, die sich auf verschiedenen IBM MQ-Servern befinden müssen. In diesem Beispiel werden die Warteschlangenmanager in zwei Gruppen unterteilt, wobei verschiedene Benutzer den beiden Gruppen zugeordnet sind. Die beiden Gruppen können Test-und Produktionswarteschlangenmanager sein. Die alternativen Sicherheitsgruppen werden als `wmq1` und `wmq2` bezeichnet. Sie müssen die globalen Gruppen `wmq1` und `wmq2` manuell zu den richtigen Warteschlangenmanagern hinzufügen, je nach dem, ob sie sich in der Test-oder Produktionsabteilung befinden. Die Konfiguration kann nicht nutzen, dass die Installation von IBM MQ `Domain mqm` an

die lokale mqm -Gruppe wie in Abbildung 75 auf Seite 581weitergegeben wird, da es zwei Benutzergruppen gibt.

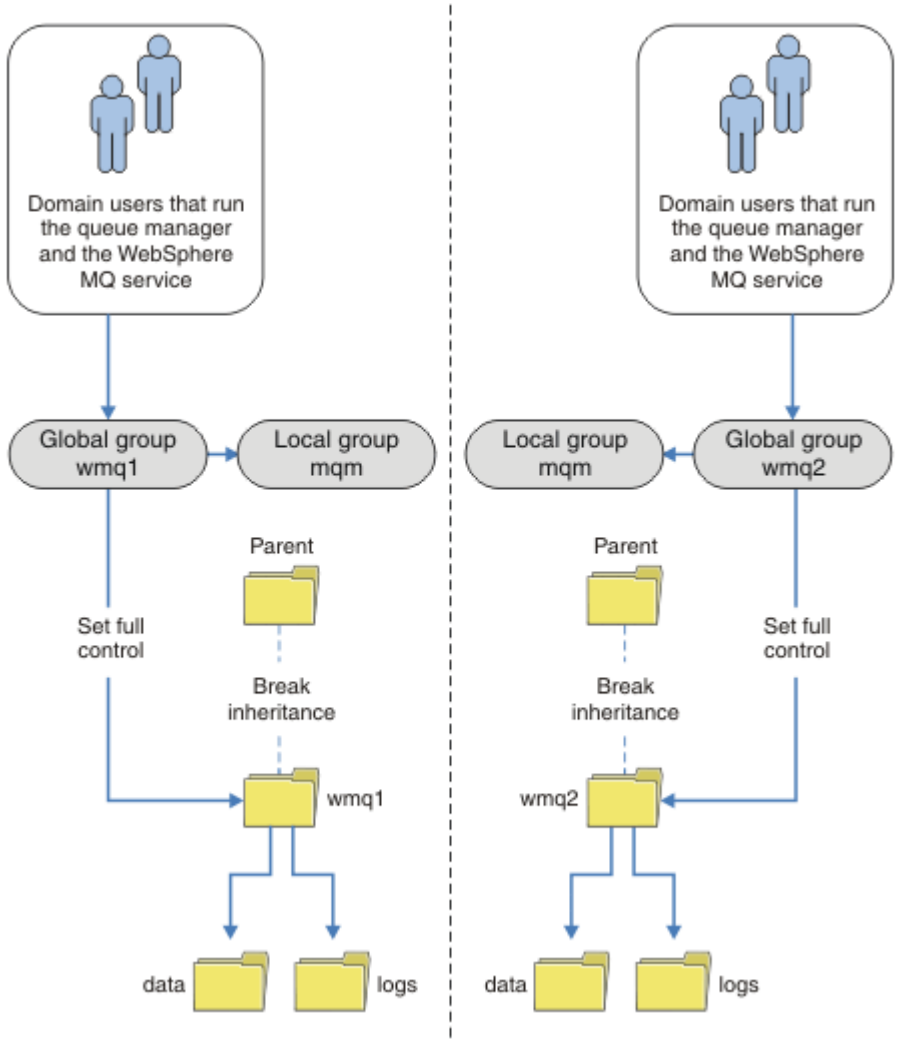



*Abbildung 76. Sichern von Warteschlangenmanagerdaten und -protokollen mit einem alternativen globalen Sicherheitsprovider (4)*

Eine alternative Möglichkeit, zwei Abteilungen zu partitionieren, besteht darin, sie in zwei Windows-Domänen zu platzieren. In diesem Fall können Sie zur Verwendung des einfacheren Modells zurückkehren, das in Abbildung 75 auf Seite 581 angezeigt wird.

Windows *Nicht gemeinsam genutzte WS-Manager-Daten und -Protokollverzeichnisse und -Dateien unter Windows schützen*

In diesem Thema wird beschrieben, wie Sie eine alternative Position für WS-Manager-Daten und -Protokolldateien sichern können, indem Sie die lokale mqm -Gruppe und eine alternative Sicherheitsgruppe verwenden.

In der Regel wird keine alternative Position für WS-Manager-Daten und -Protokolldateien eingerichtet. Wenn Sie IBM MQ for Windows installieren, erstellt das Installationsprogramm ein Ausgangsverzeichnis Ihrer Wahl für alle Warteschlangenmanager, die erstellt werden. Er sichert die Verzeichnisse mit der lokalen mqm-Gruppe und konfiguriert eine Benutzer-ID für den IBM MQ-Service, um auf die Verzeichnisse zuzugreifen.

In zwei Beispielen wird gezeigt, wie die Zugriffssteuerung für IBM MQ konfiguriert wird. In den Beispielen wird gezeigt, wie ein Warteschlangenmanager mit seinen Daten und Protokollen in Verzeichnissen erstellt

wird, die sich nicht auf den von der Installation erstellten Daten und Protokollpfaden befinden. Im ersten Beispiel ermöglichen Sie „Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind" auf Seite 584 den Zugriff auf die Warteschlangen- und Protokollverzeichnisse, indem Sie die Berechtigung durch die lokale mqm-Gruppe zulassen. Das zweite Beispiel, „Lesen und Schreiben von Daten-und Protokolldateien, die von einer alternativen lokalen Sicherheitsgruppe autorisiert sind" auf Seite 588, unterscheidet sich dadurch, dass der Zugriff auf die Verzeichnisse durch eine alternative Sicherheitsgruppe autorisiert wird. Wenn auf die Verzeichnisse von einem Warteschlangenmanager zugegriffen wird, der auf einem einzigen Server ausgeführt wird, müssen Sie die Daten-und Protokolldateien mit der alternativen Sicherheitsgruppe sichern, um verschiedene Warteschlangenmanager mit unterschiedlichen lokalen Gruppen oder Principals zu sichern. Wenn auf die Verzeichnisse von einem Warteschlangenmanager zugegriffen wird, der auf verschiedenen Servern ausgeführt wird, z. B. mit einem Warteschlangenmanager mit mehreren Instanzen, ist die Sicherung der Daten und Protokolldateien mit der alternativen Sicherheitsgruppe die einzige Option (siehe „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern" auf Seite 579).

Die Konfiguration der Sicherheitsberechtigungen von WS-Manager-Daten und -Protokolldateien ist keine allgemeine Task unter Windows. Wenn Sie IBM MQ for Windows installieren, geben Sie entweder Verzeichnisse für WS-Manager-Daten und -Protokolle an, oder akzeptieren Sie die Standardverzeichnisse. Das Installationsprogramm sichert diese Verzeichnisse automatisch mit der lokalen `mqm` -Gruppe und gibt ihm die vollständige Steuerungsberechtigung. Der Installationsprozess stellt sicher, dass die Benutzer-ID, die Warteschlangenmanager ausführt, Mitglied der lokalen `mqm` -Gruppe ist. Sie können die anderen Zugriffsberechtigungen für die Verzeichnisse so ändern, dass sie Ihren Zugriffsvoraussetzungen entsprechen.

Wenn Sie das Verzeichnis für Daten-und Protokolldateien an neue Positionen verschieben, müssen Sie die Sicherheit der neuen Speicherpositionen konfigurieren. Sie können die Position der Verzeichnisse ändern, wenn Sie einen Warteschlangenmanager sichern und auf einem anderen Computer wiederherstellen, oder wenn Sie den Warteschlangenmanager als WS-Manager mit mehreren Instanzen ändern. Sie haben die Wahl zwischen zwei Möglichkeiten, die WS-Manager-Daten und die Protokollverzeichnisse an ihrer neuen Position zu sichern. Sie können die Verzeichnisse sichern, indem Sie den Zugriff auf die lokale `mqm` -Gruppe einschränken, oder Sie können den Zugriff auf eine beliebige Sicherheitsgruppe Ihrer Wahl einschränken.

Es wird mindestens die Anzahl der Schritte zur Sicherung der Verzeichnisse mit der lokalen Gruppe `mqm` benötigt. Legen Sie die Berechtigungen für die Daten-und Protokollverzeichnisse fest, um die vollständige Steuerung der lokalen `mqm` -Gruppe zu ermöglichen. Ein typischer Ansatz besteht darin, die vorhandene Gruppe von Berechtigungen zu kopieren und die Vererbung aus dem übergeordneten Element zu entfernen. Anschließend können Sie die Berechtigungen anderer Principals entfernen oder einschränken.

Wenn Sie den Warteschlangenmanager unter einer anderen Benutzer-ID als der vom IBM MQ-Vorbereitungsassistenten festgelegte Service ausführen, muss diese Benutzer-ID Mitglied der lokalen Gruppe `mqm` sein. Die Task „Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind" auf Seite 584 führt Sie durch die Schritte.

Sie können auch WS-Manager-Daten und -Protokolldateien mit einer alternativen Sicherheitsgruppe sichern. Der Prozess der Sicherung der WS-Manager-Daten und -Protokolldateien mit der alternativen Sicherheitsgruppe enthält eine Reihe von Schritten, die sich auf Abbildung 77 auf Seite 584 beziehen. Die lokale Gruppe, `wmq`, ist ein Beispiel für eine alternative Sicherheitsgruppe.

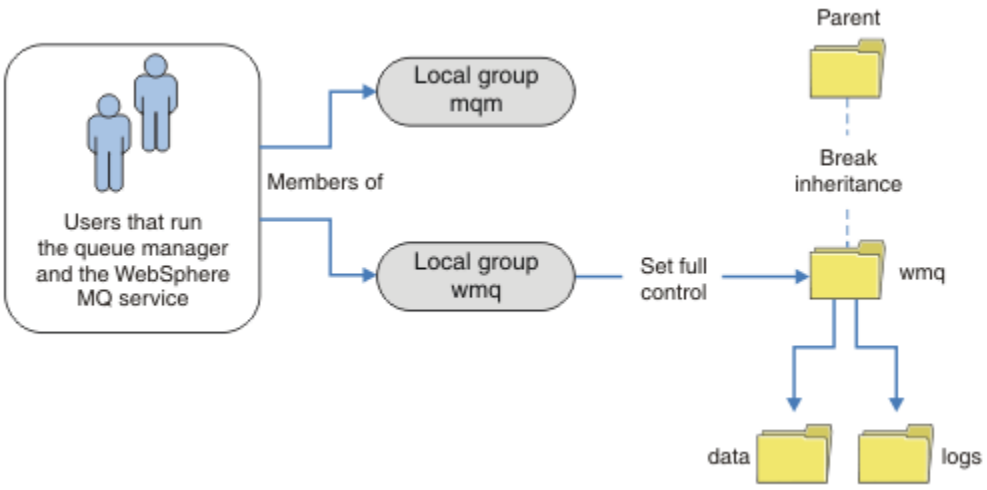



*Abbildung 77. Warteschlangenmanager-Daten und -Protokolle mithilfe einer alternativen lokalen Sicherheitsgruppe sichern, wmq*

1. Erstellen Sie entweder separate Verzeichnisse für die WS-Manager-Daten und -Protokolle, ein allgemeines Verzeichnis oder ein gemeinsames übergeordnetes Verzeichnis.
2. Kopieren Sie die vorhandene Gruppe von übernommenen Berechtigungen für die Verzeichnisse oder das übergeordnete Verzeichnis, und ändern Sie sie entsprechend Ihren Anforderungen.
3. Sichern Sie die Verzeichnisse, die den Warteschlangenmanager und die Protokolle enthalten sollen, indem Sie der alternativen Gruppe `wmq` die vollständige Steuerberechtigung für die Verzeichnisse angeben.
4. Geben Sie allen Benutzer-IDs, die Warteschlangenmanager ausführen, die Berechtigungsnachweise der alternativen Sicherheitsgruppe oder des Principals an:
   a. Wenn Sie einen Benutzer als alternativen Sicherheitprincipal definieren, muss der Benutzer derselbe Benutzer sein, unter dem der WS-Manager ausgeführt wird. Der Benutzer muss ein Mitglied der lokalen `mqm` -Gruppe sein.
   b. Wenn Sie eine lokale Gruppe als alternative Sicherheitsgruppe definieren, fügen Sie den Benutzer hinzu, den der WS-Manager unter der alternativen Gruppe ausführen wird. Der Benutzer muss auch ein Mitglied der lokalen `mqm` -Gruppe sein.
   c. Wenn Sie eine globale Gruppe als alternative Sicherheitsgruppe definieren, finden Sie weitere Informationen in „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern” auf Seite 579.
5. Erstellen Sie den WS-Manager unter Angabe der alternativen Sicherheitsgruppe oder des Principals im Befehl **`crtmqm`** mit dem Parameter `-a` .

Windows *Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind*

Die Task veranschaulicht, wie ein Warteschlangenmanager mit seinen Daten erstellt wird und die Dateien in einem beliebigen Verzeichnis Ihrer Wahl gespeichert werden. Der Zugriff auf die Dateien wird durch die lokale `mqm` -Gruppe gesichert. Das Verzeichnis wird nicht gemeinsam genutzt.

## Vorbereitende Schritte

1. IBM MQ for Windows als primäre Installation installieren.
2. Führen Sie die Prepare IBM MQ Wizard aus.

   Weitere Informationen finden Sie im Abschnitt IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren.

Konfigurieren Sie für diese Task die Installation entweder für die Ausführung mit einer lokalen Benutzer-ID oder für eine Domänenbenutzer-ID. Beachten Sie allerdings, dass die Installation später ohnehin für eine Domäne konfiguriert werden muss, um alle Tasks von „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550 ausführen zu können.

3. Melden Sie sich mit Administratorberechtigung an, um den ersten Teil der Task auszuführen.

## Informationen zu diesem Vorgang

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550.

Unter Windows können Sie als Standardpfad für Daten- und Protokolldateien für einen IBM MQ for Windows beliebige Verzeichnisse festlegen. Der Installations-und Konfigurationsassistent erteilt der lokalen `mqm` -Gruppe und der Benutzer-ID, die die Warteschlangenmanagerprozesse ausführt, automatisch Zugriff auf die Verzeichnisse. Wenn Sie einen Warteschlangenmanager erstellen, der unterschiedliche Verzeichnisse für WS-Manager-Daten und -Protokolldateien angibt, müssen Sie die vollständige Steuerberechtigung für die Verzeichnisse konfigurieren.

In diesem Beispiel geben Sie dem Warteschlangenmanager die vollständige Kontrolle über seine Daten- und Protokolldateien, indem Sie der lokalen `mqm`-Gruppe die Berechtigung für das Verzeichnis *c:\wmq* erteilen.

Der Befehl **crtmqm** erstellt einen Warteschlangenmanager, der automatisch beim Start der Workstation durch den IBM MQ-Service gestartet wird.

Die Task ist anschaulich. Sie verwendet bestimmte Werte, die Sie ändern können. Die Werte, die Sie ändern können, sind in Kursivschrift. Führen Sie am Ende der Task die Anweisungen aus, um alle Änderungen zu entfernen, die Sie vorgenommen haben.

## Vorgehensweise

1. Öffnen Sie eine Eingabeaufforderung.
2. Geben Sie den folgenden Befehl ein:

```
md c:\wmq\data, c:\wmq\logs
```

3. Legen Sie die Berechtigungen für die Verzeichnisse fest, um der lokalen `mqm`-Gruppe Lese- und Schreibzugriff zu erteilen.

```
cacls c:\wmq/T /E /G mqm:F
```

   Systemantwort:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. Optional: Wechseln Sie zu einer Benutzer-ID, die Mitglied der lokalen `mqm`-Gruppe ist.

   Sie können als Administrator fortfahren, aber für eine realistische Produktionskonfiguration mit einer Benutzer-ID fortfahren, die mehr eingeschränkte Rechte hat. Die Benutzer-ID muss mindestens ein Mitglied der lokalen `mqm`-Gruppe sein.

   Wenn die IBM MQ-Installation als Teil einer Domäne konfiguriert ist, machen Sie die Benutzer-ID zu einem Mitglied der Gruppe `Domain mqm`. Der Assistent "IBM MQ vorbereiten " macht die globale Gruppe `Domain mqm` zu einem Mitglied der lokalen `mqm`-Gruppe, sodass Sie die Benutzer-ID nicht direkt zu einem Mitglied der lokalen `mqm`-Gruppe machen müssen.

5. Erstellen Sie den Warteschlangenmanager.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

   Systemantwort:

```
IBM MQ queue manager created.
Directory 'c:\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

6. Stellen Sie sicher, dass sich die vom Warteschlangenmanager erstellten Verzeichnisse im Verzeichnis *c:\wmq* befinden.

```
dir c:\wmq/D /B /S
```

7. Stellen Sie sicher, dass die Dateien Lese-und Schreibberechtigung oder vollständige Steuerungsberechtigung für die lokale mqm-Gruppe haben.

```
cacls c:\wmq\*.*
```

## Nächste Schritte

Testen Sie den Warteschlangenmanager, indem Sie eine Nachricht in eine Warteschlange einreihen und eine Nachricht erhalten.

1. Starten Sie den Warteschlangenmanager.

```
strmqm QMGR
```

   Systemantwort:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Erstellen Sie eine Testwarteschlange.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

   Systemantwort:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
```

```
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. Reihen Sie eine Testnachricht mit dem Beispielprogramm **amqsput**ein.

```
echo 'A test message' | amqsput QTEST QMGR
```

Systemantwort:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. Rufen Sie die Testnachricht mit dem Beispielprogramm **amqsget**ab.

```
amqsget QTEST QMGR
```

Systemantwort:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. Stoppen Sie den Warteschlangenmanager.

```
endmqm -i QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. Löschen Sie den Warteschlangenmanager.

```
dltmqm QMGR
```

Systemantwort:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Löschen Sie die von Ihnen erstellten Verzeichnisse.

**Tipp:** Fügen Sie die Option /Q zu den Befehlen hinzu, um zu verhindern, dass der Befehl zum Löschen jeder Datei oder jedes Verzeichnisses auffordert.

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

**Zugehörige Konzepte**

„Windows-Domänen und Multi-Instanz-Warteschlangenmanager” auf Seite 550
Ein Multi-Instanz-Warteschlangenmanager unter Windows erfordert die gemeinsame Nutzung seiner Daten und Protokolle. Die Freigabe muss für alle Instanzen des Warteschlangenmanagers, die auf verschiedenen Servern oder Workstations ausgeführt werden, zugänglich sein. Konfigurieren Sie die Warteschlangenmanager und nutzen Sie sie als Teil einer Windows-Domäne gemeinsam. Der Warteschlangenmanager

kann auf einer Domänenworkstation oder einem Server oder auf dem Domänencontroller ausgeführt werden.

**Zugehörige Tasks**

Windows Lesen und Schreiben von Daten-und Protokolldateien, die von einer alternativen lokalen Sicherheitsgruppe autorisiert sind
Diese Task zeigt, wie das Flag -a im **crtmqm** -Befehl verwendet wird. Die Markierung stellt dem Warteschlangenmanager eine alternative lokale Sicherheitsgruppe zur Verfügung, mit der er Zugriff auf seine Protokoll-und Datendateien erhält.

„Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind" auf Seite 565

„WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen" auf Seite 551

Windows *Lesen und Schreiben von Daten-und Protokolldateien, die von einer alternativen lokalen Sicherheitsgruppe autorisiert sind*
Diese Task zeigt, wie das Flag -a im **crtmqm** -Befehl verwendet wird. Die Markierung stellt dem Warteschlangenmanager eine alternative lokale Sicherheitsgruppe zur Verfügung, mit der er Zugriff auf seine Protokoll-und Datendateien erhält.

## Vorbereitende Schritte

1. IBM MQ for Windows als primäre Installation installieren.
2. Führen Sie die Prepare IBM MQ Wizard aus.

   Weitere Informationen finden Sie im Abschnitt IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren.

   Konfigurieren Sie für diese Task die Installation entweder für die Ausführung mit einer lokalen Benutzer-ID oder für eine Domänenbenutzer-ID. Beachten Sie allerdings, dass die Installation später ohnehin für eine Domäne konfiguriert werden muss, um alle Tasks von „Windows-Domänen und Multi-Instanz-Warteschlangenmanager" auf Seite 550 ausführen zu können.
3. Melden Sie sich mit Administratorberechtigung an, um den ersten Teil der Task auszuführen.

## Informationen zu diesem Vorgang

Diese Task ist eine von einer Gruppe zusammengehöriger Tasks, die den Zugriff auf Warteschlangenmanagerdaten und Protokolldateien veranschaulichen. Die Tasks zeigen, wie ein Warteschlangenmanager erstellt wird, der berechtigt ist, Daten und Protokolldateien zu lesen und zu schreiben, die in einem Verzeichnis Ihrer Wahl gespeichert sind. Sie begleiten die Task „Windows-Domänen und Multi-Instanz-Warteschlangenmanager" auf Seite 550.

Unter Windows können Sie als Standardpfad für Daten- und Protokolldateien für einen IBM MQ for Windows beliebige Verzeichnisse festlegen. Der Installations-und Konfigurationsassistent erteilt der lokalen `mqm` -Gruppe und der Benutzer-ID, die die Warteschlangenmanagerprozesse ausführt, automatisch Zugriff auf die Verzeichnisse. Wenn Sie einen Warteschlangenmanager erstellen, der unterschiedliche Verzeichnisse für WS-Manager-Daten und -Protokolldateien angibt, müssen Sie die vollständige Steuerberechtigung für die Verzeichnisse konfigurieren.

In diesem Beispiel stellen Sie dem Warteschlangenmanager eine alternative lokale Sicherheitsgruppe zur Verfügung, die über die vollständige Steuerungsberechtigung für die Verzeichnisse verfügt. Die alternative Sicherheitsgruppe erteilt dem Warteschlangenmanager die Berechtigung zum Verwalten von Dateien im Verzeichnis. Der primäre Zweck der alternativen Sicherheitsgruppe besteht darin, eine alternative globale Sicherheitsgruppe zu autorisieren. Verwenden Sie eine alternative globale Sicherheitsgruppe, um einen WS-Manager mit mehreren Instanzen einzurichten. In diesem Beispiel konfigurieren Sie eine lokale Gruppe, um sich mit der Verwendung einer alternativen Sicherheitsgruppe vertraut zu machen, ohne IBM MQ in einer Domäne installieren zu müssen. Es ist ungewöhnlich, eine lokale Gruppe als alternative Sicherheitsgruppe zu konfigurieren.

Der Befehl **crtmqm** erstellt einen Warteschlangenmanager, der automatisch beim Start der Workstation durch den IBM MQ-Service gestartet wird.

Die Task ist anschaulich. Sie verwendet bestimmte Werte, die Sie ändern können. Die Werte, die Sie ändern können, sind in Kursivschrift. Führen Sie am Ende der Task die Anweisungen aus, um alle Änderungen zu entfernen, die Sie vorgenommen haben.

## Vorgehensweise

1. Richten Sie eine alternative Sicherheitsgruppe ein.

   Die alternative Sicherheitsgruppe ist normalerweise eine Domänengruppe. In diesem Beispiel erstellen Sie einen Warteschlangenmanager, der eine lokale alternative Sicherheitsgruppe verwendet. Wenn Sie eine lokale alternative Sicherheitsgruppe verwenden, können Sie die Task mit einer IBM MQ-Installation ausführen, die nicht Teil einer Domäne ist.

   a) Führen Sie den Befehl **lusrmgr.msc** aus, um das Fenster "Lokale Benutzer und Gruppen" zu öffnen.

   b) Klicken Sie mit der rechten Maustaste auf **Gruppen** > **Neue Gruppe...**

   c) Geben Sie im Feld **Gruppenname** *altmqm* ein und klicken Sie auf **Erstellen** > **Schließen**.

   d) Identifizieren Sie die Benutzer-ID, die den IBM MQ-Service ausführt.

      i) Klicken Sie auf **Start** > **Ausführen ...**, Geben Sie `services.msc` ein und klicken Sie auf **OK**.

      ii) Klicken Sie in der Liste der Services auf den IBM MQ-Service, und klicken Sie auf die Registerkarte 'Anmelden'.

      iii) Merken Sie sich die Benutzer-ID, und schließen Sie den Service-Explorer

   e) Fügen Sie die Benutzer-ID, die den IBM MQ-Service ausführt, der Gruppe *altmqm* hinzu. Fügen Sie außerdem die Benutzer-ID hinzu, mit der Sie sich anmelden, um einen Warteschlangenmanager zu erstellen, und führen Sie ihn interaktiv aus.

      Windows überprüft die Zugriffsberechtigung des Warteschlangenmanagers auf die Daten- und Protokollverzeichnisse, indem er die Berechtigung der Benutzer-ID überprüft, die Warteschlangenmanagerprozesse ausführt. Die Benutzer-ID muss direkt oder indirekt über eine globale Gruppe ein Mitglied der *altmqm*-Gruppe, die die Verzeichnisse autorisiert hat, sein.
      Wenn Sie IBM MQ als Teil einer Domäne installiert haben, und Tasks in „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen” auf Seite 551 ausführen werden, lauten die Domänen-Benutzer-IDs in „Active Directory- und DNS-Domäne unter Windows erstellen” auf Seite 555 *wmquser1* únd *wmquser2*.
      Wenn Sie den Warteschlangenmanager nicht als Teil einer Domäne installiert haben, lautet die standardmäßige lokale Benutzer-ID, unter der der IBM MQ-Service ausgeführt wird, `MUSR_MQADMIN`. Wenn Sie die Tasks ohne Administratorberechtigung ausführen möchten, erstellen Sie einen Benutzer, der Mitglied der lokalen `mqm`-Gruppe ist.
      Befolgen Sie diese Schritte, um *wmquser1* und *wmquser2* zu *altmqm* hinzuzufügen. Wenn Ihre Konfiguration anders ist, ersetzen Sie Ihre Namen durch die Benutzer-IDs und die Gruppe.

      i) Klicken Sie in der Liste der Gruppen mit der rechten Maustaste auf **altmqm** > **Eigenschaften** > **Hinzufügen...**.

      ii) Geben Sie im Fenster "Benutzer, Computer oder Gruppen auswählen" *wmquser1 ; wmquser2* ein und klicken Sie auf **Namen prüfen**.

      iii) Geben Sie den Namen und das Kennwort eines Domänenadministrators in das Fenster "Windows Security" ein und klicken Sie dann auf **OK** > **OK** > **Anwenden** > **OK**.

2. Öffnen Sie eine Eingabeaufforderung.
3. Starten Sie den IBM MQ-Service erneut.

   Sie müssen den Service erneut starten, damit die Benutzer-ID, unter der er ausgeführt wird, die zusätzlichen Sicherheitsberechtigungsnachweise erhält, die Sie für die Benutzer-ID konfiguriert haben.

Geben Sie die Befehle ein:

```
endmqsvc
strmqsvc
```

Die Systemantworten:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' ended successfully.
```

Und:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' started successfully.
```

4. Geben Sie den folgenden Befehl ein:

```
md c:\wmq\data, c:\wmq\logs
```

5. Legen Sie die Berechtigungen für die Verzeichnisse fest, um dem lokalen Benutzer *user* Lese- und Schreibzugriff zu erteilen.

```
cacls c:\wmq/T /E /G altmqm:F
```

Systemantwort:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

6. Optional: Wechseln Sie zu einer Benutzer-ID, die Mitglied der lokalen mqm-Gruppe ist.

   Sie können als Administrator fortfahren, aber für eine realistische Produktionskonfiguration mit einer Benutzer-ID fortfahren, die mehr eingeschränkte Rechte hat. Die Benutzer-ID muss mindestens ein Mitglied der lokalen mqm-Gruppe sein.

   Wenn die IBM MQ-Installation als Teil einer Domäne konfiguriert ist, machen Sie die Benutzer-ID zu einem Mitglied der Gruppe Domain mqm. Der Assistent "IBM MQ vorbereiten " macht die globale Gruppe Domain mqm zu einem Mitglied der lokalen mqm-Gruppe, sodass Sie die Benutzer-ID nicht direkt zu einem Mitglied der lokalen mqm-Gruppe machen müssen.

7. Erstellen Sie den Warteschlangenmanager.

```
crtmqm -a altmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

Systemantwort:

```
IBM MQ queue manager created.
Directory 'c:\wmq1\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

8. Stellen Sie sicher, dass sich die vom Warteschlangenmanager erstellten Verzeichnisse im Verzeichnis *c:\wmq* befinden.

```
dir c:\wmq/D /B /S
```

9. Stellen Sie sicher, dass die Dateien Lese-und Schreibberechtigung oder vollständige Steuerungsberechtigung für die lokale mqm-Gruppe haben.

```
cacls c:\wmq\*.*
```

## Nächste Schritte

Testen Sie den Warteschlangenmanager, indem Sie eine Nachricht in eine Warteschlange einreihen und eine Nachricht erhalten.

1. Starten Sie den Warteschlangenmanager.

```
strmqm QMGR
```

   Systemantwort:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Erstellen Sie eine Testwarteschlange.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

   Systemantwort:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.


1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. Reihen Sie eine Testnachricht mit dem Beispielprogramm **amqsput**ein.

```
echo 'A test message' | amqsput QTEST QMGR
```

   Systemantwort:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. Rufen Sie die Testnachricht mit dem Beispielprogramm **amqsget**ab.

```
amqsget QTEST QMGR
```

Systemantwort:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. Stoppen Sie den Warteschlangenmanager.

   ```
   endmqm -i QMGR
   ```

   Systemantwort:

   ```
   IBM MQ queue manager 'QMGR' ending.
   IBM MQ queue manager 'QMGR' ended.
   ```

6. Löschen Sie den Warteschlangenmanager.

   ```
   dltmqm QMGR
   ```

   Systemantwort:

   ```
   IBM MQ queue manager 'QMGR' deleted.
   ```

7. Löschen Sie die von Ihnen erstellten Verzeichnisse.

   **Tipp:** Fügen Sie die Option /Q zu den Befehlen hinzu, um zu verhindern, dass der Befehl zum Löschen jeder Datei oder jedes Verzeichnisses auffordert.

   ```
   del /F /S C:\wmq\*.*
   rmdir /S C:\wmq
   ```

**Zugehörige Tasks**

Windows Lesen und Schreiben von Daten und Protokolldateien, die von der lokalen mqm-Gruppe autorisiert sind

Die Task veranschaulicht, wie ein Warteschlangenmanager mit seinen Daten erstellt wird und die Dateien in einem beliebigen Verzeichnis Ihrer Wahl gespeichert werden. Der Zugriff auf die Dateien wird durch die lokale mqm -Gruppe gesichert. Das Verzeichnis wird nicht gemeinsam genutzt.

Linux *Warteschlangenmanager mit mehreren Instanzen unter Linux erstellen*

Ein Beispiel, das zeigt, wie ein Warteschlangenmanager mit mehreren Instanzen unter Linuxeingerichtet wird. Die Konfiguration ist klein, um die Konzepte zu veranschaulichen. Das Beispiel basiert auf Linux Red Hat Enterprise 5. Die Schritte unterscheiden sich auf anderen UNIX-Plattformen.

## Informationen zu diesem Vorgang

Das Beispiel wird auf einem 2-GHz-Notebook-Computer mit 3 GB RAM, der Windows 7 Service-Pack 1 ausführt, konfiguriert. Zwei virtuelle VMware-Maschinen, Server1 und Server2, führen Linux Red Hat Enterprise 5 in 640 MB Images aus. Server1 enthält das Network File System (NFS), die WS-Manager-Protokolle und eine HA-Instanz. Es ist nicht üblich, dass der NFS-Server auch eine der WS-Manager-Instanzen hosten kann. Dies ist die Vereinfachung des Beispiels. Server2 hängt die Protokolle des Server1-Warteschlangenmanagers mit einer Standby-Instanz an. Ein MQI-Client von WebSphere wird auf einem zusätzlichen 400-MB-VMware-Image installiert, auf dem Windows 7 Service Pack 1 und die Anwendungen mit

hoher Verfügbarkeit ausgeführt werden. Alle virtuellen Maschinen werden aus Sicherheitsgründen als Teil eines VMware-Host-Netzes konfiguriert.

**Anmerkung:** Sie sollten nur WS-Manager-Daten auf einem NFS-Server in die Warteschlange stellen. Verwenden Sie im -NFS die folgenden drei Optionen mit dem Befehl mount, um das System sicher zu machen:

- **noexec**
  Wenn Sie diese Option verwenden, stoppen Sie die Ausführung von Binärdateien auf dem NFS, wodurch verhindert wird, dass ein ferner Benutzer nicht mehr benötigten Code auf dem System ausführen kann.
- **nosuid**
  Wenn Sie diese Option verwenden, verhindern Sie die Verwendung der Bits "set-user-identifier" und "set-group-identifier bits", die verhindert, dass ein ferner Benutzer höhere Berechtigungen erhält.
- **nodev**
  Wenn Sie diese Option verwenden, stoppen Sie die Zeichen-und Blockspezial-Einheiten, die verwendet oder definiert werden, wodurch verhindert wird, dass ein ferner Benutzer aus einem chroot-Gefängnis heraus kommt.

## Vorgehensweise

1. Melden Sie sich als Root an.
2. Lesen Sie Installieren von IBM MQ-Übersicht und folgen Sie dem entsprechenden Link, um IBM MQ zu installieren, den Benutzer und die Gruppe 'mqm' zu erstellen und `/var/mqm` zu definieren.
3. Führen Sie die Task Verhalten des gemeinsam genutzten Dateisystems überprüfen aus, um zu überprüfen, ob das Dateisystem mehrere Instanzen von Warteschlangenmanagern mit mehreren Instanzen unterstützt.
4. Führen Sie für Server1 den folgenden Schritt aus:
   a. Erstellen Sie Protokoll- und Datenverzeichnisse in einem allgemeinen Ordner (`/MQHA`), der gemeinsam genutzt werden soll. For example:
      i) **mkdir** */MQHA*
      ii) **mkdir** */MQHA/logs*
      iii) **mkdir** */MQHA/qmgrs*
5. Führen Sie für Server2 den folgenden Schritt aus:
   a. Erstellen Sie den Ordner `/MQHA`, um das gemeinsam genutzte Dateisystem zu mounten. Behalten Sie den Pfad auf dem Pfad zu Server1. For example:
      i) **mkdir** */MQHA*
6. Stellen Sie sicher, dass die `MQHA` -Verzeichnisse Eigentum von Benutzer und Gruppe mqm sind, und die Zugriffsberechtigungen für Benutzer und Gruppe auf `rwx` gesetzt sind. Beispiel: **`ls -al`** zeigt `drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA` an.
   a. **chown -R** `mqm:mqm` */MQHA*
   b. **chmod -R** `ug+rwx` */MQHA*
7. Erstellen Sie den Warteschlangenmanager, indem Sie den Befehl **crtmqm -ld** */MQHA/logs* **-md** */MQHA/qmgrs QM1* eingeben.
8. Hinzufügen[2] `/MQHA *(rw,sync,no_wdelay,fsid=0)` bis `/etc/exports`
9. Führen Sie für Server1 die folgenden Schritte aus:
   a. Starten Sie den NFS -Dämon: */etc/init.d/* **nfs** `start`
   b. Kopieren Sie die Konfigurationsdetails des Warteschlangenmanagers von Server1:

[2] Der '*' ermöglicht es allen Maschinen, die diesen einen Mount/MQHA für Lese-/Schreibvorgänge erreichen können. Schränken Sie den Zugriff auf eine Produktionsmaschine ein.

```
dspmqinf -o command QM1
```

und kopieren Sie das Ergebnis in die Zwischenablage:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

10. Führen Sie für Server2 die folgenden Schritte aus:
    a. Hängen Sie das exportierte Dateisystem /MQHA an, indem Sie den Befehl **mount -t** nfs4 **-o** hard,intr *Server1:/ /MQHA* eingeben.
    b. Fügen Sie den Konfigurationsbefehl des Warteschlangenmanagers in Server2 ein:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

11. Starten Sie die Warteschlangenmanagerinstanzen in einer beliebigen Reihenfolge mit dem Parameter-**x** : **strmqm** -**x** QM1.

    Der zum Starten der Warteschlangenmanagerinstanzen verwendete Befehl muss von derselben IBM MQ-Installation wie der Befehl **addmqinf** ausgegeben werden. Wenn Sie den Warteschlangenmanager von einer anderen Installation aus starten und stoppen möchten, müssen Sie zuerst die dem Warteschlangenmanager zugeordnete Installation mit dem Befehl **setmqm** festlegen. Weitere Informationen finden Sie in setmqm .

Linux *Multi-Instanz-Warteschlangenmanager unter Linux überprüfen*

Verwenden Sie die Beispielprogramme **amqsghac**, **amqsphac** und **amqsmhac** , um die Konfiguration eines Multi-Instanz-Warteschlangenmanagers zu überprüfen. Dieser Abschnitt enthält eine Beispielkonfiguration für die Überprüfung einer Konfiguration mit einem Multi-Instanz-Warteschlangenmanager unter Linux Red Hat Enterprise 5.

Die Beispielprogramme für hohe Verfügbarkeit verwenden die automatische Clientwiederverbindung. Wenn der verbundene Warteschlangenmanager ausfällt, versucht der Client, die Verbindung zu einem WS-Manager in derselben Warteschlangenmanager-Gruppe wieder herzustellen. Die Beschreibung der Beispiele Hochverfügbarkeits-Musterprogramme veranschaulicht die Clientwiederverbindung unter Verwendung eines einzigen Instanz-WS-Managers für die Einfachheit. Sie können dieselben Muster mit Warteschlangenmanagern mit mehreren Instanzen verwenden, um eine Konfiguration mit mehreren Instanzen des Warteschlangenmanagers zu überprüfen.

Im Beispiel wird die Konfiguration mit mehreren Instanzen verwendet, die in „Warteschlangenmanager mit mehreren Instanzen unter Linux erstellen” auf Seite 592 beschrieben wird. Überprüfen Sie mit der Konfiguration, ob der Warteschlangenmanager mit mehreren Instanzen in die Standby-Instanz umschaltet. Stoppen Sie den Warteschlangenmanager mit dem Befehl **endmqm** und verwenden Sie die Option -s(Switchover). Die Clientprogramme stellen die Verbindung zur neuen Warteschlangenmanagerinstanz wieder her und arbeiten nach einer geringfügigen Verzögerung weiterhin mit der neuen Instanz.

In dem Beispiel wird der Client auf einem System mit Windows 7 Service Pack 1 ausgeführt. Das System dient als Host für zwei VMware Linux-Server, auf denen der Multi-Instanz-Warteschlangenmanager ausgeführt wird.

### Funktionsübernahme mit IBM MQ Explorer überprüfen

Bevor Sie die Beispielanwendungen verwenden, um die Funktionsübernahme zu überprüfen, führen Sie den IBM MQ Explorer auf jedem Server aus. Fügen Sie jedem Explorer die beiden WS-Manager-Instanzen hinzu, indem Sie den Assistenten **Remote Queue Manager hinzufügen > Direkt mit einem Multi-In-**

**stanz-Warteschlangenmanager verbinden** verwenden. Stellen Sie sicher, dass beide Instanzen aktiv sind und den Standby-Modus zulassen. Schließen Sie das Fenster, in dem das VMware-Image ausgeführt wird, mit der aktiven Instanz, den Server virtuell ausschalten oder die aktive Instanz stoppen, um die Umschaltung auf die Standby-Instanz zu ermöglichen.

**Anmerkung:** Wenn Sie den Server ausschalten, stellen Sie sicher, dass es sich nicht um jenen handelt, der /MQHA hostet!

**Anmerkung:** Die Option **Umschalten auf eine Standby-Instanz zulassen** ist unter Umständen nicht im Dialog **Warteschlangenmanager stoppen** verfügbar. Die Option fehlt, weil der WS-Manager als einzelner Instanz-Warteschlangenmanager ausgeführt wird. Sie müssen diese Option ohne die Option **Standby-Instanz zulassen** gestartet haben. Wenn Ihre Anforderung zum Stoppen des Warteschlangenmanagers zurückgewiesen wird, sehen Sie sich das Fenster **Details** an, da möglicherweise keine Standby-Instanz aktiv ist.

#### Funktionsübernahme mit den Musterprogrammen überprüfen

**Wählen Sie einen Server aus, der die aktive Instanz ausführen soll.**
Möglicherweise haben Sie einen der Server ausgewählt, um das MQHA-Verzeichnis oder das Dateisystem zu hosten. Wenn Sie die Funktionsübernahme testen möchten, indem Sie das VMware-Fenster, auf dem der aktive Server ausgeführt wird, schließen, stellen Sie sicher, dass es sich nicht um den handelt, der MQHA hostet!

**Auf dem Server, auf dem die aktive WS-Manager-Instanz ausgeführt**

**Anmerkung:** Wenn Sie den SVRCONN -Kanal mit dem MCAUSER auf mqm ausführen, können Sie die Anzahl der Konfigurationsschritte im Beispiel reduzieren. Wenn eine andere Benutzer-ID ausgewählt ist und Ihr System anders konfiguriert ist als in dem Beispiel, können Sie Probleme mit der Zugriffsberechtigung haben. Verwenden Sie mqm nicht als MCAUSER auf einem exponierten System; es ist wahrscheinlich, dass die Sicherheit erheblich beeinträchtigt wird.

1. Ändern Sie *ipaddr1* und *ipaddr2* und speichern Sie die folgenden Befehle in /MQHA/hasamples.tst.

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN)  TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2
(1414)') QMNAME(QM1) REPLACE
START  CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START  LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. Öffnen Sie ein Terminalfenster mit dem Pfad /MQHA und führen Sie den folgenden Befehl aus:

```
runmqsc -m QM1 < hasamples.tst
```

3. Überprüfen Sie, ob der Listener aktiv ist und die Steuerung des Warteschlangenmanagers hat, indem Sie die Ausgabe des Befehls **runmqsc** überprüfen.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

Oder verwenden Sie den IBM MQ Explorer, der vom TCPIP-Listener ausgeführt wird, und auf dem Control = Queue Manager eingestellt ist.

**Auf dem Client**

1. Kopieren Sie die Clientverbindungstabelle AMQCLCHL.TAB von /MQHA/qmgrs/QM1.000/@ipcc auf dem Server auf den C:\ auf dem Client.

2. Öffnen Sie eine Eingabeaufforderung mit dem Pfad `C:\` , und legen Sie die Umgebungsvariable `MQCHLLIB` so fest, dass sie auf die Definitionstabelle für den Clientkanal (CCDT) verweist.

```
SET MQCHLLIB=C:\
```

3. Geben Sie an der Eingabeaufforderung die folgenden Befehle ein:

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

**Auf dem Server, auf dem die aktive WS-Manager-Instanz ausgeführt**

1. Entweder:
   - Schließen Sie das Fenster, in dem das VMware-Image mit der aktiven Serverinstanz ausgeführt wird.
   - Stoppen Sie mit dem IBM MQ Explorer die aktive Warteschlangenmanagerinstanz; lassen Sie dabei das Umschalten auf die Standby-Instanz zu und weisen Sie die wiederverbindungsfähigen Clients an, die Verbindung wiederherzustellen.
2. Die drei Clients erkennen schließlich, dass die Verbindung unterbrochen ist, und stellen Sie dann die Verbindung wieder her. Wenn Sie in dieser Konfiguration das Serverfenster schließen, wird für alle drei Verbindungen, die neu aufgebaut werden sollen, ungefähr sieben Minuten Zeit. Einige Verbindungen werden vor anderen wieder hergestellt.

**Ergebnisse**

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

### Multi *Multi-Instanz-WS-Manager löschen*

Wenn Sie auf Multiplatforms einen Warteschlangenmanager mit mehreren Instanzen vollständig löschen möchten, verwenden Sie den Befehl **dltmqm** , um den Warteschlangenmanager zu löschen, und entfernen Sie dann Instanzen von anderen Servern mit dem Befehl **rmvmqinf** oder **dltmqm** .

Führen Sie den Befehl **dltmqm** aus, um einen Warteschlangenmanager zu löschen, für den Instanzen auf anderen Servern definiert sind, auf jedem Server, auf dem dieser Warteschlangenmanager definiert ist. Sie müssen den Befehl **dltmqm** nicht auf demselben Server ausführen, auf dem er erstellt wurde. Führen Sie anschließend den Befehl **rmvmqinf** oder **dltmqm** auf allen anderen Servern aus, die über eine Definition des Warteschlangenmanagers verfügen.

Sie können einen WS-Manager nur löschen, wenn er gestoppt wurde. Zu dem Zeitpunkt, zu dem Sie es löschen, werden keine Instanzen ausgeführt, und der Warteschlangenmanager, genau genommen, ist weder ein einzelner noch ein Warteschlangenmanager mit mehreren Instanzen. Es handelt sich lediglich um einen Warteschlangenmanager, der über die zugehörigen WS-Manager-Daten und Protokolle auf einem fernen freigesetzten Verzeichnis verfügt. Wenn Sie einen Warteschlangenmanager löschen, werden seine Warteschlangenmanagerdaten und -protokolle gelöscht und die Zeilengruppe für den Warteschlangenmanager wird aus der Datei `mqs.ini` auf dem Server entfernt, auf dem Sie den Befehl **dltmqm** ausgegeben haben. Sie müssen Zugriff auf die Netzfreigabe haben, die die Daten und Protokolle des WS-Managers enthält, wenn Sie den Warteschlangenmanager löschen.

Auf anderen Servern, auf denen Sie zuvor Instanzen des Warteschlangenmanagers erstellt haben, befinden sich auch Einträge in den `mqs.ini`-Dateien auf diesen Servern. Sie müssen jeden Server nacheinander besuchen und die Zeilengruppe des Warteschlangenmanagers entfernen, indem Sie den Befehl **rmvmqinf** *Zeilengruppenname des Warteschlangenmanagers*ausführen.

Linux AIX Wenn Sie auf AIX and Linux-Systemen eine allgemeine `mqs.ini`-Datei in den Netzspeicher gestellt und von allen Servern referenziert haben, indem Sie die Umgebungsvariable `AMQ_MQS_INI_LOCATION` auf jedem Server definieren, müssen Sie den Warteschlangenmanager nur von einem seiner Server löschen, da nur eine `mqs.ini`-Datei zu aktualisieren ist.

**Beispiel**

**Erster Server**

**dltmqm** QM1

**Andere Server, auf denen Instanzen definiert sind**

**rmvmqinf** QM1 , oder

**dltmqm** QM1

### Multi *Starten und Stoppen eines Multi-Instanz-Warteschlangenmanagers*

Starten und Stoppen eines auf mehreren Plattformen konfigurierten Warteschlangenmanagers entweder als einzelne Instanz oder als Warteschlangenmanager mit mehreren Instanzen.

Wenn Sie einen Warteschlangenmanager mit mehreren Instanzen auf einem Serverpaar definiert haben, können Sie den Warteschlangenmanager entweder in einem einzigen Instanz-Warteschlangenmanager oder als WS-Manager mit mehreren Instanzen auf beiden Servern ausführen.

Um einen Warteschlangenmanager mit mehreren Instanzen auszuführen, starten Sie den Warteschlangenmanager mit dem Befehl **strmqm** -x *QM1* auf einem der Server. Die Option -x ermöglicht der Instanz die Funktionsübernahme. Es wird zur *aktiven Instanz* . Starten Sie die Standby-Instanz auf dem anderen Server mit demselben Befehl **strmqm** -x *QM1* . Die Option -x ermöglicht den Start der Instanz als Standby-Instanz.

Der WS-Manager wird jetzt mit einer aktiven Instanz ausgeführt, die alle Anforderungen verarbeitet, und eine Standby-Instanz, die bereit ist, zu übernehmen, wenn die aktive Instanz fehlschlägt. Der aktiven Instanz wird exklusiven Zugriff auf die Daten und Protokolle des Warteschlangenmanagers gewährt. Der Standby-Wartestatus wartet auf exklusiven Zugriff auf die Daten und Protokolle des Warteschlangenmanagers. Wenn der Standby-Server exklusiven Zugriff gewährt wird, wird er zur aktiven Instanz.

Sie können die Steuerung auch manuell zur Standby-Instanz wechseln, indem Sie den Befehl **endmqm** -s für die aktive Instanz absetzen. Der Befehl **endmqm** -s beendet die aktive Instanz, ohne die Bereitschaftsdatenbank herunterzufahren. Die exklusive Zugriffssperre für die WS-Manager-Daten und -Protokolle wird freigegeben, und die Bereitschaftsdatenbank übernimmt die Verwaltung.

Sie können auch einen Warteschlangenmanager starten und stoppen, der mit mehreren Instanzen auf verschiedenen Servern als einzelner Instanz-WS-Manager konfiguriert ist. Wenn Sie den Warteschlangenmanager starten, ohne die Option -x im Befehl **strmqm** zu verwenden, werden die Instanzen des auf anderen Maschinen konfigurierten Warteschlangenmanagers nicht als Standby-Instanzen gestartet. Wenn Sie versuchen, eine andere Instanz zu starten, erhalten Sie die Antwort, dass die WS-Manager-Instanz nicht als Standby-Instanz ausgeführt werden darf.

Wenn Sie die aktive Instanz eines Multi-Instanz-Warteschlangenmanagers mit dem Befehl **endmqm** ohne die Option -s stoppen, werden sowohl die aktive als auch die Standby-Instanz gestoppt. Wenn Sie die Standby-Instanz mit dem Befehl **endmqm** mit der Option -x stoppen, wird sie als Standby-Instanz gestoppt und die aktive Instanz wird weiter ausgeführt. Sie können **endmqm** nicht ohne die Option -x in der Bereitschaftsdatenbank absetzen.

Es können nur zwei WS-Manager-Instanzen gleichzeitig ausgeführt werden. Eine Instanz ist die aktive Instanz, und die andere Instanz ist eine Standby-Instanz. Wenn Sie zwei Instanzen gleichzeitig starten, hat IBM MQ keine Kontrolle darüber, welche Instanz zur aktiven Instanz wird. Sie wird durch das Netzdateisystem bestimmt. Die erste Instanz, die exklusiven Zugriff auf die WS-Manager-Daten erhält, wird zum aktiven Exemplar.

**Anmerkung:** Bevor Sie einen fehlgeschlagenen Warteschlangenmanager erneut starten, müssen Sie die Verbindung zu den Anwendungen von dieser Instanz des Warteschlangenmanagers trennen. Ist dies nicht der Fall, wird der WS-Manager möglicherweise nicht ordnungsgemäß erneut gestartet.

### Multi *Gemeinsam genutztes Dateisystem*

Auf Mulitplatforms verwendet ein Warteschlangenmanager mit mehreren Instanzen ein Netzdateisystem, um Warteschlangenmanagerinstanzen zu verwalten.

Ein Warteschlangenmanager mit mehreren Instanzen automatisiert die Funktionsübernahme mit einer Kombination aus Dateisystemsperren und gemeinsam genutzten WS-Manager-Daten und -Protokollen. Nur eine Instanz eines Warteschlangenmanagers kann exklusiven Zugriff auf die Daten und Protokolle des gemeinsam genutzten Warteschlangenmanagers haben. Wenn er Zugriff erhält, wird er zur aktiven Instanz. Die andere Instanz, die den exklusiven Zugriff nicht erhält, wartet als Standby-Instanz, bis die WS-Manager-Daten und -Protokolle verfügbar werden.

Das Netzdateisystem ist für die Freigabe der Sperren verantwortlich, die er für die aktive Warteschlangenmanagerinstanz enthält. Wenn die aktive Instanz in irgendeiner Weise ausfällt, gibt das vernetzte Dateisystem die Sperren frei, die sie für die aktive Instanz hält. Sobald die exklusive Sperre freigegeben wird, wartet ein Standby-Warteschlangenmanager, der auf die Sperre wartet, um die Sperre zu erhalten. Ist sie erfolgreich, wird sie zur aktiven Instanz und verfügt über exklusiven Zugriff auf die Daten und Protokolle des Warteschlangenmanagers und die Protokolle im gemeinsam genutzten Dateisystem. Anschließend wird die Verarbeitung fortgesetzt.

Das zugehörige Thema Unterstützung der Planungsdateisysteme beschreibt, wie Sie konfigurieren und überprüfen, ob Ihr Dateisystem mehrere Instanzen von Warteschlangenmanagern unterstützt.

Ein WS-Manager mit mehreren Instanzen schützt Sie nicht vor einem Fehler im Dateisystem. Es gibt eine Reihe von Möglichkeiten, Ihre Daten zu schützen.

- Investieren Sie in einen zuverlässigen Speicher, wie z. B. redundante Platteneinheiten (RAID), und schließen Sie sie in ein Netzdateisystem ein, das über Netzausfallsicherheit verfügt.
- Sichern Sie die linearen IBM MQ-Protokolle auf alternativen Datenträgern. Wenn Ihr primärer Protokolldatenträger ausfällt, können Sie dann die Protokolle auf den alternativen Datenträgern wiederherstellen. Sie können einen Sicherungswarteschlangenmanager verwenden, um diesen Prozess zu verwalten.

## Multi *Mehrere WS-Manager-Instanzen*

Ein Warteschlangenmanager mit mehreren Instanzen ist belastbar, da er eine Standby-WS-Manager-Instanz verwendet, um die Verfügbarkeit des Warteschlangenmanagers nach einem Fehler wiederherzustellen.

Das Replizieren von Warteschlangenmanagerinstanzen ist eine sehr effektive Möglichkeit, die Verfügbarkeit von Warteschlangenmanagerprozessen zu verbessern. Verwenden Sie ein einfaches Verfügbarkeitsmodell, nur zur Veranschaulichung: Wenn die Zuverlässigkeit einer Instanz eines Warteschlangenmanagers 99% beträgt (über ein Jahr beträgt die kumulative Ausfallzeit 3,65 Tage), erhöht das Hinzufügen einer weiteren Instanz des Warteschlangenmanagers die Verfügbarkeit auf 99,99% (über ein Jahr, kumulative Ausfallzeit von etwa einer Stunde).

Dies ist zu einfach ein Modell, um Ihnen praktische numerische Schätzungen der Verfügbarkeit zu geben. Um die Verfügbarkeit realistisch zu modellieren, müssen Sie statistische Daten für die mittlere Zeit zwischen Fehlern (MTBF) und die mittlere Reparaturzeit (MTTR) und die Wahrscheinlichkeitsverteilung der Zeit zwischen Ausfällen und Reparaturzeiten erfassen.

Der Begriff Multi-Instanz-WS-Manager bezeichnet die Kombination aus aktiven und Standby-Instanzen des Warteschlangenmanagers, die die Daten und Protokolle des Warteschlangenmanagers gemeinsam nutzen. Warteschlangenmanager mit mehreren Instanzen schützen Sie vor dem Ausfall von WS-Manager-Prozessen, indem eine Instanz des auf einem Server aktiven WS-Managers und eine weitere Instanz des Warteschlangenmanagers auf einem anderen Server auf einem anderen Server aktiv ist. Diese Instanz kann automatisch übernommen werden, wenn die aktive Instanz fehlschlägt.

## Multi *Failover oder Switchover*

Eine Standby-WS-Manager-Instanz übernimmt die aktive Instanz entweder auf Anforderung (Switchover) oder wenn die aktive Instanz ausfällt (Failover).

- *Switchover* findet statt, wenn eine Standby-Instanz als Antwort auf den Befehl **endmqm** `-s` gestartet wird, der an die aktive Warteschlangenmanagerinstanz ausgegeben wird. Mit den **endmqm** -Parametern `-c`, `-i` oder `-p` können Sie steuern, wie der WS-Manager abrupt gestoppt wird.

  **Anmerkung:** Switchover findet nur statt, wenn bereits eine Standby-Warteschlangenmanagerinstanz gestartet wurde. Der Befehl **endmqm** `-s` gibt die Sperre des aktiven Warteschlangenmanagers frei und ermöglicht das Umschalten: Er startet keine Standby-Warteschlangenmanagerinstanz.
- *Failover* tritt auf, wenn die Sperre für Warteschlangenmanagerdaten, die von der aktiven Instanz gehalten werden, freigegeben wird, weil die Instanz anscheinend unerwartet gestoppt wurde (d. h., ohne dass ein Befehl **endmqm** ausgegeben wurde).

Wenn die Standby-Instanz als aktive Instanz übernimmt, schreibt sie eine Nachricht in das Fehlerprotokoll des Warteschlangenmanagers.

Bei einem Ausfall oder Umschaltern eines WS-Managers werden die wiederverbindbaren Clients automatisch wieder verbunden. Sie müssen das Flag `-r` nicht in den Befehl **endmqm** einschließen, um die Clientverbindungswiederholung anzufordern. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt.

Wenn Sie feststellen, dass eine fehlgeschlagene Instanz nicht erneut gestartet werden kann, obwohl ein Failover aufgetreten ist und die Standby-Instanz aktiv geworden ist, überprüfen Sie, ob Anwendungen, die

lokal mit der fehlgeschlagenen Instanz verbunden sind, die Verbindung zu der fehlgeschlagenen Instanz getrennt haben.

Lokal verbundene Anwendungen müssen eine fehlgeschlagene WS-Manager-Instanz beenden oder trennen, damit die fehlgeschlagene Instanz erneut gestartet werden kann. Alle lokal verbundenen Anwendungen, die gemeinsam genutzte Bindungen verwenden (dies ist die Standardeinstellung), die eine Verbindung zu einer fehlgeschlagenen Instanz enthalten, um zu verhindern, dass die Instanz erneut gestartet wird.

Wenn es nicht möglich ist, die lokal verbundenen Anwendungen zu beenden, oder stellen Sie sicher, dass die Verbindung getrennt wird, wenn die lokale Warteschlangenmanagerinstanz fehlschlägt, sollten Sie die Verwendung isolierter Bindungen in Betracht ziehen. Lokal verbundene Anwendungen, die isolierte Bindungen verwenden, verhindern nicht, dass die lokale WS-Manager-Instanz erneut gestartet wird, auch wenn sie nicht getrennt werden.

### Multi *Kanal- und Clientverbindungswiederholung*

Die Verbindung zwischen Kanal und Client ist ein wesentlicher Bestandteil der Wiederherstellung der Nachrichtenverarbeitung, nachdem eine Standby-Warteschlangenmanagerinstanz aktiv geworden ist.

Instanzen von Warteschlangenmanagern mit mehreren Instanzen werden auf Servern mit unterschiedlichen Netzadressen installiert. Sie müssen IBM MQ-Kanäle und -Clients mit Verbindungsinformationen für alle Warteschlangenmanagerinstanzen konfigurieren. Wenn eine Bereitschaftsdatenbank übernommen wird, werden Clients und Kanäle automatisch mit der neu aktiven WS-Manager-Instanz an der neuen Netzadresse verbunden. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt.

Das Design unterscheidet sich von der Art und Weise, wie Hochverfügbarkeitsumgebungen, wie z. B. HA-CMP, funktionieren HA-CMP stellt eine virtuelle IP-Adresse für den Cluster bereit und überträgt die Adresse an den aktiven Server. Durch die IBM MQ-Verbindungswiederholung werden keine IP-Adressen geändert oder umgeleitet. Es funktioniert, indem die Verbindung unter Verwendung der Netzwerkadressen, die Sie in Kanaldefinitionen und Clientverbindungen definiert haben, wieder hergestellt wird. Als Administrator müssen Sie die Netzadressen in Kanaldefinitionen und Clientverbindungen zu allen Instanzen eines beliebigen Multi-Instanz-Warteschlangenmanagers definieren. Die beste Möglichkeit, Netzadressen für einen Warteschlangenmanager mit mehreren Instanzen zu konfigurieren, hängt von der Verbindung ab:

**WS-Manager-Kanäle**
Das Attribut `CONNAME` von Kanälen ist eine durch Kommas getrennte Liste mit Verbindungsnamen, z. B. `CONNAME('127.0.0.1(1234), 192.0.2.0(4321)')`. Die Verbindungen werden in der Reihenfolge versucht, die in der Verbindungsliste angegeben ist, bis eine Verbindung erfolgreich hergestellt wurde. Wenn keine Verbindung erfolgreich hergestellt wurde, versucht der Kanal, die Verbindung herzustellen.

**Clusterkanäle**

In der Regel ist keine zusätzliche Konfiguration erforderlich, um WS-Manager mit mehreren Instanzen in einem Cluster zu bearbeiten.

Wenn ein Warteschlangenmanager eine Verbindung zu einem Repository-WS-Manager herstellt, erkennt das Repository die Netzadresse des Warteschlangenmanagers. Er bezieht sich auf den `CONNAME` des `CLUSRCVR` -Kanals auf dem Warteschlangenmanager. Bei TCPIP setzt der Warteschlangenmanager den `CONNAME` automatisch, wenn Sie ihn weglassen oder ihn in Leerzeichen konfigurieren. Wenn eine Standby-Instanz übernimmt, ersetzt die IP-Adresse der vorherigen aktiven Instanz die IP-Adresse der vorherigen aktiven Instanz als `CONNAME` .

Wenn dies erforderlich ist, können Sie `CONNAME` manuell mit der Liste der Netzadressen der Warteschlangenmanagerinstanzen konfigurieren.

**Clientverbindungen**
Clientverbindungen können Verbindungslisten oder Warteschlangenmanagergruppen verwenden, um alternative Verbindungen auszuwählen.

Wenn eine Funktionsübernahme erfolgt, nimmt die Verbindungswiederverbindung einige Zeit in Anspruch. Der Standby-WS-Manager muss den Start beenden. Die Clients, die mit dem fehlgeschlagenen Warteschlangenmanager verbunden waren, müssen den Verbindungsfehler feststellen und eine neue Clientverbindung starten. Wenn eine neue Clientverbindung den Standby-WS-Manager auswählt, der neu aktiv ist, wird der Client erneut mit demselben Warteschlangenmanager verbunden.

Wenn sich der Client während der Verbindungswiederverbindung in der Mitte eines MQI-Aufrufs befindet, muss er eine erweiterte Wartezeit tolerieren, bevor der Aufruf abgeschlossen wird.

Wenn der Fehler während einer Stapelübertragung auf einem Nachrichtenkanal stattfindet, wird die Stapelverarbeitung rückgängig gemacht und erneut gestartet.

Das Umschalten ist schneller als ein fehlerhaftes Umschalten und dauert nur so lange, wie eine Instanz des Warteschlangenmanagers gestoppt und eine andere Instanz gestartet wird. Bei einem WS-Manager mit nur wenigen Protokollsätzen, die wiedergegeben werden sollen, kann bei bester Umschaltung die Reihenfolge einiger Sekunden in Anspruch nehmen. Um zu schätzen, wie lange die Funktionsübernahme dauert, müssen Sie die Zeit hinzufügen, die für das Erkennen des Fehlers benötigt wird. Am besten ist die Erkennung in der Größenordnung von 10 Sekunden, und kann mehrere Minuten, je nach Netzwerk und Dateisystem.

### Multi *Anwendungswiederherstellung*

Die Anwendungswiederherstellung ist die automatisierte Fortsetzung der Anwendungsverarbeitung nach dem Failover. Die Anwendungswiederherstellung nach dem Failover erfordert sorgfältige Konstruktion. Einige Anwendungen müssen ein Failover-Failover durchgeführt haben.

Das Ziel der Anwendungswiederherstellung ist es, dass die Anwendung die Verarbeitung mit nur einer kurzen Verzögerung fortsetzt. Bevor Sie mit der neuen Verarbeitung fortfahren, muss die Anwendung die Arbeitseinheit, die sie während des Fehlers verarbeitet hat, wieder zurücksenden und erneut übergeben.

Ein Problem für die Anwendungswiederherstellung besteht darin, dass der gemeinsame Kontext zwischen dem IBM MQ MQI client und dem Warteschlangenmanager, der im Warteschlangenmanager gespeichert ist, verloren geht. Der IBM MQ MQI client stellt den größten Teil des Kontexts wieder her, es gibt jedoch einige Teile des Kontexts, die nicht zuverlässig wiederhergestellt werden können. In den folgenden Abschnitten werden einige Merkmale der Anwendungswiederherstellung beschrieben und die Auswirkungen auf die Wiederherstellung von Anwendungen, die mit einem Warteschlangenmanager mit mehreren Instanzen verbunden sind, beeinflusst.

## Transaktionsorientiertes Messaging

Aus der Perspektive der Zustellung von Nachrichten ändert das Failover die persistenten Eigenschaften des IBM MQ-Messaging nicht. Wenn Nachrichten persistent sind und ordnungsgemäß in Arbeitseinheiten verwaltet werden, gehen die Nachrichten während einer Funktionsübernahme nicht verloren.

Aus der Perspektive der Transaktionsverarbeitung werden Transaktionen nach einem Failover entweder zurückgesetzt oder festgeschrieben.

Nicht festgeschriebene Transaktionen werden rückgängig gemacht. Nach der Funktionsübernahme erhält eine erneut zuschaltbare Anwendung einen `MQRC_BACKED_OUT` -Ursachencode, um anzuzeigen, dass die Transaktion fehlgeschlagen ist. Anschließend muss die Transaktion erneut gestartet werden.

Festgeschriebene Transaktionen sind Transaktionen, die die zweite Phase einer zweiphasigen Festschreibung erreicht haben, oder einphasige Transaktionen (nur Nachrichten), die begonnen haben `MQCMIT`.

Wenn der Warteschlangenmanager der Transaktionskoordinator ist und `MQCMIT` die zweite Phase der zweiphasigen Festschreibung vor dem Fehler begonnen hat, wird die Transaktion erfolgreich abgeschlossen. Die Fertigstellung wird unter der Steuerung des Warteschlangenmanagers ausgeführt und wird fortgesetzt, wenn der Warteschlangenmanager erneut ausgeführt wird. In einer wiederverbindbaren Anwendung wird der Aufruf `MQCMIT` normal beendet.

In einer einphasigenFestschreibung, die nur Nachrichten umfasst, wird eine Transaktion, die die COMMIT-Verarbeitung gestartet hat, normalerweise unter der Steuerung des Warteschlangenmanagers abge-

schlossen, sobald sie erneut ausgeführt wird. In einer wiederverbindbaren Anwendung wird MQCMIT normal beendet.

Wiederanschlussfähige Clients können unter der Steuerung des Warteschlangenmanagers als Transaktionskoordinator einzelne Phasentransaktionen verwenden. Der erweiterte transaktionsorientierte Client unterstützt keine erneute Verbindung. Wenn eine erneute Verbindung angefordert wird, wenn der transaktionsorientierte Client eine Verbindung herstellt, ist die Verbindung erfolgreich, aber ohne dass die Verbindung erneut hergestellt werden kann. Die Verbindung verhält sich so, als ob sie nicht wieder angeschlossen werden kann.

## Anwendungsneustart oder -wiederaufnehmen

Failover unterbricht eine Anwendung. Nach einem Fehler kann eine Anwendung von Anfang an erneut gestartet werden, oder sie kann die Verarbeitung nach der Unterbrechung wieder aufnehmen. Letzteres wird als *automatische Clientwiederverbindung* bezeichnet. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt.

Bei einer IBM MQ MQI client-Anwendung können Sie eine Verbindungsoption festlegen, die für die automatische Verbindungswiederherstellung des Clients gelten soll. Die Optionen sind MQCNO_RECONNECT oder MQCNO_RECONNECT_Q_MGR . Wenn keine Option festgelegt ist, versucht der Client nicht, die Verbindung automatisch wieder herzustellen, und der Fehler des Warteschlangenmanagers MQRC_CONNECTION_BROKEN wird an den Client zurückgegeben. Sie können den Client so gestalten, dass er versucht, eine neue Verbindung zu starten, indem Sie einen neuen Aufruf MQCONN oder MQCONNX absetzen.

Serverprogramme müssen erneut gestartet werden. Sie können nicht automatisch durch den Warteschlangenmanager an dem Punkt, an dem sie verarbeitet wurden, wieder verbunden werden, wenn der Warteschlangenmanager oder der Server fehlgeschlagen ist. IBM MQ-Serverprogramme werden in der Standby-Warteschlangenmanagerinstanz in der Regel nicht erneut gestartet, wenn eine Instanz des Multi-Instanz-Warteschlangenmanagers fehlschlägt.

Sie können ein IBM MQ-Serverprogramm auf zwei Arten so automatisieren, dass es auf dem Standby-Server erneut startet:

1. Packen Sie Ihre Serveranwendung als WS-Manager-Service. Sie wird erneut gestartet, wenn der Standby-WS-Manager erneut gestartet wird.
2. Schreiben Sie Ihre eigene Failoverlogik, die beispielsweise ausgelöst wird, wenn die Failover-Protokollnachricht von einer Standby-WS-Manager-Instanz geschrieben wird, wenn sie gestartet wird. Die Anwendungsinstanz muss dann MQCONN oder MQCONNX aufrufen, nachdem sie gestartet wurde, um eine Verbindung zum Warteschlangenmanager herzustellen.

## Failover wird ermittelt

Einige Anwendungen müssen sich der Funktionsübernahme bewusst sein, andere nicht. Betrachten Sie diese beiden Beispiele.

1. Eine Messaging-Anwendung, die Nachrichten über einen Nachrichtenübertragungskanal empfängt oder empfängt, erfordert normalerweise nicht, dass der Warteschlangenmanager am anderen Ende des Kanals aktiv ist: Es ist unwahrscheinlich, dass es betroffen ist, wenn der Warteschlangenmanager am anderen Ende des Kanals auf einer Standby-Instanz erneut gestartet wird.
2. Eine IBM MQ MQI client-Anwendung verarbeitet den persistenten Nachrichteneingang aus einer Warteschlange und reiht persistente Nachrichtenantworten im Rahmen einer einzelnen Arbeitseinheit in eine andere Warteschlange ein: Wenn sie einen MQRC_BACKED_OUT-Ursachencode von MQPUT, MQGET oder MQCMIT innerhalb eines Synchronisationspunkts verarbeitet, indem sie die Arbeitseinheit erneut startet, gehen keine Nachrichten verloren. Darüber hinaus muss die Anwendung keine spezielle Verarbeitung ausführen, um einen Verbindungsfehler zu bewältigen.

Nehmen Sie jedoch im zweiten Beispiel an, dass die Anwendung die Warteschlange durchsucht, um die zu verarbeitende Nachricht mit der Option MQGET , MQGMO_MSG_UNDER_CURSOR, auszuwählen. Die Verbindungswiederholung setzt den Anzeigecursor zurück und der Aufruf MQGET gibt nicht die richtige Nachricht zurück. In diesem Beispiel muss die Anwendung eine Übernahme durch eine Funktionsüber-

nahme (Failover) durchgeführt werden. Außerdem muss die Anwendung vor der Ausgabe einer weiteren MQGET für die Nachricht unter dem Cursor den Anzeigecursor wiederherstellen.

Das Sperren des Anzeigecursors ist ein Beispiel dafür, wie der Anwendungskontext nach der Verbindungswiederverbindung geändert wird. Andere Fälle werden im Abschnitt „Wiederherstellung eines automatisch verbundenen Clients" auf Seite 603 beschrieben.

Sie verfügen über drei alternative Entwurfsmuster für IBM MQ MQI client-Anwendungen nach dem Failover. Nur bei einem von ihnen muss das Failover nicht erkannt werden.

**Keine Verbindungswiederverbindung**

In diesem Muster stoppt die Anwendung die gesamte Verarbeitung für die aktuelle Verbindung, wenn die Verbindung unterbrochen ist. Damit die Anwendung die Verarbeitung fortsetzen kann, muss sie eine neue Verbindung zum WS-Manager herstellen. Die Anwendung ist vollständig für die Übertragung aller erforderlichen Statusinformationen verantwortlich, um die Verarbeitung für die neue Verbindung fortzusetzen. Vorhandene Clientanwendungen, die nach dem Verlust ihrer Verbindung die Verbindung zu einem WS-Manager herstellen, werden auf diese Weise geschrieben.

Der Client empfängt einen Ursachencode, z. B. MQRC_CONNECTION_BROKEN, oder MQRC_Q_MGR_NOT_AVAILABLE vom nächsten MQI-Aufruf, nachdem die Verbindung unterbrochen wurde. Die Anwendung muss alle zugehörigen IBM MQ-Statusinformationen, wie z. B. Warteschlangenkennungen, löschen und einen neuen Aufruf MQCONN oder MQCONNX ausgeben, um eine neue Verbindung herzustellen, und anschließend die IBM MQ-Objekte erneut öffnen, die sie verarbeiten muss.

Das standardmäßige MQI-Verhalten ist, dass die WS-Manager-Verbindungskennung nicht mehr verwendbar wird, nachdem eine Verbindung zum Warteschlangenmanager verloren gegangen ist. Der Standardwert entspricht der Einstellung der Option MQCNO_RECONNECT_DISABLED auf MQCONNX , um die Wiederherstellung der Anwendungsverbindung nach einem Failover zu verhindern.

**Failover-tolerant**

Schreiben Sie die Anwendung so, dass sie von einem Failover nicht betroffen ist. Manchmal ist eine sorgfältige Fehlerbehandlung ausreichend, um das Failover zu bewältigen.

**Verbindungskonnektion**

Registrieren Sie einen Ereignishandler MQCBT_EVENT_HANDLER mit dem Warteschlangenmanager. Der Ereignishandler wird mit MQRC_RECONNECTING bereitgestellt, wenn der Client versucht, die Verbindung zum Server erneut herzustellen, und MQRC_RECONNECTED nach einer erfolgreichen Verbindungswiederaufschaltung. Anschließend können Sie eine Routine ausführen, um einen vorhersehbaren Status wieder herzustellen, so dass die Clientanwendung die Verarbeitung fortsetzen kann.

## Wiederherstellung eines automatisch verbundenen Clients

Der Failover ist ein unerwartetes Ereignis, und für einen automatisch verbundenen Client, der die Auswirkungen der Verbindungswiederaufschaltung hat, müssen die Auswirkungen vorhersehbar sein.

Ein Hauptelement, bei dem ein unerwarteter Fehler in eine vorhersehbare und zuverlässige Wiederherstellung verwandelt wird, ist die Verwendung von Transaktionen.

Im vorherigen Abschnitt wurde ein Beispiel, „2" auf Seite 602, von einem IBM MQ MQI client mit einer lokalen Transaktion zur Koordination von MQGET und MQPUT vorgestellt. Der Client gibt einen MQCMIT -oder MQBACK -Aufruf als Antwort auf einen Fehler MQRC_BACKED_OUT aus und übergibt die zurückgesetzte Transaktion erneut. Der Warteschlangenmanager-Fehler führt dazu, dass die Transaktion zurückgesetzt wird, und das Verhalten der Clientanwendung stellt sicher, dass keine Transaktionen ausgeführt werden und keine Nachrichten verloren gehen.

Beachten Sie, dass es für einen Rückruf erforderlich sein kann, die konsumierende Anwendung wiederaufzunehmen, wenn der Status des Rückrufkonsumentenparameters MQCS_SUSPENDED_USER_ACTION lautet.

Nicht der gesamte Programmstatus wird als Teil einer Transaktion verwaltet, und daher werden die Auswirkungen der Neuverbindung schwerer zu verstehen. Sie müssen wissen, wie die Verbindungswie-

derherstellung den Status eines IBM MQ MQI client ändert, um Ihre Clientanwendung so zu gestalten, dass die Failover-Funktion des Warteschlangenmanagers überstanden wird.

Sie können Ihre Anwendung ohne speziellen Failover-Code entwerfen, da die Verbindungsfehler mit der gleichen Logik wie andere Fehler behandelt werden. Alternativ können Sie feststellen, dass für die erneute Verbindung eine spezielle Fehlerbehandlung erforderlich ist, und in IBM MQ einen Ereignishandler registrieren, um eine Routine zur Funktionsübernahme auszuführen. Die Routine kann die Verarbeitung der Verbindungswiederherstellung selbst verarbeiten oder eine Markierung setzen, um dem Hauptprogrammthread anzuzeigen, dass die Verarbeitung wiederaufgenommen werden muss, wenn die Verarbeitung wieder aufgenommen werden muss.

Die IBM MQ MQI client-Umgebung erkennt den Failover selbst und stellt nach der Verbindungswiederherstellung so viel Kontext wie möglich wieder her, indem sie einige Statusinformationen im Client speichert und zusätzliche MQI-Aufrufe im Namen der Clientanwendung ausgibt, um ihren IBM MQ-Status wiederherzustellen. Zum Beispiel werden die Kennungen für Objekte, die am Point of Failure geöffnet waren, wiederhergestellt, und temporäre dynamische Warteschlangen werden mit demselben Namen geöffnet. Es gibt jedoch Änderungen, die unvermeidlich sind, und Sie brauchen Ihr Design, um mit diesen Änderungen umzugehen. Die Änderungen lassen sich in fünf Arten kategorisieren:

1. Neue oder zuvor nicht diagnostizierte Fehler werden von MQI-Aufrufen zurückgegeben, bis ein konsistenter neuer Kontextstatus durch das Anwendungsprogramm wiederhergestellt wird.

   Ein Beispiel für den Empfang eines neuen Fehlers ist der Rückkehrcode MQRC_CONTEXT_NOT_AVAILABLE , wenn versucht wird, Kontext zu übergeben, nachdem der Kontext vor der erneuten Verbindung gespeichert wurde. Der Kontext kann nach der Neuverbindung nicht wiederhergestellt werden, da der Sicherheitskontext nicht an ein nicht berechtigtes Clientprogramm übergeben wird. Zu diesem Zweck würde ein zerstörerliches Anwendungsprogramm den Sicherheitskontext abrufen.

   In der Regel bearbeiten Anwendungen häufig auftretende und vorhersehbare Fehler sorgfältig und führen unübliche Fehler zu einem generischen Fehlerbehandlungsprogramm zurück. Das Fehlerbehandlungsprogramm kann die Verbindung zu IBM MQ trennen und die Verbindung erneut herstellen oder sogar das Programm vollständig stoppen. Um die Kontinuität zu verbessern, müssen Sie möglicherweise einige Fehler auf eine andere Art und Weise behandeln.
2. Nicht persistente Nachrichten gehen möglicherweise verloren.
3. Transaktionen werden zurückgesetzt (dies kann auch asynchrone Konsumenten aussetzen, siehe vorherigen Text).
4. MQGET -oder MQPUT -Aufrufe, die außerhalb eines Synchronisationspunkts verwendet werden, werden möglicherweise mit dem Verlust einer Nachricht unterbrochen.
5. Timing-induzierter Fehler, aufgrund eines längeren Wartestatus in einem MQI-Aufruf.

Einige Details zum verlorenen Kontext werden im folgenden Abschnitt aufgelistet.

- Nicht persistente Nachrichten werden gelöscht, es sei denn, sie werden mit der Option NPMCLASS（HIGH） in eine Warteschlange gestellt, und der Fehler des Warteschlangenmanagers hat die Option zum Speichern nicht persistenter Nachrichten beim Herunterfahren nicht unterbrochen.
- Eine nicht permanente Subskription geht verloren, wenn eine Verbindung unterbrochen wird. Bei der Neuverbindung wird sie erneut aufgebaut. Ziehen Sie die Verwendung einer permanenten Subskription in Betracht.
- Das Intervall für den Abwartestatus wird neu berechnet. Wenn der Grenzwert überschritten wird, wird MQRC_NO_MSG_AVAILABLE zurückgegeben. Analog wird die Subskriptionsverfallszeit so neu berechnet, dass sie die gleiche Gesamtverfallszeit erhält.
- Die Position des Suchcursors in einer Warteschlange ist verloren; sie wird in der Regel vor der ersten Nachricht neu erstellt.
  - MQGET -Aufrufe, die MQGMO_BROWSE_MSG_UNDER_CURSOR oder MQGMO_MSG_UNDER_CURSORangeben, schlagen mit dem Ursachencode MQRC_NO_MSG_AVAILABLEfehl.
  - Nachrichten, die zum Durchsuchen gesperrt sind, werden entsperr
  - Durchsuchbare Nachrichten mit dem Geltungsbereich der Handle sind nicht markiert und können erneut durchsucht werden.

  - In den meisten Fällen sind die markierten Nachrichten nicht markiert.
- Der Sicherheitskontext ist verloren. Der Versuch, den gespeicherten Nachrichtenkontext zu verwenden, wie z. B. das Einreihen einer Nachricht mit MQPMO_PASS_ALL_CONTEXT , schlägt mit MQRC_CONTEXT_NOT_AVAILABLE fehl.
- Nachrichtentoken gehen verloren. MQGET gibt mit einem Nachrichtentoken den Ursachencode MQRC_NO_MSG_AVAILABLEzurück.

  **Anmerkung:** *MsgId* und *CorrelId*, da sie Teil der Nachricht sind, werden während der Funktionsübernahme mit der Nachricht beibehalten, sodass die MQGET Verwendung von MsgId oder CorrelId wie erwartet funktioniert.
- Nachrichten, die in eine Warteschlange unter Synchronisationspunkt in einer nicht festgeschriebenen Transaktion gestellt werden, sind nicht mehr verfügbar.
- Die Verarbeitung von Nachrichten in einer logischen Reihenfolge oder in einer Nachrichtengruppe führt zu einem Rückkehrcode von MQRC_RECONNECT_INCOMPATIBLE nach der Verbindungswiederverbindung.
- Ein MQI-Aufruf gibt möglicherweise MQRC_RECONNECT_FAILED zurück und nicht die allgemeinere MQRC_CONNECTION_BROKEN , die Clients normalerweise heute empfangen.
- Die Wiederherstellung während eines MQPUT-Aufrufs außerhalb des Synchronisationspunkts gibt MQRC_CALL_INTERRUPTED zurück, wenn der IBM MQ MQI client nicht weiß, ob die Nachricht dem Warteschlagenmanager erfolgreich zugestellt wurde. Die Verbindungswiederherstellung während MQCMIT verhält sich ähnlich.
- MQRC_CALL_INTERRUPTED wird nach erfolgreicher Verbindungswiederherstellung zurückgegeben, wenn der IBM MQ MQI client vom Warteschlangenmanager keine Antwort über Erfolg oder Fehlschlagen folgender Aktionen erhalten hat:
  - die Zustellung einer persistenten Nachricht unter Verwendung eines MQPUT-Aufrufs außerhalb des Synchronisationspunkts.
  - die Zustellung einer persistenten Nachricht oder einer Nachricht mit Standardpersistenz unter Verwendung eines MQPUT1-Aufrufs außerhalb des Synchronisationspunkts.
  - Festschreiben einer Transaktion mit Hilfe eines MQCMIT-Aufrufs. Die Antwort wird nur nach einer erfolgreichen Verbindungswiederverbindung zurückgegeben.
- Kanäle werden als neue Instanzen erneut gestartet (sie können auch unterschiedliche Kanäle sein), sodass kein Kanalexitstatus beibehalten wird.
- Temporäre dynamische Warteschlangen werden als Teil des Prozesses zur Wiederherstellung wiederverbindbarer Clients, die temporäre dynamische Warteschlangen geöffnet haben, wiederhergestellt. Es werden keine Nachrichten in einer temporären dynamischen Warteschlange zurückgespeichert, aber Anwendungen, bei denen die Warteschlange geöffnet war oder sich an den Namen der Warteschlange erinnert haben, können die Verarbeitung fortsetzen.

  Es besteht die Möglichkeit, dass, wenn die Warteschlange von einer anderen Anwendung als der erstellt wird, die sie erstellt hat, sie möglicherweise nicht schnell genug zurückgeschrieben wird, wenn sie als nächste Referenz angegeben wird. Wenn ein Client beispielsweise eine temporäre dynamische Warteschlange als Empfangswarteschlange für Antworten erstellt und eine Antwortnachricht von einem Kanal in die Warteschlange gestellt werden soll, wird die Warteschlange möglicherweise nicht in der Zeit wiederhergestellt. In diesem Fall würde der Kanal normalerweise die Antwort-Nachricht in die Warteschlange für nicht zustellbare Nachrichten stellen.

  Wenn eine wiederverbindungsfähige Clientanwendung eine temporäre dynamische Warteschlange anhand des Namens öffnet (da eine andere Anwendung sie bereits erstellt hat), ist der IBM MQ MQI client bei der Verbindungswiederherstellung nicht in der Lage, die temporäre dynamische Warteschlange erneut zu erstellen, da er nicht über das Modell verfügt, nach dem sie erstellt werden kann. In der MQI kann nur eine Anwendung die temporäre dynamische Warteschlange nach Modell öffnen. Andere Anwendungen, die die temporäre dynamische Warteschlange verwenden möchten, müssen MQPUT1-oder -Serverbindungen verwenden oder die erneute Verbindung erneut versuchen, wenn sie fehlschlägt.

Es können nur nicht persistente Nachrichten in eine temporäre dynamische Warteschlange gestellt werden, und diese Nachrichten gehen während der Funktionsübernahme verloren. Dieser Verlust gilt für Nachrichten, die während der erneuten Verbindung mit MQPUT1 in eine temporäre dynamische Warteschlange gestellt werden. Wenn das Failover während des MQPUT1-Befehls auftritt, wird die Nachricht möglicherweise nicht gestellt, obwohl die Nachricht MQPUT1 erfolgreich ist. Eine Fehlerumgehung für dieses Problem besteht darin, permanente dynamische Warteschlangen zu verwenden. Jede Serverbindungsanwendung kann die temporäre dynamische Warteschlange nach Namen öffnen, weil sie nicht wieder verbunden werden kann.

### Multi *Datenwiederherstellung und hohe Verfügbarkeit*

Hochverfügbarkeitslösungen unter Verwendung von Warteschlangenmanagern mit mehreren Instanzen müssen einen Mechanismus zum Wiederherstellen von Daten nach einem Speicherfehler enthalten.

Ein Warteschlangenmanager mit mehreren Instanzen erhöht die Verfügbarkeit von WS-Managerprozessen, aber nicht die Verfügbarkeit anderer Komponenten, wie z. B. das Dateisystem, das der Warteschlangenmanager zum Speichern von Nachrichten verwendet, und andere Informationen.

Eine Möglichkeit, Daten hoch verfügbar zu machen, ist die Verwendung von vernetztem, ausfallsicheren Datenspeicher. Sie können entweder Ihre eigene Lösung mit Hilfe eines vernetzten Dateisystems und einer flexiblen Datenspeicherung erstellen, oder Sie können eine integrierte Lösung kaufen. Wenn Sie die Ausfallsicherheit mit Disaster Recovery kombinieren möchten, ist die asynchrone Plattenreplikation verfügbar, die eine Plattenreplikation über mehrere Dutzend oder Hunderte von Kilometern zulässt.

Sie können die Art und Weise konfigurieren, in der verschiedene IBM MQ-Verzeichnisse Speichermedien zugeordnet werden, um die beste Verwendung der Datenträger zu ermöglichen. Für *Multi-Instanz*-Warteschlangenmanager gibt es eine wichtige Unterscheidung zwischen zwei Typen von IBM MQ-Verzeichnissen und -Dateien.

**Verzeichnisse, die gemeinsam von den Instanzen eines Warteschlangenmanagers gemeinsam genutzt werden müssen.**

Die Informationen, die zwischen verschiedenen Instanzen eines Warteschlangenmanagers gemeinsam genutzt werden müssen, befinden sich in zwei Verzeichnissen: den Verzeichnissen `qmgrs` und `logs`. Die Verzeichnisse müssen sich in einem gemeinsam genutzten Netzdateisystem befinden. Es wird empfohlen, einen Speicherdatenträger zu verwenden, der eine ständige hohe Verfügbarkeit und eine ausgezeichnete Leistung bietet, da die Daten ständig geändert werden, da Nachrichten erstellt und gelöscht werden.

**Verzeichnisse und Dateien, die nicht *haben* , um von Instanzen eines Warteschlangenmanagers gemeinsam genutzt zu werden.**

Einige andere Verzeichnisse müssen nicht von verschiedenen Instanzen eines Warteschlangenmanagers gemeinsam genutzt werden und werden schnell mit anderen Verzeichnissen zurückgeschrieben, als mit einem gespiegelten Dateisystem.

- Ausführbare IBM MQ-Dateien und das Verzeichnis 'tools'. Durch die erneute Installation oder durch Sichern und Zurückschreiben aus einem gesicherten Dateiarchiv ersetzen.
- Konfigurationsinformationen, die für die Installation als Ganzes geändert werden. Die Konfigurationsinformationen werden entweder von IBM MQ verwaltet, wie z. B. die `mqs.ini`-Datei auf AIX, Linux, and Windows-Systemen oder Teil Ihrer eigenen Konfigurationsmanagement, wie z. B. **MQSC**-Konfigurationsscripts. Sichern und Zurückschreiben mit einem Dateiarchiv.
- Installationsweite Ausgabe, wie z. B. Traces, Fehlerprotokolle und FFDC-Dateien. Die Dateien werden in den Unterverzeichnissen `errors` und `trace` im Standarddatenverzeichnis gespeichert. Das Standarddatenverzeichnis auf AIX and Linux-Systemen ist `/var/mqm`. Unter Windows ist das Standarddatenverzeichnis das IBM MQ-Installationsverzeichnis.

Sie können auch einen Sicherungswarteschlangenmanager verwenden, um regelmäßige Datenträgersicherungen eines Multi-Instanz-Warteschlangenmanagers mit linearer Protokollierung zu verwenden. Ein Sicherungswarteschlangenmanager stellt keine Wiederherstellung bereit, die so schnell wie von einem gespiegelten Dateisystem ist, und die Änderungen seit der letzten Sicherung werden nicht wiederhergestellt. Der Sicherungswarteschlangenmanager-Mechanismus eignet sich besser für die Verwendung in

Off-Site-Szenarios zur Wiederherstellung nach einem Katastrophenfall als die Wiederherstellung eines Warteschlangenmanagers nach einem lokalisierten Speicherfehler.

## IBM MQ-Verfügbarkeitslösungen kombinieren

Von Anwendungen werden weitere IBM MQ-Leistungsmerkmale genutzt, um die Verfügbarkeit zu steigern. Warteschlangenmanager mit mehreren Instanzen ergänzen andere Hochverfügbarkeitsfunktionen.

### IBM MQ-Cluster steigern die Warteschlangenverfügbarkeit

Sie können die Warteschlangenverfügbarkeit erhöhen, indem Sie mehrere Definitionen einer Clusterwarteschlange erstellen; bis zu einer jeden Warteschlange auf den einzelnen Managern im Cluster.

Angenommen, ein Member des Clusters schlägt fehl, und anschließend wird eine neue Nachricht an eine Clusterwarteschlange gesendet. Wenn die Nachricht *has* nicht in den fehlgeschlagenen Warteschlangenmanager wechseln soll, wird die Nachricht an einen anderen aktiven WS-Manager im Cluster gesendet, der über eine Definition der Warteschlange verfügt.

Obwohl Cluster die Verfügbarkeit erheblich erhöhen, gibt es zwei zusammengehörige Fehlerszenarios, die zu verzögerten Nachrichten führen. Durch die Erstellung eines Clusters mit Multi-Instanz-WS-Managern wird die Wahrscheinlichkeit, dass eine Nachricht verzögert wird, verringert.

**Marooned-Nachrichten**
: Wenn ein Warteschlangenmanager im Cluster fehlschlägt, werden keine weiteren Nachrichten, die an andere WS-Manager im Cluster weitergeleitet werden können, an den fehlgeschlagenen Warteschlangenmanager weitergeleitet. Nachrichten, die bereits gesendet wurden, werden bis zum Neustart des fehlgeschlagenen Warteschlangenmanagers gemieckt.

**Affinitäten**

: Affinität ist der Begriff, der verwendet wird, um Informationen zu beschreiben, die zwischen zwei ansonsten getrennten Berechnungen gemeinsam genutzt werden. Beispielsweise besteht eine Affinität zwischen einer Anwendung, die eine Anforderungsnachricht an einen Server sendet, und der gleichen Anwendung, die die Verarbeitung der Antwort erwartet. Ein weiteres Beispiel wäre eine Folge von Nachrichten, die die Verarbeitung jeder Nachricht in Abhängigkeit von den vorherigen Nachrichten enthält.

: Wenn Sie Nachrichten an geclusterte Warteschlangen senden, müssen Sie Affinitäten berücksichtigen. Müssen Sie aufeinanderfolgende Nachrichten an denselben WS-Manager senden, oder kann jede Nachricht an ein beliebtenes Member des Clusters gesendet werden?

: Wenn Sie Nachrichten an denselben Warteschlangenmanager im Cluster senden müssen und die Nachrichten fehlschlagen, warten die Nachrichten in der Übertragungswarteschlange des Senders, bis der fehlgeschlagene Cluster-WS-Manager erneut ausgeführt wird.

Wenn der Cluster mit Multi-Instanz-WS-Managern konfiguriert ist, ist die Verzögerung, die auf den Neustart des fehlgeschlagenen Warteschlangenmanagers wartet, auf die Reihenfolge einer Minute begrenzt, während die Bereitschaftsdatenbank die Zeit übernimmt. Wenn die Bereitschaftsdatenbank aktiv ist, werden die Nachrichten wieder aufgenommen, die Kanäle zur neu aktivierten Warteschlangenmanagerinstanz gestartet werden und die Nachrichten, die in Übertragungswarteschlangen anstanden, werden gestartet.

Eine Möglichkeit, einen Cluster so zu konfigurieren, dass die Nachrichten, die von einem fehlgeschlagenen Warteschlangenmanager verzögert werden, überwunden werden, besteht darin, zwei verschiedene Warteschlangenmanager auf jedem Server im Cluster zu implementieren und eine aktive und eine als die Standby-Instanz der verschiedenen Warteschlangenmanager zu definieren. Hierbei handelt es sich um eine Aktiv-Standby-Konfiguration, die die Verfügbarkeit des Clusters erhöht.

Neben den Vorteilen einer reduzierten Verwaltung und einer erhöhten Skalierbarkeit stellen Cluster weiterhin zusätzliche Elemente der Verfügbarkeit zur Verfügung, um Multi-Instanz-WS-Manager zu ergänzen. Cluster schützen vor anderen Typen von Fehlern, die sowohl die aktiven als auch die Standby-Instanzen eines Warteschlangenmanagers betreffen.

**Ununterbrochener Service**
Ein Cluster stellt einen ununterbrochenen Service bereit. Neue Nachrichten, die vom Cluster empfangen werden, werden an aktive Warteschlangenmanager gesendet, die verarbeitet werden sollen. Verlassen Sie sich nicht auf einen Warteschlangenmanager mit mehreren Instanzen, um einen unterbrechungsfreien Service bereitzustellen, da es Zeit für den Standby-WS-Manager benötigt, um den Fehler zu erkennen und seinen Start abzuschließen, damit die Kanäle erneut verbunden werden, und für fehlgeschlagene Nachrichtenstapel, die erneut übergeben werden sollen.

**Lokalisierter Ausfall**
Es gibt praktische Einschränkungen, wie weit die aktiven, die Standby-und die Dateisystemserver voneinander entfernt werden können, da sie mit einer Millisekundengeschwindigkeit interagieren müssen, um eine akzeptable Leistung zu erzielen.

Clusterwarteschlangenmanager erfordern Interaktionsgeschwindigkeiten in der Größenordnung von vielen Sekunden und können geographisch überall auf der Welt verteilt werden.

**Betriebsfehler**
Durch die Verwendung von zwei verschiedenen Mechanismen zur Erhöhung der Verfügbarkeit reduzieren Sie die Wahrscheinlichkeit, dass ein Betriebsfehler, wie z. B. ein menschlicher Fehler, Ihre Verfügbarkeitsanstrengungen beeinträchtigt.

## Gruppen mit gemeinsamer Warteschlange erhöhen die Verfügbarkeit der Nachrichtenverarbeitung

z/OS Gruppen mit gemeinsamer Warteschlange, eine nur unter z/OS verfügbare Option, ermöglichen es einer Gruppe von Warteschlangenmanagern, die Wartung einer Warteschlange gemeinsam zu übernehmen. Wenn ein Warteschlangenmanager ausfällt, verarbeiten die anderen WS-Manager weiterhin alle Nachrichten in der Warteschlange. Warteschlangenmanager mit mehreren Instanzen werden in z/OS nicht unterstützt und ergänzen Gruppen mit gemeinsamer Warteschlange nur als Teil einer umfassenderen Messaging-Architektur.

## IBM MQ-Clients erhöhen die Anwendungsverfügbarkeit

IBM MQ MQI client-Programme können je nach Verfügbarkeit der Warteschlangenmanager, Verbindungsgewichtungen und Affinitäten eine Verbindung zu verschiedenen Warteschlangenmanagern in einer Warteschlangenmanagergruppe herstellen. Wenn Sie eine Anwendung auf einem anderen System als dem Warteschlangenmanager ausführen, auf dem der Warteschlangenmanager ausgeführt wird, können Sie die Gesamtverfügbarkeit einer Lösung verbessern, solange eine Möglichkeit besteht, die Anwendung erneut zu verbinden, wenn die WS-Manager-Instanz, mit der sie verbunden ist, fehlgeschlagen ist.

WS-Manager-Gruppen werden verwendet, um die Clientverfügbarkeit zu erhöhen, indem ein Client aus einem Warteschlangenmanager entfernt wird, der gestoppt wurde, und die Lastverteilung von Clientverbindungen in eine Gruppe von Warteschlangenmanagern, und zwar wie ein IP-Sprayer. Die Clientanwendung darf keine Affinitäten mit dem fehlgeschlagenen Warteschlangenmanager haben, z. B. eine Abhängigkeit in einer bestimmten Warteschlange oder die Verarbeitung kann nicht fortgesetzt werden.

Automatische Clientwiederverbindungs-und Multi-Instanz-Warteschlangenmanager erhöhen die Verfügbarkeit der Clients, indem einige Affinitätsprobleme behoben werden. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt.

Sie können die Option `MQCNO  MQCNO_RECONNECT_Q_MGR` festlegen, um einen Client zu zwingen, die Verbindung zum selben Warteschlangenmanager erneut herzustellen:

1. Wenn der zuvor verbundene einzelne Instanz-WS-Manager nicht aktiv ist, wird die Verbindung wiederholt, bis der Warteschlangenmanager wieder aktiv ist.
2. Wenn der Warteschlangenmanager als Multi-Instanz-Warteschlangenmanager konfiguriert ist, stellt der Client die Verbindung zu der Instanz wieder her, die aktiv ist.

Durch die automatische Verbindung zu demselben Warteschlangenmanager werden viele der Statusinformationen, die der Warteschlangenmanager im Namen des Clients gespeichert hat, wie z. B. die von ihm geöffneten Warteschlangen und das von ihm subskribierte Topic, wiederhergestellt. Wenn der Client eine

dynamische Empfangswarteschlange für Antworten geöffnet hat, um eine Antwort auf eine Anforderung zu empfangen, wird auch die Verbindung zur Warteschlange für Antwortantworten wiederhergestellt.

## Linux MQ Adv. RDQM-Hochverfügbarkeit

RDQM (Warteschlangenmanager für replizierte Daten) ist eine Hochverfügbarkeitslösung, die auf Red Hat Enterprise Linux for x86-64 -Plattformen verfügbar ist.

Eine RDQM-Konfiguration besteht aus drei Servern, die, jeder mit einer Instanz des Warteschlangenmanagers, in einer Hochverfügbarkeitsgruppe (HA-Gruppe) konfiguriert werden. Eine Instanz ist der aktive Warteschlangenmanager, der seine Daten synchron zu den anderen beiden Instanzen repliziert. Fällt der Server mit dem aktiven Warteschlangenmanager aus, wird eine andere Instanz des Warteschlangenmanagers gestartet, die über die aktuellen Betriebsdaten verfügt. Die drei Instanzen des Warteschlangenmanagers können optional eine variable IP-Adresse gemeinsam nutzen, sodass Clients nur mit einer einzigen IP-Adresse konfiguriert werden müssen. Es kann jeweils nur eine Instanz des Warteschlangenmanagers ausgeführt werden, selbst wenn die HA-Gruppe aufgrund von Netzproblemen partitioniert wird. Der Server, auf dem der Warteschlangenmanager ausgeführt wird, wird als 'primärer Server' bezeichnet. Jeder der beiden anderen Server wird als 'sekundärer' Server bezeichnet.

Es werden drei Knoten verwendet, um die Möglichkeit einer Split-Brain-Situation stark zu reduzieren. In einem Hochverfügbarkeitsystem mit zwei Knoten kann das Hochverfügbarkeitssystem auftreten, wenn die Verbindung zwischen den beiden Knoten unterbrochen ist. Wenn keine Verbindung besteht, können beide Knoten gleichzeitig den Warteschlangenmanager ausführen und dabei verschiedene Daten akkumulieren. Wenn die Verbindung wiederhergestellt wird, gibt es zwei verschiedene Versionen der Daten (ein 'split-brain'), und es ist ein manueller Eingriff erforderlich, um zu entscheiden, welche Datei beibehalten werden soll und welche Daten gelöscht werden sollen.

RDQM verwendet ein Drei-Knoten-System mit Quorum, um die Split-Brain-Situation zu vermeiden. Knoten, die mit mindestens einem der anderen Knoten kommunizieren können, bilden ein Quorum. WS-Manager können nur auf einem Knoten ausgeführt werden, der das Quorum hat. Der WS-Manager kann nicht auf einem Knoten ausgeführt werden, der nicht mit mindestens einem anderen Knoten verbunden ist. Daher kann die Ausführung auf zwei Knoten nicht gleichzeitig erfolgen:

- Wenn ein einzelner Knoten ausfällt, kann der WS-Manager auf einem der beiden anderen Knoten ausgeführt werden. Wenn zwei Knoten fehlschlagen, kann der Warteschlangenmanager nicht auf dem verbleibenden Knoten ausgeführt werden, da der Knoten nicht über das Quorum verfügt (der verbleibende Knoten kann nicht feststellen, ob die beiden anderen Knoten ausgefallen sind oder noch aktiv sind und die Konnektivität verloren gegangen ist).
- Wenn ein einzelner Knoten die Konnektivität verliert, kann der WS-Manager auf diesem Knoten nicht ausgeführt werden, da der Knoten nicht über das Quorum verfügt. Der WS-Manager kann auf einem der verbleibenden zwei Knoten ausgeführt werden, die das Quorum haben. Wenn alle Knoten die Verbindung verlieren, kann der Warteschlangenmanager auf keinem der Knoten ausgeführt werden, da keiner der Knoten das Quorum hat.

**Anmerkung:** Die IBM MQ Console unterstützt keine Warteschlangenmanager mit replizierten Daten. Sie können den IBM MQ Explorer mit Warteschlangenmanagern für replizierte Daten verwenden, aber dadurch werden keine Informationen angezeigt, die spezifisch für die RDQM-Funktionen sind.

Die Gruppenkonfiguration der drei Knoten wird von Pacemaker gehandhabt. Die Replikation zwischen den drei Knoten wird von DRBD verarbeitet. (Unter https://clusterlabs.org/pacemaker/ finden Sie Informationen zu Pacemaker und unter https://docs.linbit.com/docs/users-guide-9.0/ finden Sie Informationen zu DRBD.)

Sie können Ihre Warteschlangenmanager für replizierte Daten mit dem im Abschnitt „WS-Manager-Daten sichern” auf Seite 724 beschriebenen Prozess sichern. Das Stoppen und Sichern des Warteschlangenmanagers hat keine Auswirkung auf die Knotenüberwachung, die von der RDQM-Konfiguration ausgeführt wird.

Die folgende Abbildung zeigt eine typische Implementierung mit einem RDQM, der auf jedem der drei Knoten in der HA-Gruppe ausgeführt wird.

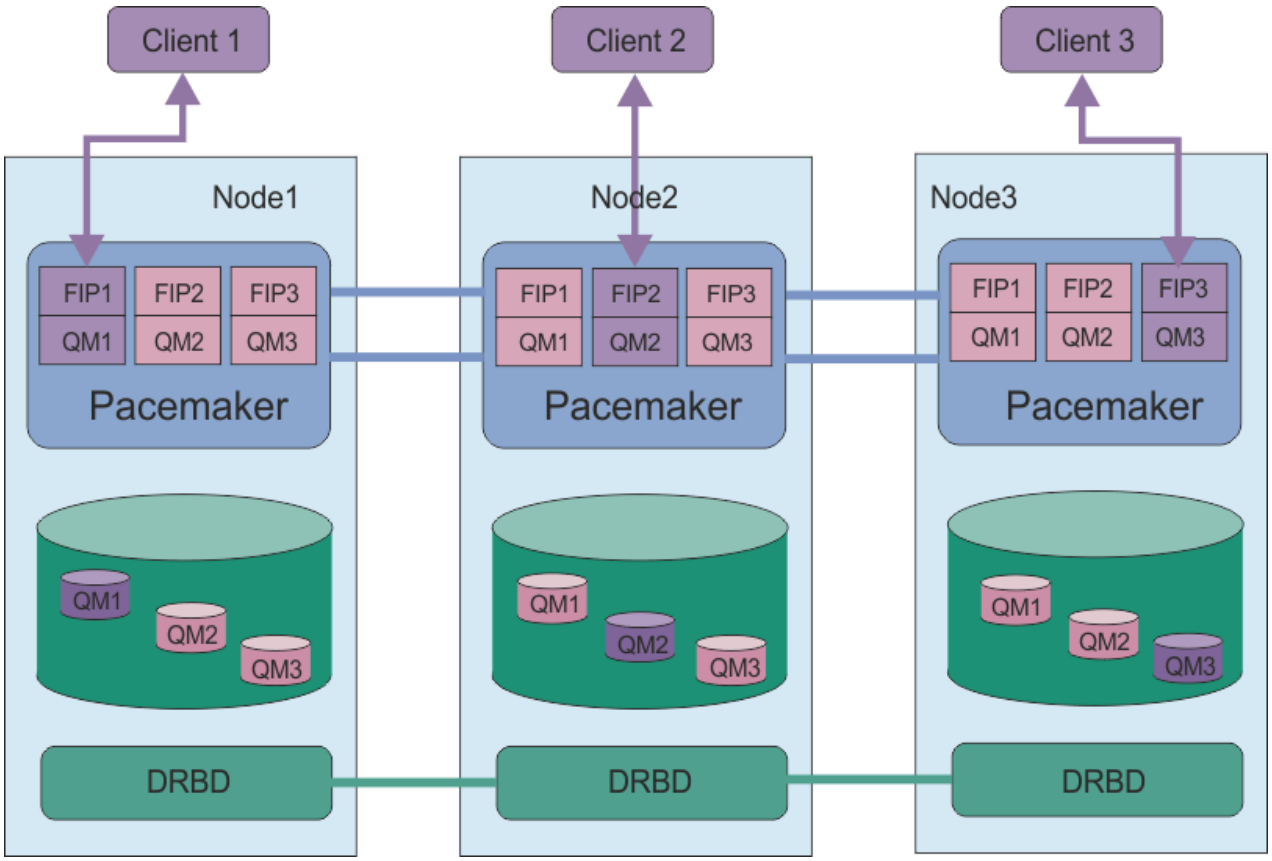



*Abbildung 78. Beispiel für eine HA-Gruppe mit drei RDQMs*

In der nächsten Abbildung ist Node3 fehlgeschlagen, die Pacemaker-Links sind verloren gegangen, und der Warteschlangenmanager QM3 wird stattdessen auf Knoten2 ausgeführt.

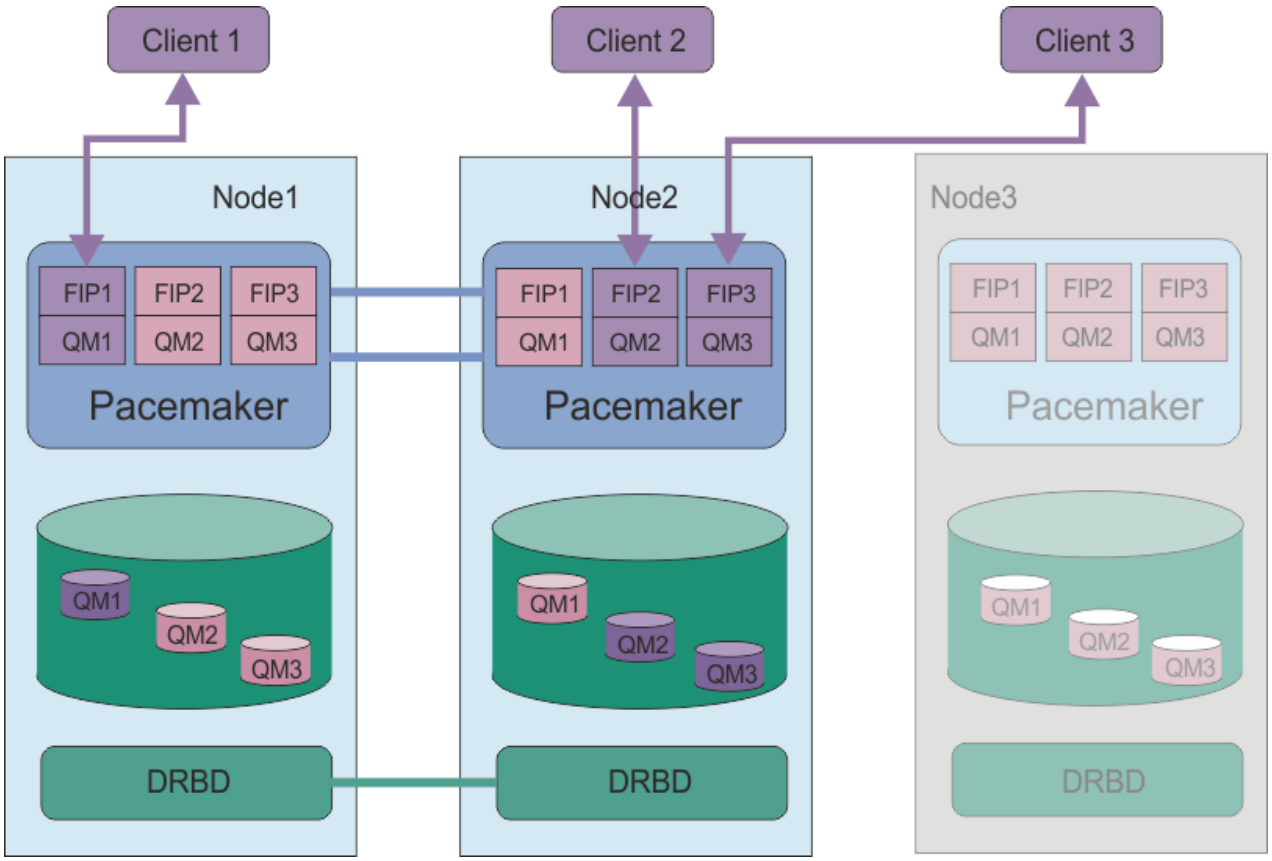



*Abbildung 79. Beispiel für Fehlschlagen von node3*

**Anmerkung:** Wenn die Warteschlangenmanager von einem anderen Knoten übernommen werden, behalten Sie den Status bei, den Sie vor der Funktionsübernahme hatten. Warteschlangenmanager, die ausgeführt wurden, werden gestartet, und Warteschlangenmanager, die gestoppt waren, bleiben gestoppt.

**Zugehörige Tasks**

RDQM (replizierte Datenwarteschlangenmanager) installieren

Aktualisierungen der Wartungsstufe für RDQM anwenden

Warteschlangenmanager mit replizierten Daten migrieren

Fehlerbehebung für RDQM-Konfigurationen

## Linux *Voraussetzungen für RDQM HA-Lösung*

Sie müssen eine Reihe von Anforderungen erfüllen, bevor Sie die RDQM-Gruppe mit hoher Verfügbarkeit konfigurieren können.

### Systemvoraussetzungen

Bevor Sie die RDQM-HA-Gruppe konfigurieren, müssen Sie auf jedem der drei Server, die Teil der HA-Gruppe sein sollen, eine Konfiguration abschließen.

- Für jeden Knoten ist eine Datenträgergruppe mit dem Namen `drbdpool` erforderlich. Der Speicher für jeden replizierten Datenwarteschlangenmanager wird als separater logischer Datenträger pro WS-Manager aus dieser Datenträgergruppe zugeordnet. Um die beste Leistung zu erhalten, sollte diese Datenträgergruppe aus einem oder mehreren physischen Datenträgern bestehen, die internen Plattenlaufwerken entsprechen (vorzugsweise SSDs). Sie können `drbdpool` vor oder nach der Installation der RDQM-HA-Lösung erstellen, aber Sie müssen `drbdpool` erstellen, bevor Sie tatsächlich RDQMs

erstellen. Überprüfen Sie Ihre Datenträgergruppenkonfiguration mit dem Befehl **vgs**. Die Ausgabe sollte etwa folgendermaßen aussehen:

```
  VG       #PV #LV #SN Attr   VSize   VFree
  drbdpool   1   9   0 wz--n- <16.00g <7.00g
  rhel       1   2   0 wz--n- <15.00g     0
```

Überprüfen Sie insbesondere, ob das Zeichen c in der sechsten Spalte der Attribute (d. h. `wz--nc`) vorhanden ist. c gibt an, dass das Clustering aktiviert ist. Falls dies der Fall ist, müssen Sie die Datenträgergruppe löschen und ohne Clustering erneut erstellen.

- Nachdem Sie die `drbdpool`-Datenträgergruppe erstellt haben, können Sie keine weiteren Aktionen ausführen. IBM MQ verwaltet die logischen Datenträger, die in `drbdpool` erstellt wurden, und wie und wo sie bereitgestellt werden.
- Für jeden Knoten sind bis zu drei Schnittstellen erforderlich, die für die Konfiguration der RDQM-Unterstützung verwendet werden:
  - Eine primäre Schnittstelle für den Pacemaker, um die HA-Gruppe zu überwachen.
  - Eine alternative Schnittstelle für den Pacemaker, um die HA-Gruppe zu überwachen.
  - Eine Schnittstelle für die synchrone Datenreplikation, die als Replikationsschnittstelle bezeichnet wird. Dies sollte eine ausreichende Bandbreite haben, um die Replikationsanforderungen bei der erwarteten Auslastung aller replizierten Datenwarteschlangenmanager, die in der HA-Gruppe ausgeführt werden, zu unterstützen.

  Sie können die HA-Gruppe so konfigurieren, dass für alle drei Schnittstellen dieselbe IP-Adresse verwendet wird. Für jede Schnittstelle wird eine separate IP-Adresse verwendet, oder die IP-Adresse wird für primäre und alternative IP-Adresse und eine separate IP-Adresse für die Replikationsschnittstelle verwendet.

  Für die maximale Fehlertoleranz sollten diese Schnittstellen unabhängige Netzschnittstellenkarten (NICs) sein.
- DRBD erfordert, dass jeder Knoten in der HA-Gruppe einen gültigen Internet-Host-Namen hat (der von `uname -n` zurückgegebene Wert), wie in RFC 952, geändert durch RFC 1123, definiert.
- Wenn zwischen den Knoten in der HA-Gruppe eine Firewall vorhanden ist, muss die Firewall den Datenverkehr zwischen den Knoten in einem Portbereich zulassen. Es wird ein Beispielscript bereitgestellt, `/opt/mqm/samp/rdqm/firewalld/configure.sh`, das die erforderlichen Ports öffnet, wenn Sie die Standardfirewall unter RHEL ausführen. Sie müssen das Script als `root` ausführen. Wenn Sie eine andere Firewall verwenden, überprüfen Sie die Servicedefinitionen `/usr/lib/firewalld/services/rdqm*` , um festzustellen, welche Ports geöffnet werden müssen. Das Script fügt die folgenden permanenten firewallD-Serviceregeln für DRBD, Pacemaker und IBM MQ hinzu:
  - MQ-INSTALLATIONSPFAD/samp/rdqm/firewalld/services/rdqm-drbd.xml lässt TCP-Ports 7000-7100 zu.
  - MQ-INSTALLATIONSPFAD/samp/rdqm/firewalld/services/rdqm-pacemaker.xml lässt UDP-Ports 5404-5407 zu.
  - MQ-INSTALLATIONSPFAD/samp/rdqm/firewalld/services/rdqm-mq.xml lässt TCP-Port 1414 zu (Sie müssen das Script bearbeiten, wenn Sie einen anderen Port verwenden)
- Wenn das System SELinux im restriktiven Modus verwendet, müssen Sie möglicherweise folgenden Befehl ausführen:

```
semanage permissive -a drbd_t
```

V 9.3.0.1 V 9.3.2 Wenn Sie das Paket `drbd-selinux` installiert haben, müssen Sie **semanage**nicht ausführen. Sie müssen dieses Paket entweder auf jedem Knoten installieren oder **semanage** auf jedem Knoten ausführen.

## Netzvoraussetzungen

Es wird empfohlen, die drei Knoten in der RDQM-HA-Gruppe in demselben Rechenzentrum zu lokalisieren.

Wenn Sie die Knoten in verschiedenen Rechenzentren suchen, beachten Sie die folgenden Einschränkungen:

- Die Leistung verschlechtert sich schnell mit zunehmender Latenzzeit zwischen Rechenzentren. Obwohl IBM eine Latenzzeit von bis zu 5 ms unterstützt, stellen Sie möglicherweise fest, dass Ihre Anwendungsleistung nicht mehr als 1 bis 2 ms Latenzzeit tolerieren kann.
- Die über die Replikationsverbindung gesendeten Daten unterliegen keiner zusätzlichen Verschlüsselung, die über die evtl. durch die Verwendung von IBM MQ AMS vorgegebene Verschlüsselung hinausgeht.

Sie können optional eine variable IP-Adresse konfigurieren, damit ein Client dieselbe IP-Adresse für einen replizierten Datenwarteschlangenmanager (RDQM) verwenden kann, unabhängig davon, auf welchem Knoten in der HA-Gruppe er ausgeführt wird. Die Gleitadresse wird an eine benannte physische Schnittstelle auf dem Primärknoten für den RDQM gebunden. Wenn RDQM nicht ausgeführt wird und ein anderer Knoten zum primären Knoten wird, wird die variable IP-Adresse an eine Schnittstelle mit demselben Namen auf dem neuen primären Server gebunden. Die physischen Schnittstellen auf den drei Knoten müssen alle denselben Namen haben und gehören zum selben Teilnetz wie die variable IP-Adresse.

## Benutzervoraussetzungen für die Konfiguration des Clusters

Sie können die RDQM-HA-Gruppe als Benutzer `root` konfigurieren. Wenn Sie nicht als `root` konfigurieren möchten, konfigurieren Sie stattdessen als Benutzer in der Gruppe `mqm`. Damit ein Benutzer in der Gruppe `mqm` den RDQM-Cluster konfigurieren kann, muss Folgendes erfüllt sein:

- Der Benutzer `mqm` muss sudo verwenden können, um Befehle auf jedem der drei Server auszuführen, aus denen die RDQM-HA-Gruppe besteht.
- Wenn der `mqm`-Benutzer SSH ohne Kennwort verwenden kann, um Befehle auf jedem der drei Server auszuführen, aus denen die RDQM-HA-Gruppe besteht, muss der Benutzer Befehle nur auf einem der Server ausführen.
- Der Benutzer `mqm` muss auf allen drei Servern dieselbe UID haben.
- Die Gruppe `mqm` muss auf allen drei Servern dieselbe GID haben.

Sie müssen sudo so konfigurieren, dass der Benutzer `mqm` die folgenden Befehle mit Rootberechtigung ausführen kann:

```
/opt/mqm/bin/crtmqm
/opt/mqm/bin/dltmqm
/opt/mqm/bin/rdqmadm
/opt/mqm/bin/rdqmstatus
```

## Benutzeranforderungen für die Arbeit mit Warteschlangenmanagern

Zum Erstellen, Löschen oder Konfigurieren von Warteschlangenmanagern für replizierte Daten (RDQMs) müssen Sie eine Benutzer-ID verwenden, die zu den Gruppen `mqm` und `haclient` gehört (die Gruppe `haclient` wird während der Pacemaker-Installation erstellt).

Linux *passwordless SSH konfigurieren*

Sie können passwordless SSH konfigurieren, so dass Sie nur Befehle zum Ausgeben von Befehlen auf einem Knoten in der HA-Gruppe benötigen. (Die Einrichtung von kennwortlosem SSH ist optional. Alternativ können Sie Befehle manuell auf jeden Knoten kopieren.)

## Informationen zu diesem Vorgang

Wenn Sie passwordless SSH konfigurieren möchten, müssen Sie die `mqm` -ID auf jedem Knoten konfigurieren und anschließend einen Schlüssel für jeden Knoten für diesen Benutzer generieren. Anschließend

verteilen Sie die Schlüssel an die anderen Knoten, und testen Sie die Verbindung, um die einzelnen Knoten zur Liste der bekannten Hosts hinzuzufügen. Schließlich sperren Sie die ID `mqm` .

**Anmerkung:** In den Anweisungen wird davon ausgegangen, dass Sie eine HA-Gruppe mit separaten primären, alternativen und Replikationsschnittstellen definieren, und Sie daher passwordless SSH-Zugriff auf die primäre und die alternative Schnittstelle definieren. Wenn Sie planen, ein System mit einer einzigen IP-Adresse zu konfigurieren, definieren Sie passwordless SSH-Zugriff über diese einzelne Schnittstelle.

RDQM erfordert, dass der Befehl `ssh` ohne Interaktion funktioniert, d. h. ohne Aufforderung zur Eingabe eines Kennworts usw.

## Vorgehensweise

1. Führen Sie auf jedem der drei Knoten die folgenden Schritte aus, um den `mqm` -Benutzer zu konfigurieren und einen SSH-Schlüssel zu generieren:
   a) Ändern Sie das `mqm` -Ausgangsverzeichnis in `/home/mqm`:

   ```
   usermod -d /home/mqm mqm
   ```

   b) Erstellen Sie das Verzeichnis `/home/mqm`:

   ```
   mkhomedir_helper mqm
   ```

   c) Fügen Sie das `mqm` -Kennwort hinzu:

   ```
   passwd mqm
   ```

   d) Führen Sie die interaktive Shell wie folgt aus: `mqm`:

   ```
   su mqm
   ```

   e) Generieren Sie den Authentifizierungsschlüssel für `mqm` :

   ```
   ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
   ```

2. Führen Sie auf jedem der drei Knoten die folgenden Schritte aus, um den Schlüssel des Knotens zu den anderen beiden Knoten hinzuzufügen, und testen Sie die Verbindungen für die einzelnen Primärknoten und (falls verwendet) alternativen Adressen:
   a) Fügen Sie den Schlüssel zu den fernen Knoten hinzu.

   ```
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address
   ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
   ```

   b) Überprüfen Sie passwordless ssh und aktualisieren Sie known_hosts für ferne Knoten:

   ```
   ssh remote_node1_primary_address uname -n
   ssh remote_node1_alternate_address uname -n
   ssh remote_node2_primary_address uname -n
   ssh remote_node2_alternate_address uname -n
   ```

   Für jede Verbindung werden Sie aufgefordert, zu bestätigen, dass Sie fortfahren möchten. Bestätigen Sie für jede Aktualisierung, um die known_hosts zu aktualisieren. Sie müssen dies ausführen, bevor Sie versuchen, die HA-Gruppe mit passwordless SSH zu konfigurieren.

   c) Beenden Sie die interaktive Shell wie folgt: `mqm`:

   ```
   exit
   ```

3. Führen Sie auf jedem Knoten als Root die folgenden Schritte aus, um das `mqm` -Kennwort zu entfernen und die ID zu sperren:
   a) Entfernen Sie das Kennwort für `mqm` :

```
passwd -d mqm
```

b) Sperren `mqm`:

```
passwd -l mqm
```

4. Führen Sie auf jedem Knoten als Root die folgenden Schritte aus, um den Sudo-Zugriff für den Benutzer `mqm` einzurichten:

a) Bearbeiten Sie die Datei `sudoers` mit dem Befehl **`visudo`**:

```
visudo
```

b) Suchen Sie die Zeile "*## Allows people in group wheel to run all commands*" und fügen Sie unterhalb dieser Zeile den folgenden Text hinzu:

```
#%mqm    ALL=(ALL)       ALL
```

c) Suchen Sie die Zeile "*## Same thing without a password*" und fügen Sie unterhalb dieser Zeile den folgenden Text hinzu:

```
%mqm            ALL=(ALL)       NOPASSWD: ALL
```

## Linux *Definieren des Pacemaker-Clusters (HA-Gruppe)*

Die HA-Gruppe ist ein Pacemaker-Cluster. Sie definieren den Pacemaker -Cluster, indem Sie die Datei `/var/mqm/rdqm.ini` bearbeiten und den Befehl **`rdqmadm`** ausführen.

### Informationen zu diesem Vorgang

Weitere Informationen zu Pacemaker finden Sie unter https://clusterlabs.org/pacemaker/. Sie können den Pacemaker-Cluster als Benutzer in der `mqm`-Gruppe erstellen, wenn der Benutzer `mqm` `sudo` verwenden kann. Wenn der Benutzer auch ohne Kennwort SSH zu jedem Server verwenden kann, müssen Sie nur die Datei `rdqm.ini` bearbeiten und **`rdqmadm`** auf einem der Server ausführen, um den Pacemaker -Cluster zu erstellen. Andernfalls müssen Sie die Datei erstellen und den Befehl als `root` auf jedem der Server ausführen, die Knoten sein sollen.

Die Datei `rdqm.ini` enthält die IP-Adressen, die RDQM für die Knoten im Pacemaker -Cluster verwendet. Bei Installationen von RHEL 8 und RHEL 9 müssen Sie den Namen jedes Knotens angeben. Dies muss der vom Befehl **`uname -n`** zurückgegebene Hostname sein. Bei RHEL 7-Installationen ist die Angabe des Knotennamens optional.

Eine RDQM-HA-Gruppe kann so konfiguriert werden, dass sie eine, zwei oder drei IP-Adressen verwendet:

- Eine IP-Adresse: Heartbeats und Replikation verwenden denselben Link
- Zwei IP-Adressen: Heartbeats und Replikation verwenden separate Links
- Drei IP-Adressen: ein Link für die Replikation und zwei separate Links für Heartbeats

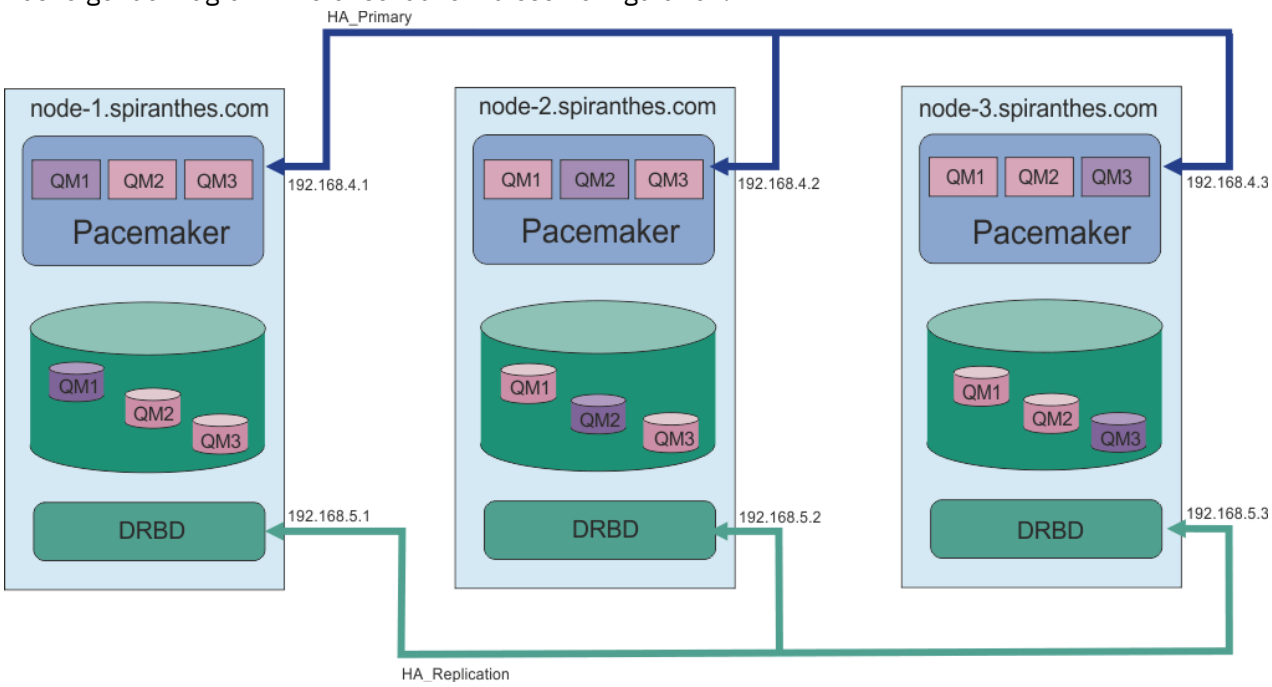
Diese Optionen werden bereitgestellt, um verschiedene Implementierungsmuster für RDQM zu unterstützen. Die verschiedenen Optionen können verwendet werden, um die Ausfallsicherheit der RDQM-Lösung basierend auf der verwendeten Umgebung zu maximieren. Die Konfigurationen, die entweder zwei oder drei IP-Adressen verwenden, sind in erster Linie für Implementierungen gedacht, bei denen eine differenzierte Steuerung erforderlich ist, über die das physische Netz die Überwachungssignale und den Replikationsdatenverkehr verbindet, um Redundanz für die Konnektivität zwischen Knoten zu konfigurieren. Alternativ kann die hoch verfügbare und ausfallsichere Konnektivität auf der Netzebene implementiert werden, beispielsweise durch die Verwendung von Link-Aggregation. Bei der Verbindungszusammenlegung werden mehrere physische Netzverbindungen verwendet, um eine einzelne logische Verbindung bereitzustellen, die weiterhin funktionieren kann, wenn einzelne physische Verbindungen fehlschlagen. Wenn RDQM in einer Umgebung bereitgestellt wird, in der die Netzkonnektivität virtualisiert ist und/oder

in der die ausfallsichere Konnektivität auf der Netzebene implementiert ist, ist die Verwendung einer einzigen IP-Adresse für Überwachungssignale und Replikation in der Regel vorzuziehen.

Das folgende Beispiel veranschaulicht die Verwendung zweier IP-Adressen. Ihre Datei `rdqm.ini` enthält ein Feld `HA_Primary` und ein Feld `HA_Replication` für jeden Knoten, aber kein Feld `HA_Alternate` :

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

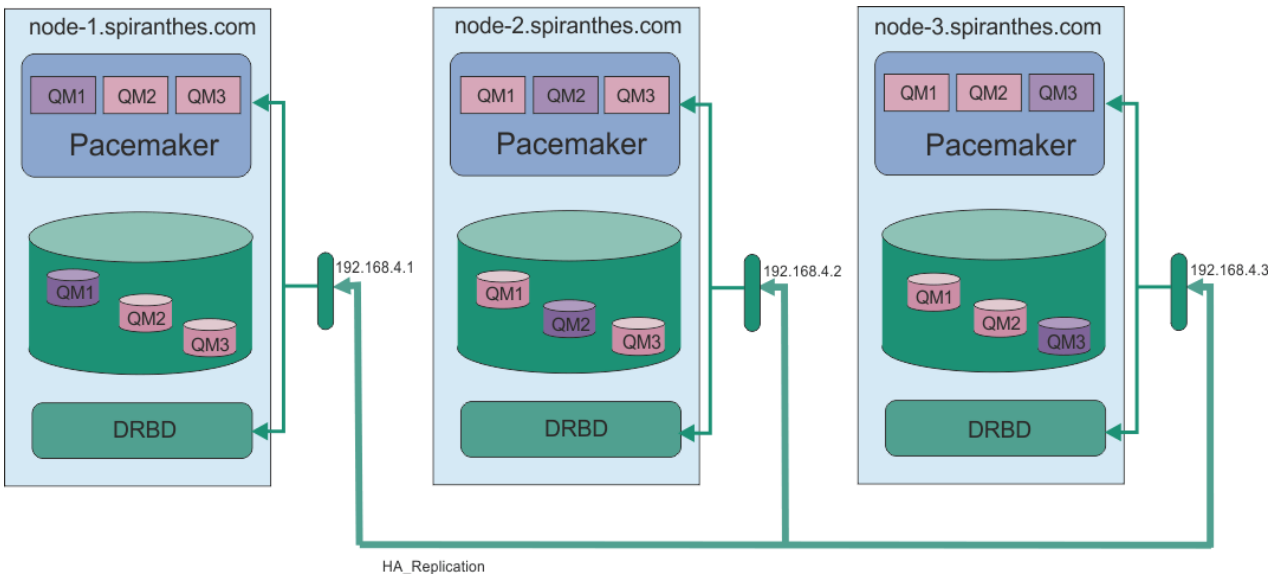
Das folgende Diagramm veranschaulicht diese Konfiguration:




Die folgende Beispieldatei zeigt die Konfiguration für einen Pacemaker -Beispielcluster, der die Schnittstelle `HA_Replication` für die Überwachung verwendet (dies könnte beispielsweise für eine Konzeptnachweisbereitstellung verwendet werden). In diesem Fall geben Sie nur die Schnittstelle `HA_Replication` an:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Replication=192.168.4.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Replication=192.168.4.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Replication=192.168.4.3
```

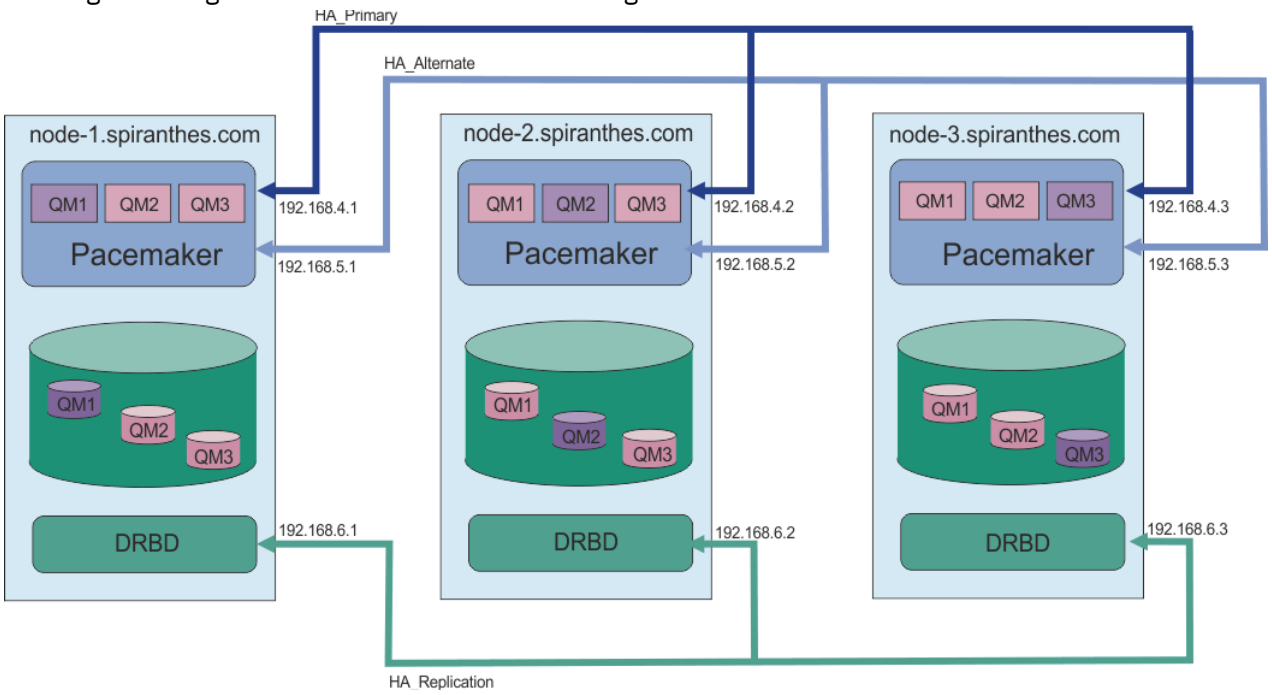
Das folgende Diagramm veranschaulicht diese Konfiguration:




Die folgende Beispieldatei zeigt die Konfiguration für einen Beispiel-Pacemaker-Cluster an, der für jede Schnittstelle eine separate IP-Adresse verwendet:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3
```

Das folgende Diagramm veranschaulicht diese Konfiguration:



Die Reihenfolge, in der Sie die Knoten angeben, muss in allen `rdqm.ini`-Dateien in Ihrer Konfiguration identisch sein. Für die drei Knoten muss es eine einheitliche Ansicht geben (welcher ist Knoten1, welcher Knoten2 usw.).

## Prozedur

- Gehen Sie wie folgt vor, um den Pacemaker-Cluster als Benutzer `root` zu
  a) Bearbeiten Sie die Datei `/var/mqm/rdqm.ini` auf einem der drei Server, so dass die Datei den Cluster definiert.
  b) Kopieren Sie die Datei auf die anderen beiden Server, die Knoten im Pacemaker-Cluster sein werden.
  c) Führen Sie den folgenden Befehl als `root` auf jedem der drei Server aus:

  ```
  rdqmadm -c
  ```

- Gehen Sie folgendermaßen vor, um das Pacemaker-Cluster als Benutzer in der Gruppe `mqm` auf jedem Knoten zu definieren:
  a) Stellen Sie sicher, dass der Benutzer `mqm` für die Ausführung von Befehlen **sudo** verwenden kann.
  b) Bearbeiten Sie die Datei `/var/mqm/rdqm.ini` auf einem der drei Server, so dass die Datei den Pacemaker-Cluster definiert.
  c) Kopieren Sie die `/var/mqm/rdqm.ini` auf die beiden anderen Server, die Knoten im Pacemaker-Cluster sein werden.
  d) Führen Sie den folgenden Befehl auf jedem Server aus:

  ```
  rdqmadm -c
  ```

- Gehen Sie folgendermaßen vor, um das Pacemaker-Cluster als Benutzer in der Gruppe `mqm` aus einem Knoten zu definieren:
  a) Stellen Sie sicher, dass der Benutzer `mqm` **sudo** zum Ausführen von Befehlen verwenden kann und über SSH ohne Kennwort eine Verbindung zu jedem Server herstellen kann.
  b) Bearbeiten Sie die Datei `/var/mqm/rdqm.ini` auf einem der drei Server, so dass die Datei den Pacemaker-Cluster definiert.
  c) Führen Sie den folgenden Befehl aus:

  ```
  rdqmadm -c
  ```

**Zugehörige Verweise**
rdqmadm (Verwaltung replizierter Daten-WS-Manager-Cluster)

Linux *Pacemaker-Cluster löschen (HA-Gruppe)*
Die HA-Gruppe ist ein Pacemaker-Cluster. Sie können eine Pacemaker -Clusterkonfiguration löschen, indem Sie den Befehl **rdqmadm** mit der Option `-u` ausführen.

## Informationen zu diesem Vorgang

Sie können die Pacemaker-Clusterkonfiguration nicht löschen, wenn noch replizierte Datenwarteschlangenmanager auf einem der Knoten vorhanden sind.

## Prozedur

- Um die Konfiguration des Pacemaker-Clusters zu löschen, geben Sie den folgenden Befehl in einem der Knoten ein:

  ```
  rdqmadm -u
  ```

**Zugehörige Verweise**

rdqmadm (Verwaltung replizierter Daten-WS-Manager-Cluster)

## Linux *HA-RDQM erstellen*

Sie verwenden den Befehl **crtmqm** , um einen replizierten Datenwarteschlangenmanager (RDQM) mit hoher Verfügbarkeit zu erstellen.

## Informationen zu diesem Vorgang

Sie können einen replizierten Datenwarteschlangenmanager (RDQM) mit hoher Verfügbarkeit als Benutzer in der Gruppe `mqm` erstellen, wenn der Benutzer 'mqm' sudo verwenden kann. Wenn der Benutzer auch ohne Kennwort SSH für jeden Knoten verwenden kann, müssen Sie nur den Befehl RDQM erstellen auf einem Knoten ausführen, um RDQM auf allen drei Knoten zu erstellen. Andernfalls müssen Sie `root` sein, um einen RDQM zu erstellen, und Sie müssen Befehle auf allen drei Knoten ausführen.

**Anmerkung:** Es gibt einen absoluten Grenzwert von 129 Warteschlangenmanagern in einer HA-Gruppe. Wenn Sie versuchen, mehr zu erstellen, schlägt der Versuch fehl. In der Praxis können beim Hinzufügen von mehr als 50 Warteschlangenmanagern zu einer HA-Gruppe Zeitlimitprobleme auftreten.

Die folgenden Punkte enthalten Anleitungen zur Dimensionierung des Dateisystems des Warteschlangenmanagers:

1. Wenn Sie einen RDQM-Warteschlangenmanager erstellen, wird ein Dateisystem zum Speichern von Warteschlangenmanagerdaten und -protokollen zugeordnet. Es ist wichtig, die Größe dieses Dateisystems entsprechend zu ändern, damit der Warteschlangenmanager fortlaufende Aktivitäten in seinen Protokollen aufzeichnen und Anwendungsnachrichten in Warteschlangen speichern kann. Berücksichtigen Sie bei der Dimensionierung des Dateisystems die Anforderungen für Spitzennachrichten, das künftige Workloadwachstum und Anwendungsausfälle, die dazu führen können, dass Nachrichten in Warteschlangen erstellt werden. Informationen zur Berechnung der Größe des Wiederherstellungsprotokolls des Warteschlangenmanagers finden Sie unter „Wie groß sollte ich mein Protokolldateisystem machen?" auf Seite 703. Bei der Berechnung des Speicherbedarfs für Anwendungsnachrichten müssen die Größe und Anzahl der Nachrichten sowie deren MQMD-Header und Nachrichteneigenschaften berücksichtigt werden.
2. Die Größe von Dateisystemen des RDQM-Warteschlangenmanagers kann nicht dynamisch geändert werden. Wenn dies erforderlich ist, müssen Sie einen RDQM-Warteschlangenmanager mit einem größeren Dateisystem sichern und wiederherstellen (siehe „Größe des Dateisystems für einen HA-RDQM-Warteschlangenmanager ändern" auf Seite 625).
3. Sie können die Größe einzelner Warteschlangen auf Platte begrenzen, indem Sie lokale Warteschlangenattribute wie MAXDEPTH und MAXFSIZE verwenden. Siehe Warteschlangendateien von IBM MQ ändern.
4. Sie sollten Ihre laufende Plattenbelegung überwachen und entsprechend reagieren, wenn die Plattenbelegung zunimmt, bevor die Dateisystembelegung kritisch wird. Die Dateisystemnutzung kann mithilfe von Plattform-/Betriebssystemfunktionen überwacht werden oder indem Metriken abonniert werden, die in IBM MQ -Systemthemen veröffentlicht werden, die unter In den Systemthemen veröffentlichte Metrikenbeschrieben sind.

## Prozedur

- Gehen Sie wie folgt vor, um einen RDQM als Benutzer in der `mqm` -Gruppe zu erstellen:
  a) Stellen Sie sicher, dass der `mqm` -Benutzer **sudo** verwenden kann, um Befehle auszuführen, und eine Verbindung zu jedem Server über SSH ohne Kennwort herstellen kann.
  b) Geben Sie den folgenden Befehl ein:

```
crtmqm -sx [-fs FilesystemSize] qmname
```

Hierbei steht *qmname* für den Namen des replizierten Datenwarteschlangenmanagers. Sie können optional die Dateisystemgröße für den Warteschlangenmanager angeben (d. a. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird).

Der Befehl versucht, SSH zu verwenden, um eine Verbindung zu den anderen Knoten im Cluster herzustellen, die der `mqm` -Benutzer ist. Wenn die Verbindung erfolgreich ist, werden die sekundären Instanzen des Warteschlangenmanagers auf den Knoten erstellt. Andernfalls müssen Sie die sekundären Instanzen erstellen und anschließend den Befehl **`crtmqm -sx`** ausführen (wie für Benutzer `root`beschrieben).

- Gehen Sie wie folgt vor, um einen RDQM als Benutzer `root` zu erstellen
  a) Geben Sie den folgenden Befehl auf jedem der Knoten ein, die sekundäre Instanzen von RDQM enthalten sollen:

  ```
  crtmqm -sxs [-fs FilesystemSize] qmname
  ```

  Hierbei steht *qmname* für den Namen des replizierten Datenwarteschlangenmanagers. Sie können optional die Dateisystemgröße für den Warteschlangenmanager angeben (d. a. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird). Sie müssen die gleiche Dateisystemgröße für RDQM auf allen drei Knoten in der HA-Gruppe angeben. Die Größe ist ein numerischer Wert, der in GB angegeben wird. Sie können einen Wert in MB angeben, indem Sie den Wert gefolgt vom Zeichen `M` eingeben.

  Der Befehl erstellt eine sekundäre Instanz von RDQM.

  b) Geben Sie auf dem verbleibenden Knoten den folgenden Befehl ein:

  ```
  crtmqm -sx [-fs FilesystemSize] qmname
  ```

  Hierbei steht *qmname* für den Namen des replizierten Datenwarteschlangenmanagers. Sie können optional die Dateisystemgröße für den Warteschlangenmanager angeben. Die Größe ist ein numerischer Wert, der in GB angegeben wird. Sie können einen Wert in MB angeben, indem Sie den Wert gefolgt vom Zeichen `M`eingeben.

  Der Befehl bestimmt, ob die sekundäre Instanz des Warteschlangenmanagers auf den anderen beiden Knoten vorhanden ist. Sind Sekundärdateien vorhanden, erstellt und startet der Befehl den primären WS-Manager. Wenn die Sekundärdateien nicht vorhanden sind, werden Sie angewiesen, den Befehl **`crtmqm -sxs`** auf jedem der Knoten auszuführen.

  Neben den Argumenten DataPath (**`-md`**) und LogPath (**`-ld`**) sind alle Argumente, die für die Erstellung eines Linux -Standardwarteschlangenmanagers gültig sind, auch für einen primären Warteschlangenmanager für replizierte Daten gültig.

  **Anmerkung:** Beim Erstellen eines RDQM wird die nächste freie Portnummer über 7000 der Replikationsverbindung zugeordnet. Wenn festgestellt wird, dass der ausgewählte Port von einer anderen Anwendung verwendet wird, schlägt der Befehl **`crtmqm`** mit der Fehlermeldung AMQ6543 fehl und dieser Port wird einer Ausschlussliste hinzugefügt. Sie müssen die sekundären Instanzen des Warteschlangenmanagers löschen und den Befehl **`crtmqm`** anschließend erneut ausführen.

**Zugehörige Verweise**
crtmqm

Linux *HA-RDQM löschen*
Mit dem Befehl **`dltmqm`** können Sie einen replizierten Datenwarteschlangenmanager (RDQM) mit hoher Verfügbarkeit löschen.

## Informationen zu diesem Vorgang

Sie müssen den Befehl ausführen, um den RDQM auf dem Primärknoten des RDQM zu löschen. RDQM muss zuerst beendet werden. Sie können den Befehl als mqm-Benutzer ausführen, wenn dieser Benutzer über die erforderlichen Zugriffsrechte für "sudo" verfügt. Andernfalls müssen Sie den Befehl als Root ausführen. Nachdem die Ressourcen, die dem primären Warteschlangenmanager zugeordnet sind, gelöscht

wurden, versucht der Befehl, die sekundären Warteschlangenmanager unter Verwendung von ssh zu löschen, um eine Verbindung zu den anderen Knoten herzustellen. Wenn dieser Löschvorgang fehlschlägt, müssen Sie `dltmqm` manuell auf den anderen Knoten ausführen, um den Prozess abzuschließen. Auf einem Sekundärknoten schlägt der Befehl fehl, wenn der primäre WS-Manager noch nicht gelöscht wurde.

## Prozedur

- Geben Sie den folgenden Befehl ein, um einen RDQM zu löschen:

```
dltmqm RDQM_name
```

**Zugehörige Verweise**
dltmqm

*Warteschlangenmanager als HA-RDQM-Warteschlangenmanager migrieren*

Sie können einen vorhandenen Warteschlangen migrieren, um einen replizierten Datenwarteschlangenmanager (RDQM) mit hoher Verfügbarkeit (HA) zu erhalten, indem Sie die zugehörigen persistenten Daten sichern und diese Daten anschließend in einem neu erstellten RDQM-Warteschlangenmanager mit dem gleichen Namen wiederherstellen.

## Informationen zu diesem Vorgang

Für replizierte Datenwarteschlangenmanager mit hoher Verfügbarkeit ist ein dedizierter logischer Datenträger (Dateisystem) und die Konfiguration der Plattenreplikation und HA-Steuerung erforderlich. Diese Komponenten werden nur konfiguriert, wenn ein neuer Warteschlangenmanager erstellt wird. Ein vorhandener Warteschlangenmanager kann migriert werden, damit ein RDQM verwendet wird, indem die zugehörigen persistenten Daten gesichert und anschließend in einem neu erstellten RDQM-Warteschlangenmanager mit dem gleichen Namen wiederhergestellt werden. Durch diese Vorgehensweise bleiben die Konfiguration, der Status und die persistenten Nachrichten des Warteschlangenmanagers so erhalten, wie sie zum Zeitpunkt der Sicherungserstellung waren.

**Anmerkung:** Sie können einen Warteschlangenmanager nur aus einer Version von IBM MQ migrieren, die identisch oder niedriger ist als die Version, auf der RDQM installiert ist. Das Betriebssystem und die Architektur müssen ebenfalls identisch sein. Andernfalls müssen sie einen neuen Warteschlangenmanager auf Ihrer Zielplattform erstellen; siehe Warteschlangenmanager in ein anderes Betriebssystem verschieben.

Die folgenden Bedingungen sollten vor der Migration eines Warteschlangenmanagers erfüllt sein:

- Bewerten Sie Ihre Anforderungen an die Hochverfügbarkeit und lesen Sie den Abschnitt „RDQM-Hochverfügbarkeit" auf Seite 609.
- Überprüfen Sie die Anwendungen und Warteschlangenmanager, die eine Verbindung zum Warteschlangenmanager herstellen. Beachten Sie dabei die Änderungen, die für die Weiterleitung der Verbindungen an den RDQM-Knoten erforderlich sind, auf dem der Warteschlangenmanager ausgeführt wird. Wenn Sie beispielsweise die RDQM-Hochverfügbarkeit konfigurieren, können Sie eine variable IP-Adresse verwenden (siehe „Variable IP-Adresse erstellen und löschen" auf Seite 628).
- Stellen Sie RDQM-Knoten für die von Ihnen ausgewählte Konfiguration bereit oder geben Sie diese an. Weitere Informationen zu den Systemvoraussetzungen für RDQM finden Sie unter „Voraussetzungen für RDQM HA-Lösung" auf Seite 611.
- Installieren Sie IBM MQ Advanced mit der integrierten RDQM-Funktion auf jedem Knoten.
- Konfigurieren Sie die RDQM-HA-Gruppenkonfiguration (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)" auf Seite 615).
- Überprüfen Sie optional die RDQM-Konfiguration mit einem Testwarteschlangenmanager, der anschließend gelöscht werden kann. Das Überprüfen der Konfiguration wird empfohlen, um mögliche Probleme vor der Migration des Warteschlangenmanagers zu ermitteln und zu beheben.
- Überprüfen Sie die Sicherheitskonfiguration für den Warteschlangenmanager und replizieren Sie anschließend die erforderlichen lokalen Benutzer und Gruppen auf jedem RDQM-Knoten.

- Überprüfen Sie die Warteschlangenmanager- und Kanalkonfiguration, um zu ermitteln, ob API-Exits, Kanalexits oder Exits zur Datenkonvertierung verwendet werden. Installieren Sie die erforderlichen Exits auf jedem RDQM-Knoten.
- Überprüfen Sie alle Warteschlangenmanagerservices, die definiert wurden, und installieren und konfigurieren Sie anschließend die erforderlichen Prozesse auf jedem RDQM-Knoten.

## Vorgehensweise

1. Sichern Sie den vorhandenen Warteschlangenmanager:
   a) Stoppen Sie den vorhandenen Warteschlangenmanager, indem Sie den Befehl `endmqm -w` ausgeben, mit dem das Beenden verzögert wird, oder indem Sie den Befehl `endmqm -i` für die sofortige Beendigung ausgeben. Dieser Schritt ist wichtig, um sicherzustellen, dass die Daten in der Sicherung konsistent sind.
   b) Ermitteln Sie die Position des Datenverzeichnisses des Warteschlangenmanagers, indem Sie die IBM MQ-Konfigurationsdatei `mqs.ini` anzeigen. Unter Linux befindet sich diese Datei im Verzeichnis `/var/mqm`. Weitere Informationen zu `mqs.ini` finden Sie unter „IBM MQ-Konfigurationsdatei, mqs.ini" auf Seite 91.

      Suchen Sie die Zeilengruppe `QueueManager` für den Warteschlangenmanager in der Datei. Wenn die Zeilengruppe einen Schlüssel mit der Bezeichnung `DataPath` enthält, handelt es sich bei dem zugehörigen Wert um das Datenverzeichnis für den Warteschlangenmanager. Wenn der Schlüssel nicht vorhanden ist, kann das Datenverzeichnis für den Warteschlangenmanager mithilfe der Werte für die Schlüssel `Prefix` und `Directory` ermittelt werden. Das Datenverzeichnis für den Warteschlangenmanager ist eine Verknüpfung dieser Werte im Format *Präfix*/qmgrs/*Verzeichnis*. Weitere Informationen zur Zeilengruppe 'QueueManager' finden Sie unter „Zeilengruppe 'QueueManager' in der Datei 'mqs.ini'" auf Seite 102.
   c) Erstellen Sie eine Sicherung des Warteschlangenmanager-Datenverzeichnisses. Unter Linux können Sie dies mit dem Befehl **tar** tun. Wenn Sie zum Beispiel das Datenverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```

   d) Ermitteln Sie die Position der Warteschlangenmanager-Protokolldatei, indem Sie sich die IBM MQ Warteschlangenmanager-Konfigurationsdatei `qm.ini` ansehen. Diese Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers. Weitere Informationen zur Datei finden Sie unter „Warteschlangenmanagerkonfigurationsdateien, qm.ini" auf Seite 104.

      Das Protokollverzeichnis für den Warteschlangenmanager wird als Wert des Schlüssels `LogPath` in der Zeilengruppe `Log` definiert. Weitere Informationen zu der Zeilengruppe finden Sie unter „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140.
   e) Erstellen Sie eine Sicherung des Warteschlangenmanager-Protokollverzeichnisses. Unter Linux können Sie dies mit dem Befehl 'tar' tun. Wenn Sie z. B. das Protokollverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

      ```
      tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
      ```

   f) Erstellen Sie eine Sicherungskopie aller Zertifikatsrepositorys, die vom Warteschlangenmanager verwendet werden, wenn diese sich nicht im Datenverzeichnis für den Warteschlangenmanager befinden. Stellen Sie sicher, dass die Schlüsseldatenbankdatei und die Kennwortstashdatei gesichert werden. Weitere Informationen zum Schlüsselrepository für den Warteschlangenmanager finden Sie in den Abschnitten Das SSL/TLS-Schlüsselrepository und Schlüsselrepository für einen Warteschlangenmanager suchen. Weitere Informationen zum Suchen des AMS-Schlüsselspeichers, wenn der Warteschlangenmanager für die Verwendung des Abfangprozesses für den Nachrichtenkanalagenten (Message Channel Agent, MCA) in AMS konfiguriert ist, finden Sie unter Überwachung des Nachrichtenkanalagenten (MCA).

g) Der vorhandene Warteschlangenmanager ist nicht mehr erforderlich und kann deshalb gelöscht werden. Wo dies möglich ist, sollten Sie den vorhandenen Warteschlangenmanager allerdings erst dann löschen, wenn er auf dem Zielsystem erfolgreich wiederhergestellt wurde. Durch das Zurückstellen des Löschvorgangs wird sichergestellt, dass der Warteschlangenmanager erneut gestartet werden kann, wenn der Migrationsprozess nicht erfolgreich abgeschlossen wird.

**Anmerkung:** Wenn Sie das Löschen des vorhandenen Warteschlangenmanagers zurückstellen, starten Sie ihn nicht erneut. Es ist wichtig, dass der Warteschlangenmanager beendet bleibt, da weitere Änderungen an der Konfiguration oder dem Status während der Migration verloren gehen.

2. Bereiten Sie den primären RDQM-Knoten vor:
   a) Erstellen Sie einen neuen RDQM-Warteschlangenmanager mit dem gleichen Namen wie der Warteschlangenmanager, den Sie gesichert haben. Stellen Sie sicher, dass das Dateisystem, das dem RDQM-Warteschlangenmanager durch **crtmqm** zugeordnet wurde, ausreichend ist für die Daten, primären und sekundären Protokolle für den vorhandenen Warteschlangenmanager sowie für einen zusätzlichen Speicherbereich für zukünftige Erweiterungen. Informationen zum Erstellen eines RDQM-Warteschlangenmanagers finden Sie unter „HA-RDQM erstellen” auf Seite 619.
   b) Bestimmen Sie den primären RDQM-Knoten für den Warteschlangenmanager. Informationen zum Ermitteln des Primärknotens finden Sie im Abschnitt rdqmstatus (RDQM-Status anzeigen).
   c) Falls der RDQM-Warteschlangenmanager gestartet ist, stoppen Sie ihn auf dem primären RDQM-Knoten mit dem Befehl `endmqm -w` oder `endmqm -i`.
   d) Ermitteln Sie auf dem primären RDQM-Knoten die Daten- und Protokollverzeichnisse für den RDQM-Warteschlangenmanager (verwenden Sie die in den Schritten 1b und 1d beschriebenen Methoden).
   e) Löschen Sie auf dem primären RDQM-Knoten den Inhalt der RDQM-Warteschlangenmanager-Daten- und -Protokollverzeichnisse, aber nicht die Verzeichnisse selbst.
3. Stellen Sie den Warteschlangenmanager auf dem primären RDQM-Knoten wieder her:
   a) Kopieren Sie die Sicherungsdateien aus den Daten- und Protokollverzeichnissen des Warteschlangenmanagers auf den primären RDQM-Knoten ebenso wie alle separaten Sicherungen von Zertifikatsrepositorys, die vom Warteschlangenmanager verwendet werden.
   b) Stellen Sie die Sicherung des Datenverzeichnisses des Warteschlangenmanagers im leeren Datenverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass das Eigentumsrecht und die Berechtigungen der Dateien beibehalten werden. Wenn die Sicherungsdateien mithilfe des Beispiels für den Befehl 'tar' in Schritt 1c erstellt wurde, kann der folgende Befehl vom Rootbenutzer für die Wiederherstellung verwendet werden:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

   c) Stellen Sie die Sicherung des Protokollverzeichnisses des Warteschlangenmanagers im leeren Protokollverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass das Eigentumsrecht und die Berechtigungen der Dateien beibehalten werden. Wenn die Sicherungsdateien mithilfe des Beispiels für den Befehl 'tar' in Schritt 1e erstellt wurde, kann der folgende Befehl vom Rootbenutzer für die Wiederherstellung verwendet werden:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

   d) Bearbeiten Sie die zurückgeschriebene Konfigurationsdatei des Warteschlangenmanagers `qm.ini` im Datenverzeichnis für den RDQM-Warteschlangenmanager. Aktualisieren Sie den Wert des Schlüssels `LogPath` in der Zeilengruppe `Log`, um das Protokollverzeichnis für den RDQM-Warteschlangenmanager anzugeben.

      Überprüfen Sie weitere in der Konfigurationsdatei definierte Dateipfade und aktualisieren Sie diese, falls erforderlich. Sie müssen z. B. möglicherweise die folgenden Pfade aktualisieren:

      - Den Pfad für Fehlerprotokolldateien, die von Diagnosenachrichtenservices generiert werden.
      - Den Pfad für Exits, die vom Warteschlangenmanager benötigt werden.
      - Den Pfad für Switchloaddateien, wenn der Warteschlangenmanager ein XA-Transaktionskoordinator ist.

e) Wenn der Warteschlangenmanager für die Verwendung des Abfangprozesses für den Nachrichtenkanalagenten (MCA) in AMS konfiguriert ist, kopieren Sie den AMS-Schlüsselspeicher in die neue RDQM-Installation und prüfen und aktualisieren anschließend die Konfiguration. Der Schlüsselspeicher muss in jedem RDQM-Knoten verfügbar sein; wenn er sich also nicht im replizierten Dateisystem für den Warteschlangenmanager befindet, muss er stattdessen auf jeden Knoten kopiert werden. Weitere Informationen finden Sie unter Überwachung des Nachrichtenkanalagenten (MCA).

f) Stellen Sie sicher, dass der Warteschlangenmanager mit dem Befehl **dspmq** angezeigt wird und dass für ihn der Status 'beendet' gemeldet wird. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-HA-Warteschlangenmanager:

```
$ dspmq -o status -o ha

QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

g) Stellen Sie sicher, dass die wiederhergestellten Warteschlangenmanagerdaten auf den sekundären RDQM-Knoten repliziert wurden; verwenden Sie dazu den Befehl **rdqmstatus**, um den Status des Warteschlangenmanagers anzuzeigen. Der HA-Status sollte auf jedem der Knoten als `Normal` gemeldet werden. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-HA-Warteschlangenmanager:

```
$ rdqmstatus -m QM1
Node:                           mqhavm10-adm
Queue manager status:           Ended normally
Queue manager file system:      50MB used, 0.2GB allocated [42%]
HA role:                        Primary
HA status:                      Normal
HA control:                     Disabled
HA current location:            This node
HA preferred location:          This node
HA floating IP interface:       None
HA floating IP address:         None

Node:                           mqhavm11-adm
HA status:                      Normal

Node:                           mqhavm12-adm
HA status:                      Normal
```

h) Starten Sie den Warteschlangenmanager auf dem primären RDQM-Knoten.

i) Stellen Sie eine Verbindung zum Warteschlangenmanager her und aktualisieren Sie den Wert des Warteschlangenmanagerattributs SSLKEYR, um die neue Position des Zertifikatsrepositorys für den Warteschlangenmanager anzugeben. Standardmäßig ist der Wert dieses Attributs auf *queue_manager_data_directory*`/ssl/key`gesetzt. Das Zertifikatsrepository muss sich auf jedem RDQM-Knoten an der gleichen Position befinden. Wenn sich das Repository nicht im replizierten Dateisystem für den Warteschlangenmanager befindet, muss es stattdessen auf jeden Knoten kopiert werden.

j) Überprüfen Sie die IBM MQ-Objektdefinitionen für den Warteschlangenmanager und aktualisieren Sie den Wert der Objektattribute, die auf geänderte Netzeinstellungen, das IBM MQ-Installationsverzeichnis oder das Datenverzeichnis des Warteschlangenmanagers verweisen, einschließlich der folgenden Objekte:

- Lokale IP-Adressen, die von Listener verwendet werden (Attribut `IPADDR`).
- Lokale IP-Adressen, die von Kanälen verwendet werden (Attribut `LOCLADDR`).
- Lokale IP-Adressen, die für Clusterempfängerkanäle definiert werden (Attribut `CONNAME`).
- Lokale IP-Adressen, die für Informationsobjekte zur Kommunikation definiert werden (Attribut`GRPADDR`).
- Systempfade, die für Prozess- und Serviceobjektdefinitionen definiert werden.

k) Stoppen Sie den Warteschlangenmanager und starten Sie ihn erneut, um sicherzustellen, dass die Änderungen wirksam werden.

l) Wiederholen Sie Schritt 3j für ferne Warteschlangenmanager und nehmen Sie entsprechende Einstellungen für Anwendungen vor, die eine Verbindung zum migrierten Warteschlangenmanager herstellen, einschließlich der Folgenden:

- Kanalverbindungsnamen (Attribut CONNAME).
- Kanalauthentifizierungsregeln, die eingehende Verbindungen vom Warteschlangenmanager auf Basis der IP-Adresse oder des Hostnamens einschränken.
- Definitionstabellen für den Clientkanal (CCDTs), Einstellungen des Domänennamens (DNS), Netzweiterleitung oder entsprechende Verbindungsinformationen.

m) Führen Sie für jeden RDQM-Knoten eine gesteuerte Funktionsübernahme des Warteschlangenmanagers durch, um sicherzustellen, dass die erforderliche Konfiguration erfolgreich eingerichtet wurde, siehe „Festlegen der bevorzugten Position für einen RDQM" auf Seite 628.

*Größe des Dateisystems für einen HA-RDQM-Warteschlangenmanager ändern*
Um die Größe des Dateisystems für einen vorhandenen HA-RDQM-Warteschlangenmanager mit replizierten Daten zu ändern, sichern Sie seine persistenten Daten und stellen anschließend die Daten in einem neu erstellten RDQM-Warteschlangenmanager wieder her, der denselben Namen, aber ein Dateisystem mit einer anderen Größe hat.

## Informationen zu diesem Vorgang

Hochverfügbarkeits-Warteschlangenmanager mit replizierten Daten (RDQM) benötigen einen dedizierten logischen Datenträger (Dateisystem) und die Konfiguration der Plattenreplikation und der HA-Steuerung. Diese Komponenten werden nur konfiguriert, wenn ein neuer Warteschlangenmanager erstellt wird. Das Dateisystem kann nach seiner Erstellung nicht geändert werden, da es auf jedem Knoten die gleiche Größe haben muss. Um die Größe des Dateisystems für einen vorhandenen Warteschlangenmanager mit replizierten Daten (RDQM) zu ändern, können Sie seine persistenten Daten sichern und anschließend die Daten in einem neu erstellten RDQM-Warteschlangenmanager wiederherstellen, der denselben Namen, aber ein Dateisystem mit einer anderen Größe hat. Durch diese Vorgehensweise bleiben die Konfiguration, der Status und die persistenten Nachrichten des Warteschlangenmanagers so erhalten, wie sie zum Zeitpunkt der Sicherungserstellung waren.

## Vorgehensweise

1. Sichern Sie den vorhandenen RDQM-WS-Manager auf dem primären RDQM-Knoten:
   a) Bestimmen Sie den primären RDQM-Knoten für den Warteschlangenmanager. Informationen zum Ermitteln des Primärknotens finden Sie im Abschnitt rdqmstatus (RDQM-Status anzeigen).
   b) Falls der RDQM-Warteschlangenmanager gestartet ist, stoppen Sie ihn auf dem primären RDQM-Knoten mit dem Befehl **endmqm -w** oder **endmqm -i**.
   c) Ermitteln Sie die Position des Datenverzeichnisses des Warteschlangenmanagers, indem Sie die IBM MQ-Konfigurationsdatei `mqs.ini` anzeigen. Unter Linux befindet sich diese Datei im Verzeichnis `/var/mqm`. Weitere Informationen zu `mqs.ini` finden Sie unter „IBM MQ-Konfigurationsdatei, mqs.ini" auf Seite 91.

      Suchen Sie die Zeilengruppe `QueueManager` für den Warteschlangenmanager in der Datei. Das Warteschlangenmanager-Datenverzeichnis ist der Wert des Schlüssels mit dem Namen `DataPath`. Weitere Informationen zu der Zeilengruppe `QueueManager` finden Sie unter „Zeilengruppe 'QueueManager' in der Datei 'mqs.ini'" auf Seite 102.
   d) Erstellen Sie eine Sicherung des Warteschlangenmanager-Datenverzeichnisses. Unter Linux können Sie dies mit dem Befehl **tar** tun. Wenn Sie zum Beispiel das Datenverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

e) Ermitteln Sie die Position der Warteschlangenmanager-Protokolldatei, indem Sie sich die IBM MQ Warteschlangenmanager-Konfigurationsdatei `qm.ini` ansehen. Diese Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers. Weitere Informationen zur Datei finden Sie unter „Warteschlangenmanagerkonfigurationsdateien, qm.ini" auf Seite 104.

Das Warteschlangenmanager-Protokollverzeichnis ist als Wert des Schlüssels `LogPath` in der Zeilengruppe 'Log' definiert. Weitere Informationen zu der Zeilengruppe finden Sie unter „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140.

f) Erstellen Sie eine Sicherung des Warteschlangenmanager-Protokollverzeichnisses. Unter Linux können Sie dies mit dem Befehl **tar** tun. Wenn Sie z. B. das Protokollverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

g) Löschen Sie den vorhandenen RDQM-Warteschlangenmanager.

2. Stellen Sie den Warteschlangenmanager mit einem Dateisystem mit der erforderlichen Größe wieder her:

a) Erstellen Sie einen neuen RDQM-Warteschlangenmanager mit dem gleichen Namen wie der Warteschlangenmanager, den Sie gesichert haben. Stellen Sie sicher, dass das Dateisystem, das dem RDQM-Warteschlangenmanager durch **crtmqm** zugeordnet wurde, die benötigte Größe hat und groß genug ist, um die Daten, primären und sekundären Protokolle für den vorhandenen Warteschlangenmanager aufzunehmen sowie zusätzlichen Speicherplatz für zukünftige Erweiterungen zu bieten. Informationen zum Erstellen eines RDQM-Warteschlangenmanagers finden Sie unter „HA-RDQM erstellen" auf Seite 619.

b) Bestimmen Sie den primären RDQM-Knoten für den Warteschlangenmanager. Informationen zum Ermitteln des Primärknotens finden Sie im Abschnitt rdqmstatus (RDQM-Status anzeigen).

c) Wenn der RDQM-Warteschlangenmanager auf dem primären RDQM-Knoten gestartet ist, stoppen Sie ihn mit dem Befehl **endmqm -w** oder **endmqm -i** .

d) Bestimmen Sie auf dem primären RDQM-Knoten die neue Position der Daten- und Protokollverzeichnisse für den RDQM-Warteschlangenmanager (verwenden Sie die in den Schritten 1c und 1e beschriebenen Methoden).

e) Löschen Sie auf dem primären RDQM-Knoten den Inhalt der RDQM-Warteschlangenmanager-Daten- und -Protokollverzeichnisse, aber nicht die Verzeichnisse selbst.

f) Stellen Sie auf dem primären RDQM-Knoten die Sicherung des Warteschlangenmanager-Datenverzeichnisses in dem leeren Datenverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass die Dateieigentumsrechte und die Berechtigungen beibehalten werden. Wenn die Sicherung mit dem in Schritt 1d gezeigten Beispielbefehl **tar** erstellt wurde, kann vom Rootbenutzer für die Wiederherstellung der folgende Befehl verwendet werden:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

g) Stellen Sie auf dem primären RDQM-Knoten die Sicherung des Warteschlangenmanager-Protokollverzeichnisses in dem leeren Protokollverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass die Dateieigentumsrechte und die Berechtigungen beibehalten werden. Wenn die Sicherung mit dem in Schritt 1f gezeigten Beispielbefehl **tar** erstellt wurde, kann vom Rootbenutzer für die Wiederherstellung der folgende Befehl verwendet werden:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

h) Bearbeiten Sie auf dem primären RDQM-Knoten die zurückgeschriebene Konfigurationsdatei des Warteschlangenmanagers `qm.ini` im Datenverzeichnis für den neuen RDQM-Warteschlangenmanager. Aktualisieren Sie den Wert des Schlüssels `LogPath` in der Zeilengruppe `Log`, so dass dort das in Schritt 2d bestimmte Protokollverzeichnis für den neuen RDQM-Warteschlangenmanager angegeben ist. Überprüfen Sie weitere in der Konfigurationsdatei definierte Dateipfade und aktualisieren Sie diese, falls erforderlich. Sie müssen z. B. möglicherweise die folgenden Pfade aktualisieren:

- Den Pfad für Fehlerprotokolldateien, die von Diagnosenachrichtenservices generiert werden.
- Den Pfad für Exits, die vom Warteschlangenmanager benötigt werden.
- Den Pfad für Switchloaddateien, wenn der Warteschlangenmanager ein XA-Transaktionskoordinator ist.

i) Stellen Sie sicher, dass der Warteschlangenmanager mit dem Befehl **dspmq** angezeigt wird und dass für ihn der Status 'beendet' gemeldet wird. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-HA-Warteschlangenmanager:

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

j) Stellen Sie sicher, dass die wiederhergestellten Warteschlangenmanagerdaten auf den sekundären RDQM-Knoten repliziert wurden; verwenden Sie dazu den Befehl **rdqmstatus**, um den Status des Warteschlangenmanagers anzuzeigen. Der HA-Status sollte auf jedem der Knoten als `Normal` gemeldet werden. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-HA-Warteschlangenmanager:

```
$ rdqmstatus -m QM1
Node: mqhavm10-adm
Queue manager status:         Ended normally
Queue manager file system:    50MB used, 0.2GB
allocated [42%]
HA role:                      Primary
HA status:                    Normal
HA control:                   Disabled
HA current location:          This node
HA preferred location:        This node
HA floating IP interface:     None
HA floating IP address:       None
Node:                         mqhavm11-adm
HA status:                    Normal
Node:                         mqhavm12-adm
HA status:                    Normal
```

k) Starten Sie den Warteschlangenmanager auf dem primären RDQM-Knoten.

l) Führen Sie für jeden RDQM-Knoten eine gesteuerte Funktionsübernahme des Warteschlangenmanagers durch, um sicherzustellen, dass die erforderliche Konfiguration erfolgreich eingerichtet wurde, siehe „Festlegen der bevorzugten Position für einen RDQM” auf Seite 628.

*Persistenten Anwendungsstatus speichern*
Persistente Statusinformationen zu Anwendungen können zusammen mit anderen Warteschlangenmanagerdaten gespeichert werden.

Jeder IBM MQ-Warteschlangenmanager verfügt über ein dediziertes Dateisystem für seinen persistenten Status, was sowohl seine Warteschlangendaten als auch das Wiederherstellungsprotokoll einschließt. In einer RDQM-Konfiguration wird das Dateisystem durch einen logischen Datenträger gesichert, der zwischen den Linux-Systemen (Knoten) repliziert wird. Das Dateisystem enthält ein `userdata`-Verzeichnis, das Sie zum Speichern persistenter Statusinformationen für Ihre Anwendungen verwenden können. Wenn ein RDQM (Replicated Data Queue Manager) auf einen anderen Knoten in Ihrer RDQM-Konfiguration verschoben wird, verfügen Sie auf diese Weise sowohl über den Anwendungs- als auch den Warteschlangenmanagerkontext. Weitere Informationen finden Sie im Abschnitt Verzeichnisinhalt auf Unix- und Linux-Systemen.

Wenn Sie den Anwendungsstatus im Verzeichnis `userdata` speichern, müssen Sie sich darüber im Klaren sein, dass Daten, die an diese Position geschrieben werden, möglicherweise den verfügbaren Plattenspeicherplatz belegen, der dem Warteschlangenmanager zugeordnet ist. Sie müssen sicherstellen, dass für den Warteschlangenmanager genügend Plattenspeicherplatz zum Speichern von Warteschlangendaten, Protokollen und anderen persistenten Statusinformationen verfügbar ist.

Das Verzeichnis `userdata` hat den Benutzer 'mqm' und das Gruppeneigentum, und es ist weltweit lesbar, sodass Benutzer darauf zugreifen können, ohne sich in der Administratorgruppe von IBM MQ (d. h. mqm) befinden zu müssen. Sie können die Berechtigungen des Verzeichnisses `userdata` nicht ändern, aber Sie können Inhalte in ihr erstellen, unabhängig davon, welche Eigentümer und Berechtigungen Sie benötigen.

Bei der Übernahme eines RDQM-Warteschlangenmanagers wird der Warteschlangenmanager beendet und sein Dateisystem wird auf seinem aktuellen RDQM-Knoten abgehängt. Das Dateisystem wird anschließend auf einem anderen Knoten in der RDQM-Konfiguration angehängt und der Warteschlangenmanager dort erneut gestartet. Ein Dateisystem kann nicht abgehängt werden, solange es einen Prozess mit einer offenen Kennung für eine seiner Dateien gibt. Um sicherzustellen, dass eine Warteschlangenmanagerübernahme auch dann abgeschlossen werden kann, wenn das Dateisystem des Warteschlangenmanagers nicht abgehängt werden kann, wird an alle Prozesse mit einer offenen Dateikennung ein SIGTERM-Signal gesendet, gefolgt von einem SIGKILL, falls die offenen Kennungen nicht freigegeben werden. Ihre Anwendungen müssen so konzipiert sein, dass sie korrekt auf SIGTERM reagieren. Wenn Anwendungen oder Prozesse als Warteschlangenmanagerservice konfiguriert sind, können sie während einer verwalteten Übernahme beim Herunterfahren des Warteschlangenmanagers beendet werden, bevor das Dateisystem abgehängt wird. Wenn eine Anwendung oder ein Prozess nicht als Warteschlangenmanagerservice konfiguriert ist oder eine nicht verwaltete Übernahme erfolgt, z. B. nach einem Verlust des Quorums, ist es wahrscheinlich, dass Signale gesendet werden, um das Dateisystem freizugeben.

## Linux *Festlegen der bevorzugten Position für einen RDQM*

Die bevorzugte Position für einen replizierten Datenwarteschlangenmanager (RDQM) gibt den Knoten an, auf dem der RDQM ausgeführt werden soll, wenn dieser Knoten verfügbar ist.

### Informationen zu diesem Vorgang

Die bevorzugte Position ist der Name des Knotens, auf dem der Warteschlangenmanager den Warteschlangenmanager ausführen soll, wenn sich die HA-Gruppe in einem normalen Status befindet (alle Knoten und Verbindungen sind verfügbar). Die bevorzugte Position wird beim Erstellen des Warteschlangenmanagers mit dem Namen des Primärknotens initialisiert. Sie können die Befehle ausführen, um die bevorzugte Position auf einem der drei Knoten festzulegen. Sie müssen ein Benutzer sein, der sowohl zu den Gruppen `mqm` als auch zu `haclient` gehört.

### Prozedur

- Geben Sie den folgenden Befehl ein, um den lokalen oder angegebenen Knoten als bevorzugte Position für den benannten Warteschlangenmanager zuzuordnen:

  ```
  rdqmadm -p -m qmname [ -n nodename[,nodename  ]
  ```

  Hierbei steht *qmname* für den Namen des RDQM-Systems, für das Sie die bevorzugte Position angeben, und *nodename* ist optional der Name des bevorzugten Knotens.

  Wenn sich die HA-Gruppe in einem normalen Status befindet und der bevorzugte Standort nicht der aktuelle Primärknoten ist, wird der Warteschlangenmanager gestoppt und erneut an der neuen bevorzugten Position erneut gestartet. Sie können eine durch Kommas getrennte Liste mit zwei Knotennamen angeben, um eine zweite Vorgabe für die bevorzugte Position zuzuordnen.
- Geben Sie den folgenden Befehl ein, um den bevorzugten Standort zu löschen, damit der Warteschlangenmanager nicht automatisch zu einem Knoten zurückkehren kann, wenn er wiederhergestellt wird:

  ```
  rdqmadm -p -m qmname -d
  ```

**Zugehörige Verweise**

rdqmadm (Verwaltung replizierter Daten-WS-Manager-Cluster)

## Linux *Variable IP-Adresse erstellen und löschen*

Eine variable IP-Adresse ermöglicht einem Client die Verwendung derselben IP-Adresse für einen replizierten Datenwarteschlangenmanager (RDQM), unabhängig davon, welcher Knoten in der HA-Gruppe ausgeführt wird. (Die Verwendung einer variablen IP-Adresse ist optional.)

## Informationen zu diesem Vorgang

Sie können eine variable IP-Adresse mit dem Befehl **rdqmint** erstellen oder löschen. Die Gleitadresse wird an eine benannte physische Schnittstelle auf dem Primärknoten für den RDQM gebunden. Wenn RDQM nicht ausgeführt wird und ein anderer Knoten zum primären Knoten wird, wird die variable IP-Adresse an eine Schnittstelle mit demselben Namen auf dem neuen primären Server gebunden. Die physischen Schnittstellen auf den drei Knoten müssen zu demselben Teilnetz gehören wie die variable IP-Adresse. Das folgende Diagramm veranschaulicht die Verwendung einer variablen IP-Adresse.

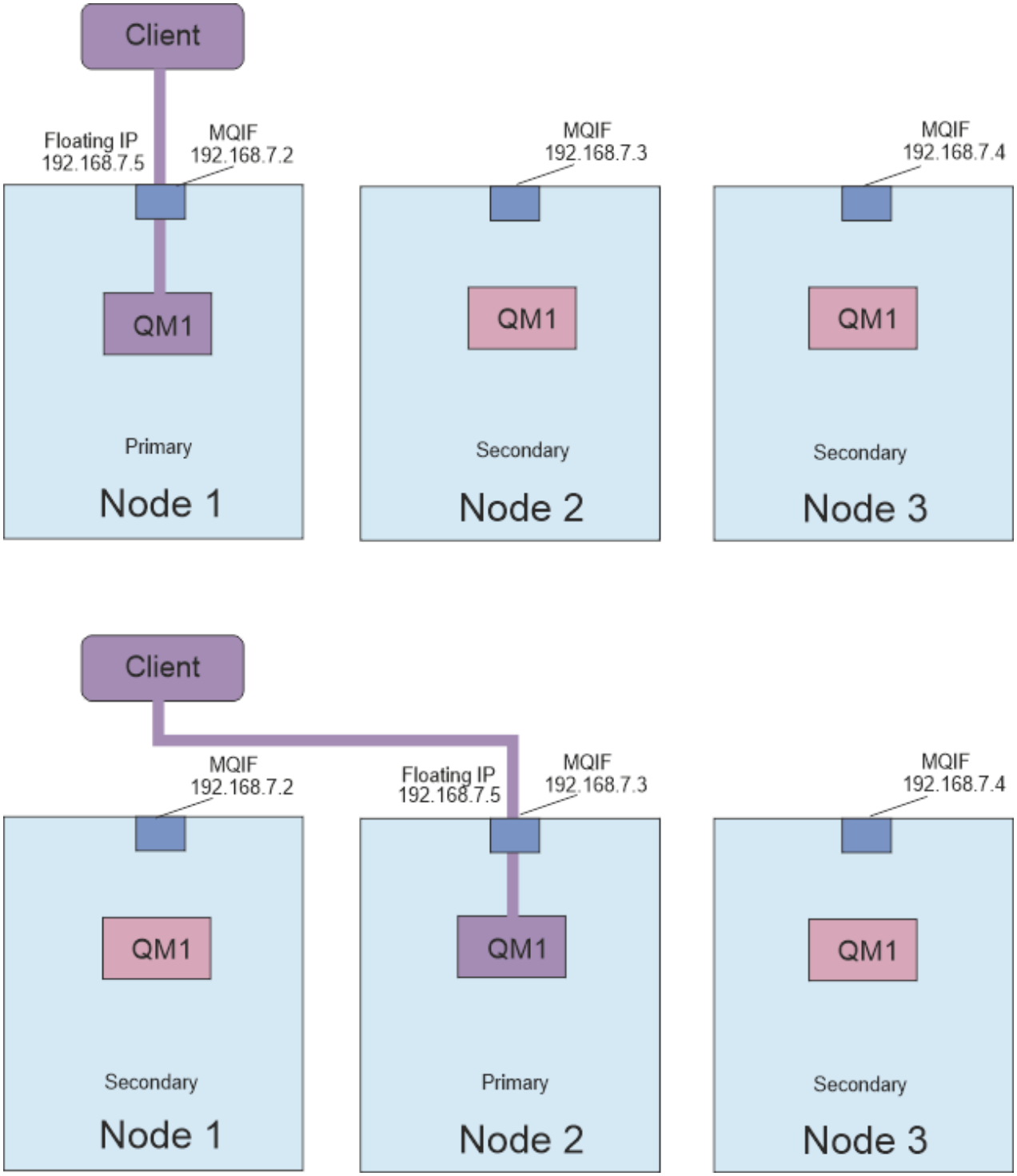



*Abbildung 80. Variable IP-Adresse*

Sie müssen ein Benutzer in den Gruppen `mqm` und `haclient` sein, um den Befehl **rdqmint** ausführen zu können. Sie können die variable IP-Adresse auf dem Primärknoten für den RDQM oder einen der Sekundärknoten erstellen oder löschen.

**Anmerkung:** Sie können nicht die gleiche variable IP-Adresse für mehrere RDQMs verwenden, die variable IP-Adresse muss für jeden einzelnen RDQM eindeutig sein.

## Prozedur

- Geben Sie den folgenden Befehl ein, um eine variable IP-Adresse für einen RDQM zu erstellen:

  ```
  rdqmint -m qmname -a  -f ipv4address -l interfacename
  ```

  Dabei gilt:

  ***QMNAME***
  Ist der Name des RDQM-Systems, für das Sie die variable IP-Adresse erstellen.

  ***ipv4address***
  Die variable IP-Adresse im ipv4-Format.

  Die Floating-IP-Adresse muss eine gültige IPv4-Adresse sein, die nicht bereits auf einem beliebigen HA-Knoten definiert ist, und sie muss zu demselben Teilnetz gehören wie die für die lokale Schnittstelle definierten statischen IP-Adressen.

  ***interfacename***
  Der Name der physischen Schnittstelle auf dem Primärknoten, an die die Bindung gebunden werden soll.

  For example:

  ```
  rdqmint -m QM1 -a -f 192.168.7.5 -l MQIF
  ```

- Geben Sie den folgenden Befehl ein, um eine vorhandene variable IP-Adresse zu löschen:

  ```
  rdqmint -m qmname -d
  ```

**Zugehörige Verweise**
rdqmint (variable IP-Adresse für RDQM hinzufügen oder löschen)

# Linux *Starten, Stoppen und Anzeigen des Status eines HA-RDQM*

Zum Starten und Stoppen eines Warteschlangenmanagers mit replizierten Daten (Replicated Data Queue Manager, RDQM) sowie zum Anzeigen seines aktuellen Status werden Varianten der standardmäßigen IBM MQ-Steuerbefehle verwendet.

## Informationen zu diesem Vorgang

Sie müssen die Befehle ausführen, mit denen der aktuelle Status eines replizierten Datenwarteschlangenmanagers (RDQM) als Benutzer, der sowohl zu den Gruppen `mqm` als auch zu `haclient` gehört, gestartet, gestoppt und angezeigt wird.

Sie müssen die Befehle ausführen, um einen Warteschlangenmanager auf dem Primärknoten für diesen Warteschlangenmanager zu starten und zu stoppen.

## Prozedur

- Geben Sie zum Starten eines RDQM folgenden Befehl auf dem Primärknoten des RDQM ein:

  ```
  strmqm qmname
  ```

  Hierbei steht *qmname* für den Namen des RDQM, den Sie starten wollen.

RDQM wird gestartet, und Pacemaker beginnt mit der Verwaltung des RDQM. Sie müssen die Option `-ns` mit `strmqm` angeben, wenn Sie andere `strmqm` -Optionen angeben möchten.

- Geben Sie zum Stoppen eines RDQM den folgenden Befehl auf dem Primärknoten des RDQM-Systems ein:

  ```
  endmqm qmname
  ```

  Hierbei steht *qmname* für den Namen des RDQM, den Sie stoppen wollen.

  Der Pacemaker wird für die Verwaltung von RDQM eingestellt, und der RDQM wird beendet. Alle anderen **endmqm** -Parameter können beim Stoppen eines RDQM verwendet werden.

- Geben Sie den folgenden Befehl ein, um den Status eines RDQM anzuzeigen:

  ```
  dspmq
  ```

  Die Statusinformationen, die ausgegeben werden, hängen davon ab, ob Sie den Befehl auf dem Primär- oder Sekundärknoten des RDQM ausführen. Wenn die Ausführung auf dem Primärknoten erfolgt, wird eine der normalen Statusnachrichten angezeigt, die von **dspmq** zurückgegeben werden. Wenn Sie den Befehl auf einem Sekundärknoten ausführen, wird der Status `running elsewhere` angezeigt. Wird **dspmq** beispielsweise auf dem Knoten RDQM7 ausgeführt wird, werden möglicherweise folgende Informationen zurückgegeben:

  ```
  QMNAME(RDQM8)                           STATUS(Running elsewhere)
  QMNAME(RDQM9)                           STATUS(Running elsewhere)
  QMNAME(RDQM7)                           STATUS(Running)
  ```

  Wenn der Primärknoten nicht verfügbar ist oder wenn **dspmq** von einem Benutzer ausgeführt wird, der nicht `root` oder ein Mitglied der Gruppe `haclient` ist, wird der Status `Unavailable` gemeldet. For example:

  ```
  QMNAME(RDQM8)              STATUS(Unavailable)
  QMNAME(RDQM9)              STATUS(Unavailable)
  QMNAME(RDQM7)              STATUS(Unavailable)
  ```

  Sie können den Befehl **dspmq -o ha** (oder **dspmq -o HA**) eingeben, um eine Liste der Warteschlangenmanager anzuzeigen, die einem Knoten bekannt sind, und ob es sich um RDQMs handelt. Beispiel:

  ```
  dspmq -o ha

  QMNAME(RDQM8)                                         HA(Replicated)
  QMNAME(RDQM9)                                         HA(Replicated)
  QMNAME(RDQM7)                                         HA(Replicated)
  QMNAME(QM7)                                           HA()
  ```

**Zugehörige Verweise**

dspmq (Warteschlangenmanager anzeigen)
endmqm (Warteschlangenmanager beenden)
strmqm (Warteschlangenmanager starten)

### V 9.3.0 *Fehlgeschlagene Ressourcenaktionen*

Fehlgeschlagene Ressourcenaktionen treten auf, wenn die Pacemaker-Komponente einer RDQM-Hochverfügbarkeitskonfiguration Probleme mit einer Ressource auf einem der Knoten in einer HA-Gruppe ermittelt.

Die RDQM-Hochverfügbarkeitslösung verwendet Pacemaker für die Überwachung und Verwaltung von Ressourcen (siehe „RDQM-Hochverfügbarkeit” auf Seite 609). Wenn Pacemaker bei der Ausführung einer Ressource auf einem Knoten einen Fehler ermittelt, werden diese Informationen mithilfe einer fehlgeschlagenen Ressourcenaktion aufgezeichnet. Einige fehlgeschlagene Ressourcenaktionen verhindern die Ausführung der Ressource und müssen gelöscht werden, damit Pacemaker die Ressource erneut starten kann.

Mit dem Befehl **rdqmstatus -m** können Sie ermitteln, ob fehlgeschlagene Ressourcenaktionen vorhanden sind, die das Starten eines Warteschlangenmanagers auf einem oder mehreren Knoten verhindern.

Mit dem Befehl **rdqmstatus -m** *Warteschlangenmanagername* **-a** können Sie die Einzelheiten der fehlgeschlagenen Ressourcenaktionen anzeigen, die einem Warteschlangenmanager zugeordnet sind. Führen Sie nach dieser Aktion den Befehl **rdqmclean** aus, um die entsprechenden fehlgeschlagenen Ressourcenaktionen zu löschen und somit eingeschränkte Ressourcen freizugeben. (Sie müssen außerdem die Probleme beheben, die die fehlgeschlagenen Ressourcenaktionen überhaupt verursacht haben.)

Die folgenden Ressourcen werden von Pacemaker in einer RDQM-Hochverfügbarkeitskonfiguration gesteuert und können Gegenstand fehlgeschlagener Ressourcenaktionen sein:

- Warteschlangenmanager
- Variable IP-Adresse
- RDQM-Steuerung
- Dateisystem
- DR-Replikation (DRBD)
- HA-Replikation (DRBD)

Jeder Ressourcentyp kann Gegenstand der folgenden Fehlertypen sein:

**Soft**
: Fehler des Typs 'Soft' sind temporär und Pacemaker versucht weiter, die Ressourcen wiederherzustellen, bis das Zeitlimit überschritten oder sie anderweitig gestoppt wird.

**Hard**
: Für einen Fehler des Typs 'Hard' ist ein Eingriff des Administrators erforderlich. Bei diesen Fehlern ist die Ausführung der Ressource auf einem bestimmten Knoten blockiert.

**Fatal**
: Für einen Fehler des Typs 'Fatal' ist ein Eingriff des Administrators erforderlich. Bei diesen Fehlern ist die Ausführung der Ressource auf allen Knoten blockiert.

Beispiele zum Status sowie fehlgeschlagene Aktionen in der Ressourcenwarteschlange finden Sie im Abschnitt „RDQM-und HA-Gruppenstatus anzeigen” auf Seite 632.

Mit dem Befehl **rdqmclean** können Sie alle fehlgeschlagenen Ressourcenaktionen, die einem bestimmten Warteschlangenmanager zugeordnet sind, oder alle fehlgeschlagenen Ressourcenaktionen in der RDQM-HA-Konfiguration löschen

**Anmerkung:** Einige fehlgeschlagene Ressourcenaktionen führen nicht dazu, dass die Ausführung eines Warteschlangenmanagers auf einem Knoten blockiert ist. Beispielsweise versucht Pacemaker nach einer unerwarteten Beendigung eines Warteschlangenmanagers, den Warteschlangenmanager auf dem Knoten erneut auszuführen, auf dem festgestellt wurde, dass er nicht aktiv ist. Bei einem erfolgreichen Start ist die Ausführung des Warteschlangenmanagers auf dem Knoten nicht mehr blockiert. Sie können die fehlgeschlagene Ressourcenaktion nur feststellen, indem Sie den Befehl **rdqmstatus -m** *Warteschlangenmanagername* **-a** ausführen.

**Zugehörige Tasks**

„RDQM-und HA-Gruppenstatus anzeigen” auf Seite 632
Sie können den Status der HA-Gruppe und von einzelnen replizierten Datenwarteschlangenmanagern (RDQMs) anzeigen.

**Zugehörige Verweise**

rdqmclean
rdqmstatus

## Linux *RDQM-und HA-Gruppenstatus anzeigen*

Sie können den Status der HA-Gruppe und von einzelnen replizierten Datenwarteschlangenmanagern (RDQMs) anzeigen.

## Informationen zu diesem Vorgang

Sie verwenden den Befehl **rdqmstatus** , um den Status einzelner RDQMs und der HA-Gruppe als Ganzes anzuzeigen.

V 9.3.0 Der Zusammenfassungsstatus für einen Knoten zeigt auch Informationen über das DRBD-Kernelmodul an, auf das RDQM angewiesen ist. Wenn Sie RDQM aktualisieren, ist es wichtig, sicherzustellen, dass die korrekte Version des DRBD Kernelmoduls installiert ist für die Version des RHEL Kernels, der auf dem System läuft. Der Status zeigt die Version des Betriebssystem-Kernels, die Kernelversion, für die das DRBD-Modul erstellt wurde, die DRBD-Version und den Status des DRBD-Kernelmoduls an.

Sie müssen ein Benutzer in den Gruppen `mqm` und `haclient` sein, um den Befehl **rdqmstatus** ausführen zu können. Sie können den Befehl auf jedem der drei Knoten ausführen.

## Prozedur

- Gehen Sie wie folgt vor, um den Zusammenfassungsstatus eines Knotens und der RDQMs anzuzeigen, die Teil der HA-Konfiguration sind:

```
rdqmstatus
```

Es werden z. B. die Identität des Knotens, auf dem Sie den Befehl ausgeführt haben, die Kernel- und DRBD-Details für diesen Knoten und der Status der RDQMs in der HA-Konfiguration angezeigt:

```
Node:                                mqhavm07.exampleco.com
OS kernel version:                   3.10.0-1160.15.2
DRBD OS kernel version:              3.10.0-1160
DRBD version:                        9.1.1
DRBD kernel module status:           Loaded

Queue manager name:                  RDQM8
Queue manager status:                Running elsewhere
HA current location:                 mqhavm08.exampleco.com
HA preferred location:               mqhavm08.exampleco.com
HA blocked location:                 None

Queue manager name:                  RDQM9
Queue manager status:                Running elsewhere
HA current location:                 mqhavm09.exampleco.com
HA preferred location:               mqhavm09.exampleco.com
HA blocked location:                 None

Queue manager name:                  RDQM7
Queue manager status:                Running
HA current location:                 This node
HA preferred location:               This node
HA blocked location:                 None
```

V 9.3.0 Der Status des DRBD-Kernelmoduls ist einer der folgenden Werte:

**Geladen**
Gibt an, dass das DRBD-Modul geladen wurde.

**Teilweise geladen**
Kann auftreten, wenn das DRBD-Modul geladen wurde, aber aufgrund einer Abweichung nicht ordnungsgemäß funktioniert.

**Nicht geladen**
Das DRBD-Modul ist nicht geladen. Diese Option kann auf einer neu installierten Konfiguration angezeigt werden, wenn noch keine RDQM-Warteschlangenmanager erstellt wurden.

**Nicht installiert**
Gibt an, dass das DRBD-Modul nicht installiert ist. oder dass IBM MQ die Betriebssystem-Kernelversion des DRBD-Moduls nicht bestimmen konnte.

**Zuvor installierte Version ist noch geladen**
Dieser Status kann auftreten, wenn ein neues DRBD-Modul installiert wird, während das vorhandene DRBD-Modul ausgeführt wird (d. h. ein RDQM-Warteschlangenmanager wird ausgeführt).

Das neu installierte Modul wird im Status gemeldet, ist aber nicht das Modul, das tatsächlich ausgeführt wird.

- Geben Sie den folgenden Befehl ein, um den Status der drei Knoten in der HA-Gruppe anzuzeigen:

  ```
  rdqmstatus -n
  ```

  Der Online-oder Offlinestatus der einzelnen Knoten wird gemeldet. For example:

  ```
  Node mqha04(mqhavm04.example.com) is online
  Node mqha05(mqhavm05.example.com) is offline
  Node mqha06(mqhavm06.example.com) is online
  ```

- Geben Sie folgenden Befehl ein, um den Status eines bestimmten Warteschlangenmanagers auf allen Knoten in der HA-Gruppe anzuzeigen:

  ```
  rdqmstatus -m qmname
  ```

  Dabei steht *qmname* für den Namen des RDQM, dessen Status angezeigt werden soll. Der Status des RDQM-Knotens auf dem aktuellen Knoten wird angezeigt, gefolgt von einer Zusammenfassung des Status der anderen beiden Knoten aus der Perspektive des aktuellen Knotens.

- V 9.3.0

  Geben Sie folgenden Befehl ein, um den Status eines bestimmten Warteschlangenmanagers auf allen Knoten in der HA-Gruppe einschließlich der Einzelheiten zu fehlgeschlagenen Ressourcenaktionen anzuzeigen:

  ```
  rdqmstatus -m qmname -a
  ```

  Dabei steht *qmname* für den Namen des RDQM, dessen Status angezeigt werden soll. Der Status des RDQM-Knotens auf dem aktuellen Knoten wird angezeigt, gefolgt von einer Zusammenfassung des Status der anderen beiden Knoten aus der Perspektive des aktuellen Knotens. Anschließend werden die Einzelheiten zu fehlgeschlagenen Ressourcenaktionen angezeigt, die dem RDQM zugeordnet sind.

- In der folgenden Tabelle sind die Informationen zum aktuellen Knoten zusammengefasst, die vom `rdqmstatus -m qmname`-Befehl für einen RDQM zurückgegeben werden können.

*Tabelle 32. Aktueller Knotenstatus*

| **Statusattribut** | **Mögliche Werte** | **Wird wann angezeigt** |
|---|---|---|
| Knotenname | *Knotenname* | Immer |
| Status des Warteschlangenmanagers | Aktiv<br>Wird an anderer Stelle ausgeführt<br>Beendet<br>Nicht verfügbar | Immer |
| CPU | *n.nn*% | Wird nur angezeigt, wenn der aktuelle Knoten über eine primäre Rolle verfügt (d. a. der RDQM wird auf diesem Knoten ausgeführt) |
| Hauptspeicher | *nnn* MB verwendet, *y.y* GB zugeordnet | Wird nur angezeigt, wenn der aktuelle Knoten über eine primäre Rolle verfügt (d. a. der RDQM wird auf diesem Knoten ausgeführt) |
| Queue manager file system | *nnn*MB used, *y.y*GB allocated [*z*%] | Wird nur angezeigt, wenn der aktuelle Knoten über eine primäre Rolle verfügt (d. a. der RDQM wird auf diesem Knoten ausgeführt) |

| *Tabelle 32. Aktueller Knotenstatus (Forts.)* | | |
|---|---|---|
| **Statusattribut** | **Mögliche Werte** | **Wird wann angezeigt** |
| HA-Rolle | Primär sekundär unbekannt | Immer |
| HA status | Alle Knoten im Standby-Modus<br>This node in standby<br>Remote nodes in standby<br>Gemischt<br><br>*status of remote nodes* | Alle Knoten im Standby-Modus<br>Aktueller Knoten im Standby-Modus<br>Beide fernen Knoten im Standby-Modus<br>Anderer Status für jeden fernen Knoten (siehe nächste Tabelle für den einzelnen Status)<br><br>Derselbe Status für beide fernen Knoten (siehe nächste Tabelle für alle Werte) |
| HA control | Enabled<br>Inaktiviert<br>Unbekannt | Immer. Zeigt an, ob RDQM von Pacemaker gesteuert wird. |
| HA preferred location | --<br>This node<br>Unbekannt<br>*Knotenname* | Immer |
| V 9.3.0 V 9.3.0 HA blocked location | None - Die Ausführung des Warteschlangenmanagers auf einem anderen Knoten ist nicht blockiert.<br>This node - Die Ausführung des Warteschlangenmanagers auf dem aktuellen Knoten ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen blockiert.<br>*Knotenname* - Die Ausführung des Warteschlangenmanagers auf *Knotenname* ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen blockiert.<br>*Knotenname1*, *Knotenname2* - Die Ausführung des Warteschlangenmanagers auf *Knotenname1* und *Knotenname2* ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen blockiert.<br>All nodes - Die Ausführung des Warteschlangenmanagers ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen auf allen Knoten blockiert. | Immer |
| HA floating IP interface | *Schnittstellenname* | Immer |
| HA floating IP address | *IPV4_address* | Immer |

In der folgenden Tabelle sind die Informationen zusammengefasst, die vom Befehl `rdqmstatus -m` *qmname* für die anderen Knoten in der HA-Gruppe zurückgegeben werden.

*Tabelle 33. Status des anderen Knotens*

| Statusattribut | Mögliche Werte | Wird wann angezeigt |
|---|---|---|
| Knotenname | *nodename* | Immer |
| HA status | Normal<br>Synchronization in progress<br>Fern nicht verfügbar<br>Inkonsistent<br>Angehalten<br>Ferner Knoten in Bereitschaft<br>Unbekannt | Knoten sind miteinander synchronisiert.<br>Synchronisieren mit fernem Knoten<br>Kommunikation mit fernem Knoten nicht möglich<br>Nicht synchron zum fernen Knoten und nicht Synchronisieren<br>Replikation angehalten<br>Ferner Knoten in Bereitschaft |
| HA-Synchronisation in Bearbeitung | *N.N*% | Wird angezeigt, wenn die Synchronisation in Bearbeitung ist und der Befehl als `root` ausgeführt wird |
| HA-geschätzte Synchronisationszeit | *yyyy-mm-dd hh:mm:ss.nnn* | Wird angezeigt, wenn die Synchronisation in Bearbeitung ist. |
| HA aus Synchronisationsdaten | *n* KB | Wird angezeigt, wenn der ferne Knoten nicht verfügbar oder inkonsistent |
| V 9.3.0 V 9.3.0 HA last in sync | *yyyy-mm-dd hh:mm:ss.nnn* | Wird angezeigt, wenn die HJA-Daten nicht synchron sind (nach der Erstsynchronisation). Gibt Zeit und Datum an, als die Daten zuletzt synchron waren. |

**Beispiel**

Beispiel für einen normalen Status auf dem Primärknoten:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                    Running
CPU:                                     0.00
Memory:                                  123MB
Queue manager file system:               606MB used, 1.0GB allocated [60%]
HA role:                                 Primary
HA status:                               Normal
HA control:                              Enabled
HA current location:                     This node
HA preferred location:                   This node
HA preferred location:                   This node
HA blocked location:                     None
HA floating IP interface:                eth4
HA floating IP address:                  192.0.2.4


Node:                                    mqhavm08.exampleco.com
HA status:                               Normal


Node:                                    mqhavm09.exampleco.com
HA status:                               Normal
```

Beispiel für einen normalen Status auf einem Sekundärknoten:

```
Node:                                   mqhavm08.exampleco.com
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    mqhavm07.exampleco.com
HA preferred location:                  mqhavm07.exampleco.com
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm07.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Normal
```

Beispiel für den Status auf dem Primärknoten, wenn die Synchronisation in Bearbeitung ist:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Synchronization in progress
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Synchronization in progress
HA synchronization progress:            11.0%
HA estimated time to completion:        2017-09-06 14:55:05

Node:                                   mqhavm09.exampleco.com
HA status:                              Synchronization in progress
HA synchronization progress:            11.0%
HA estimated time to completion:        2017-09-06 14:55:06
```

V 9.3.0 Beispiel für den Status auf dem Primärknoten, wenn die Synchronisation nicht mehr vorhanden ist:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Mixed
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Inconsistent
HA out of sync data:                    15932KB
HA last in sync:                        2017-09-06 14:55:06
```

Beispiel für einen Primärknoten mit mehreren Status:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.02
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
```

```
HA status:                              Mixed
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Inconsistent
```

V 9.3.0 Beispiel für einen Primärknoten mit fehlgeschlagenen Ressourcenaktionen:

```
Node:                                   mqhavm07.exampleco.com
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              606MB used, 1.0GB allocated [60%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  mqhavm08.exampleco.com
HA blocked location:                    mqhavm08.exampleco.com
HA floating IP interface:               eth4
HA floating IP address:                 192.0.2.4

Node:                                   mqhavm08.exampleco.com
HA status:                              Normal

Node:                                   mqhavm09.exampleco.com
HA status:                              Normal

Failed resource action:                 Start
Resource type:                          Filesystem
Failure node:                           mqhavm08.exampleco.com
Failure time:                           2017-09-06 12:00:00
Failure reason:                         Couldn't find directory  [/var/mqm/vols/qmname] to use
as a mount point
Blocked location:                       mqhavm08.exampleco.com
```

Dieser Status zeigt an, dass Pacemaker das Dateisystem auf Knoten 'mqhavm08.exampleco.com' um 12:00:00 nicht starten konnte. Diese fehlgeschlagene Ressourcenaktion zeigt an, dass die Ausführung des Warteschlangenmanagers auf 'mqhavm08.exampleco.com' blockiert wird. Wenn das zugrunde liegende Problem behoben wurde, durch das das Fehlschlagen der Ressourcenaktion verursacht wurde, führen Sie den Befehl **rdqmclean** aus, um die fehlgeschlagene Aktion zu löschen, damit Pacemaker die Aktion erneut ausführen kann (falls erforderlich).

V 9.3.0 Beispiel für einen Zusammenfassungsstatus, der eine Diskrepanz zwischen der Betriebssystem-Kernelversion (RHEL 7.9) und dem DRBD-Kernelmodul (für RHEL 7.8) zeigt. Obwohl der Status meldet, dass DRBD-Kernelmodul geladen ist und der Warteschlangenmanager aktiv ist, sollten Sie das DRBD-Kernelmodul mit der Version aktualisieren, die für den aktiven Betriebssystemkern in dieser Situation vorgesehen ist.

```
Node:                                   mqhavm07.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     RDQM7
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
```

V 9.3.0 Beispiel für einen Zusammenfassungsstatus, der eine Diskrepanz zwischen der Betriebssystem-Kernelversion (RHEL 7.9) und dem DRBD-Kernelmodul (für RHEL 7.6) zeigt. In diesem Beispiel ist

die Versionsabweichung schwerwiegender, und das DRBD-Kernelmodul kann nicht erfolgreich geladen werden. Daher schlägt der Start des Warteschlangenmanagers auf seinem bevorzugten Knoten und seinem HA-Status in Unknown fehl. Um diesen Fehler zu beheben, muss das DRBD-Kernelmodul mit dem Versionsziel für den aktiven Betriebssystemkern aktualisiert werden.

```
Node:                                mqhavm57.exampleco.com
OS kernel version:                   3.10.0-1160.15.2
DRBD OS kernel version:              3.10.0-957
DRBD version:                        9.1.2+ptf.3
DRBD kernel module status:           Partially loaded

Queue manager name:                  QM2
Queue manager status:                Running elsewhere
HA status:                           Unknown
HA current location:                 mqhavm58.exampleco.com
HA preferred location:               This node
HA blocked location:                 All nodes
```

**Zugehörige Verweise**

Linux rdqmstatus

## *IP-Adressen in Hochverfügbarkeitskonfigurationen ändern*

Wenn Sie die IP-Adressen einer der Schnittstellen in einer Hochverfügbarkeitskonfiguration ändern, sind keine Hochverfügbarkeitsvorgänge mehr verfügbar und der Warteschlangenmanager kann auf dem Knoten, dessen Adressen geändert wurden, nicht ausgeführt werden.

Sie geben bis zu drei IP-Adressen für die HA-Operation in der `rdqm.ini`-Datei an. Wenn Sie die Adressen für die Pacemaker-Überwachung bereits geändert haben, müssen Sie diese temporär auf die ursprünglichen Werte zurücksetzen, damit Sie den Vorgang ausführen können. Andernfalls kann der HA-RDQM-Warteschlangenmanager nicht gelöscht werden.

1. Entfernen Sie die HA-Konfiguration auf jedem Knoten. Sie entfernen eine HA, indem Sie die Warteschlangenmanager sichern und anschließend löschen (siehe „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen” auf Seite 723 und „HA-RDQM löschen” auf Seite 620) und daraufhin die HA-Gruppe selbst entfernen (siehe „Pacemaker-Cluster löschen (HA-Gruppe)” auf Seite 618).
2. Erstellen Sie die HA-Konfiguration erneut mit den neuen IP-Adressen (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)” auf Seite 615).
3. Erstellen Sie die HA-Warteschlangenmanager erneut und stellen Sie die Sicherungskopie wieder her (siehe „HA-RDQM erstellen” auf Seite 619 und „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen” auf Seite 723).

## Linux *Fehlgeschlagenen Knoten in einer Hochverfügbarkeitskonfiguration ersetzen*

Wenn einer der Knoten in Ihrer HA-Gruppe fehlschlägt, können Sie ihn ersetzen.

### Informationen zu diesem Vorgang

Welche Schritte zum Ersetzen eines Knotens erforderlich sind, hängt von dem Szenario ab:

- Wenn Sie den fehlerhaften Knoten durch einen Knoten durch eine identische Konfiguration ersetzen, können Sie den Knoten ersetzen, ohne die HA-Gruppe zu unterbrechen.
- Wenn der neue Knoten über eine andere Konfiguration verfügt, müssen Sie die HA-Gruppe löschen und anschließend erneut erstellen. Sie können zuerst die Warteschlangenmanager auf dem Knoten sichern, auf dem sie ausgeführt werden, und sie anschließend nach dem erneuten Erstellen der HA-Gruppe wiederherstellen.

### Prozedur

- Wenn der Ersatzknoten so konfiguriert ist, dass er wie der fehlgeschlagene Knoten aussieht (derselbe Hostname, dieselben IP-Adressen usw.), führen Sie die folgenden Schritte auf dem neuen Knoten aus:

a) Erstellen Sie eine `rdqm.ini`-Datei, die mit den Dateien auf den anderen Knoten übereinstimmt, und führen Sie dann den Befehl `rdqmadm -c` aus (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)" auf Seite 615).

b) Führen Sie den Befehl `crtmqm -sxs` *qmanager* aus, der jeden Warteschlangenmanager mit replizierten Daten neu erstellt (siehe „HA-RDQM erstellen" auf Seite 619).

- Wenn der Ersatzknoten über eine andere Konfiguration für den fehlgeschlagenen Knoten verfügt:

a) Sichern Sie bei Bedarf Ihre Warteschlangenmanager (siehe „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen" auf Seite 723).

b) Löschen Sie die replizierten Datenwarteschlangenmanager von den anderen Knoten in der HA-Gruppe, indem Sie den Befehl **dltmqm** verwenden (siehe „HA-RDQM löschen" auf Seite 620).

c) Dekonfigurieren Sie den Pacemaker -Cluster mit dem Befehl **rdqmadm -u** (siehe „Pacemaker-Cluster löschen (HA-Gruppe)" auf Seite 618).

d) Rekonfigurieren Sie den Pacemaker-Cluster, einschließlich der Informationen für den neuen Knoten, mit dem Befehl **rdqmadm -c** (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)" auf Seite 615).

e) Falls erforderlich (d. h., wenn Sie keinen SSH-Zugang zu den anderen Knoten haben) führen Sie den Befehl `crtmqm -sxs` *qmanager* aus, um jeden replizierten Datenwarteschlangenmanager auf den anderen Knoten erneut zu erstellen (siehe „HA-RDQM erstellen" auf Seite 619).

f) Führen Sie den Befehl `crtmqm -sx` *qmanager* aus, um die Warteschlangenmanager auf dem Ersatzknoten zu erstellen.

g) Stellen Sie bei Bedarf die Daten und die Konfiguration für Ihre Warteschlangenmanager wieder her (siehe „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen" auf Seite 723).

## Linux MQ Adv. RDQM-Notfallwiederherstellung

RDQM (Replicated Data Queue Manager = Warteschlangenmanager mit replizierten Daten) ist auf einer Untergruppe von Linux-Plattformen verfügbar und kann eine Disaster-Recovery-Lösung bereitstellen.

Ausführliche Informationen finden Sie unter Software Product Compatibility Reports .

Sie können eine primäre Instanz eines auf einem Server ausgeführten Disaster Recovery-Warteschlangenmanagers und eine sekundäre Instanz des Warteschlangenmanagers auf einem anderen Server, der als Recovery-Knoten fungiert, erstellen. Daten werden zwischen den Warteschlangenmanagerinstanzen repliziert. Wenn Sie den primären Warteschlangenmanager verlieren, können Sie die sekundäre Instanz manuell in die primäre Instanz aufnehmen und den Warteschlangenmanager starten. Anschließend können Sie die Arbeit am gleichen Ort wieder aufnehmen. Sie können einen Warteschlangenmanager nicht starten, solange er sich in der sekundären Rolle befindet. Die Replikation der Daten zwischen den beiden Knoten wird von DRBD verarbeitet.

Sie können zwischen der synchronen und asynchronen Replikation von Daten zwischen primären und sekundären Warteschlangenmanagern wählen. Wenn Sie die asynchrone Option auswählen, werden Operationen wie IBM MQ PUT oder GET abgeschlossen und zur Anwendung zurückgegeben, bevor das Ereignis in den sekundären Warteschlangenmanager repliziert wird. Asynchrone Replikation bedeutet, dass nach einer Recovery-Situation möglicherweise einige Messaging-Daten verloren gehen. Der sekundäre WS-Manager befindet sich jedoch in einem konsistenten Status und kann sofort gestartet werden, selbst wenn er zu einem etwas früheren Teil des Nachrichtenstroms gestartet wird.

Sie können einem vorhandenen Warteschlangenmanager keine Disaster-Recovery hinzufügen; es ist jedoch möglich, einen vorhandenen Warteschlangenmanager auf einen RDQM-Warteschlangenmanager umzumigrieren (siehe „Warteschlangenmanager als DR-RDQM-Warteschlangenmanager migrieren" auf Seite 648).

Es können mehrere Paare von RDQM-Warteschlangenmanagern auf einer Reihe unterschiedlicher Server ausgeführt werden. Sie können z. B. primäre Disaster Recovery-Warteschlangenmanager haben, die auf verschiedenen Knoten ausgeführt werden, während deren sekundäre Disaster-Recovery-Warteschlangenmanager auf ein und demselben Knoten ausgeführt werden. Einige Beispielkonfigurationen werden in den folgenden Diagrammen dargestellt.

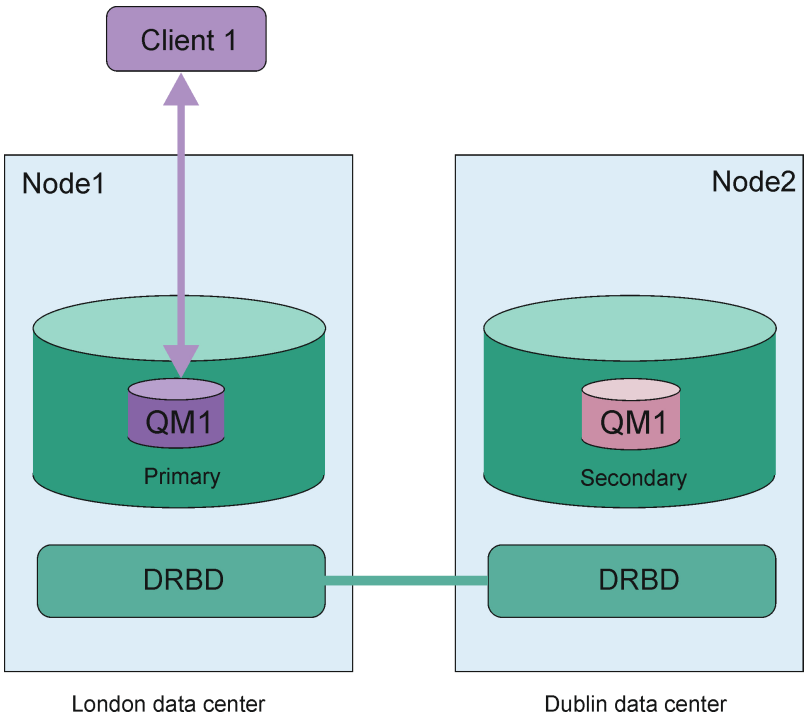



*Abbildung 81. Einzelnes RDQM-Paar*

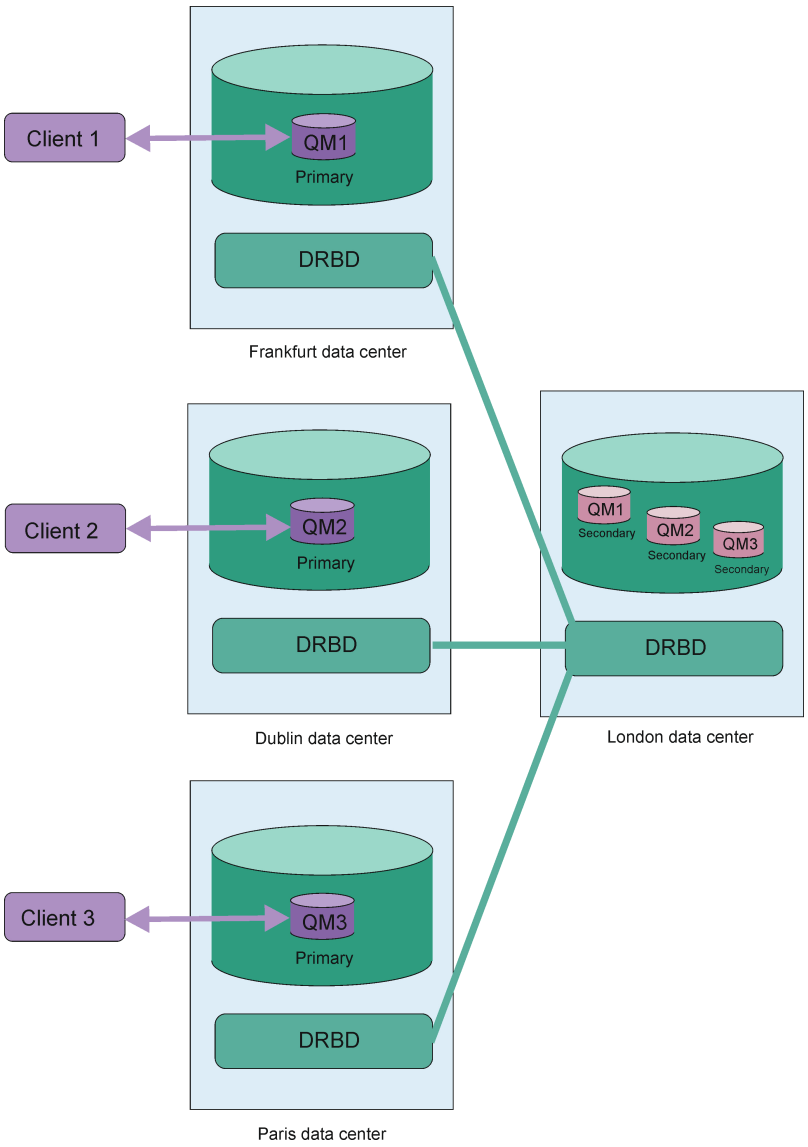



*Abbildung 82. Sekundäre Warteschlangenmanager in demselben Knoten*

## Replikation, Synchronisation und Momentaufnahmen

Während die beiden Knoten in einer Konfiguration zur Wiederherstellung nach einem Katastrophenfall verbunden sind, werden alle Aktualisierungen der persistenten Daten für einen Disaster Recovery Queue Manager von der primären Instanz des Warteschlangenmanagers an die sekundäre Instanz übertragen. Dies wird als **Replikation** bezeichnet.

Wenn die Netzverbindung zwischen den beiden Knoten verloren geht, werden die Änderungen an den persistenten Daten für die primäre Instanz eines Warteschlangenmanagers protokolliert. Wenn die Netzverbindung wiederhergestellt wird, wird ein anderer Prozess verwendet, um die sekundäre Instanz so schnell wie möglich auf die Geschwindigkeit zu bringen. Dies wird als **Synchronisation** bezeichnet.

Während die Synchronisation in Bearbeitung ist, befindet sich die Daten auf der sekundären Instanz in einem inkonsistenten Status. Es wird eine **Momentaufnahme** des Status der Daten des sekundären Warteschlangenmanagers erstellt. Wenn während der Synchronisation ein Fehler des Hauptknotens oder der Netzverbindung auftritt, wird die sekundäre Instanz auf diese Momentaufnahme zurückgesetzt und der WS-Manager kann gestartet werden. Alle Aktualisierungen, die seit dem ursprünglichen Netzausfall aufgetreten sind, gehen jedoch verloren.

## Partitionierte Daten (Aufspaltung)

Nach dem Verlust der primären Instanz eines Warteschlangenmanagers ist bei DR-RDQM-Konfigurationen Benutzereingriff erforderlich, um die sekundäre Instanz auf dem Wiederherstellungsknoten hochzustufen und auszuführen. Es liegt in der Verantwortung desjenigen, der die sekundäre Instanz hochstuft, sicherzustellen, dass der vorherige primäre Warteschlangenmanager gestoppt wird. Wenn die ursprüngliche primäre Instanz weiterläuft, könnte sie Nachrichten verarbeiten und beim Wiederherstellen des Normalbetriebs hätten die beiden Instanzen des Warteschlangenmanagers unterschiedliche Anzeigen der Daten. Dies wird als partitionierter oder Split-Brain-Status bezeichnet.

Stellen Sie sich die folgenden Situationen vor:

- Der Knoten, auf dem der primäre Warteschlangenmanager ausgeführt wird, fällt vollständig aus. Sie stufen die sekundäre Instanz hoch, so dass sie zur primären Instanz wird. Sie können keine Aktion ausführen, um die ursprüngliche Primärinstanz zu stoppen, da sie nicht aktiv ist. Wenn der ursprüngliche Knoten repariert oder ersetzt wird, wird der Warteschlangenmanager auf diesem Knoten zunächst zum sekundären Warteschlangenmanager gemacht und mit dem primären Warteschlangenmanager auf dem Wiederherstellungsknoten synchronisiert. Anschließend werden die Rollen der beiden Warteschlangenmanager umgekehrt und der normale Betrieb wird wieder aufgenommen. Der einzige mögliche Datenverlust in dieser Situation entsteht für alle Daten, die die primäre Instanz nicht vollständig repliziert hatte, bevor der Knoten ausfiel.
- Es gibt einen Netzausfall, der die Replikationsverbindung zwischen den Knoten betrifft, auf denen die primäre und die sekundäre Instanz des Warteschlangenmanagers ausgeführt werden. In dieser Situation müssen Sie sicherstellen, dass Sie die ursprüngliche primäre Instanz stoppen, bevor Sie die sekundäre hochstufen. Wenn die ursprüngliche Primärinstanz noch über andere Netzkonnektivität verfügt, haben Sie effektiv zwei primäre Instanzen, die gleichzeitig ausgeführt werden, und es können sich partitionierte Daten ansammeln. (Wenn die Replikationsverbindung funktioniert, können Sie keinen sekundären Warteschlangenmanager hochstufen, falls die primäre Instanz noch aktiv ist; der Befehl schlägt fehl.)
- Auf dem Knoten, auf dem die primäre Instanz des Warteschlangenmanagers ausgeführt wird, gibt es einen vollständigen Netzausfall. Wiederum müssen Sie sicherstellen, dass Sie die primäre Instanz stoppen, bevor Sie die sekundäre Instanz hochstufen. Wenn die vorherige Primärinstanz noch aktiv ist, wenn das Netz wiederhergestellt wird, gibt es zwei primäre Instanzen, und es entstehen erneut partitionierte Daten.

Bei einem verwalteten Failover sollte für die Warteschlangenmanagerinstanzen nicht der DR-Status `partitioned` angezeigt werden. Bei einer gesteuerten Funktionsübernahme wird der Warteschlangenmanager auf dem primären Knoten gestoppt, anschließend wird nach der vollständigen Replizierung der Daten der Warteschlangenmanager auf dem Wiederherstellungsknoten gestartet. Ein partitionierter Status ist nicht zu erwarten, da der Warteschlangenmanager beendet wird und die Daten zwischen den Knoten synchronisiert werden, bevor er auf dem Wiederherstellungsknoten gestartet wird. Wenn der Warteschlangenmanager auf dem Wiederherstellungsknoten gestartet wird, während ein Konnektivitätsverlust zwischen den Knoten besteht, ist eine Datendivergenz wahrscheinlich, wenn der Warteschlangenmanager auf dem Hauptknoten aktiv war, als die Konnektivität verloren ging. In diesem Szenario ist zu erwarten, dass ein partitionierter Status gemeldet wird, sobald die Konnektivität wiederhergestellt ist, da die Warteschlangenmanagerdaten nicht synchronisiert wurden. Wenn ein partitionierter Status auftritt, müssen Sie möglicherweise die beiden Datensätze untersuchen und anhand der Ergebnisse entscheiden, welcher beibehalten werden soll. Siehe „Problem mit Partitionierung (Spaltung) bei DR-RDQM beheben" auf Seite 665.

## Linux *Voraussetzungen für RDQM-DR-Lösung*

Sie müssen eine Reihe von Anforderungen erfüllen, bevor Sie ein RDQM-WS-Manager-Paar (DR-RDQM Disaster Recovery) konfigurieren können.

### Systemvoraussetzungen

Bevor Sie RDQM DR konfigurieren, müssen Sie auf jedem der Server, auf denen RDQM DR-Warteschlangenmanager ausgeführt werden sollen, eine Konfiguration ausführen.

- Für jeden Knoten ist eine Datenträgergruppe mit dem Namen `drbdpool` erforderlich. Der Speicher für jeden replizierten Datenwarteschlangenmanager für Notfallwiederherstellung (DR RDQM) wird als zwei separate logische Datenträger pro WS-Manager aus dieser Datenträgergruppe zugeordnet. (Jeder WS-Manager benötigt zwei logische Datenträger, um die Zurücksetzung auf die Momentaufnahmeoperation zu unterstützen, sodass jeder DR RDQM nur mehr als dem doppelten Speicher zugeordnet wird, den Sie beim Erstellen angeben.) Um die beste Leistung zu erhalten, sollte diese Datenträgergruppe aus einem oder mehreren physischen Datenträgern bestehen, die internen Plattenlaufwerken entsprechen (vorzugsweise SSDs).
- Nachdem Sie die `drbdpool`-Datenträgergruppe erstellt haben, können Sie keine weiteren Aktionen ausführen. IBM MQ verwaltet die logischen Datenträger, die in `drbdpool` erstellt wurden, und wie und wo sie bereitgestellt werden.
- Für jeden Knoten ist eine Schnittstelle erforderlich, die für die Datenreplikation verwendet wird. Dies sollte eine ausreichende Bandbreite haben, um die Replikationsanforderungen bei der erwarteten Auslastung aller replizierten Datenwarteschlangenmanager zu unterstützen.

  Für die maximale Fehlertoleranz sollte diese Schnittstelle eine unabhängige Netzschnittstellenkarte (NICs) sein.
- DRBD erfordert, dass jeder Knoten, der für RDQM verwendet wird, einen gültigen Internet-Host-Namen hat (der von `uname -n` zurückgegebene Wert), wie in RFC 952 (geändert durch RFC 1123) definiert.
- Wenn zwischen den Knoten, die für DR RDQM verwendet werden, eine Firewall vorhanden ist, muss die Firewall den Datenverkehr zwischen den Knoten in den Ports zulassen, die für die Replikation verwendet werden. Es wird ein Beispielscript bereitgestellt, `/opt/mqm/samp/rdqm/firewalld/configure.sh`, das die erforderlichen Ports öffnet, wenn Sie die Standardfirewall unter RHEL ausführen. Sie müssen das Script als `root` ausführen. Wenn Sie eine andere Firewall verwenden, überprüfen Sie die Servicedefinitionen `/usr/lib/firewalld/services/rdqm*` , um festzustellen, welche Ports geöffnet werden müssen. Das Script fügt die folgenden permanenten firewallD-Serviceregeln für DRBD und IBM MQ hinzu (Sie können das Script so bearbeiten, dass Pacemaker-Ports übergangen werden, wenn Sie keine HA verwenden):
  - MQ-INSTALLATIONSPFAD/samp/rdqm/firewalld/services/rdqm-drbd.xml lässt TCP-Ports 7000-7100 zu.
  - MQ-INSTALLATIONSPFAD/samp/rdqm/firewalld/services/rdqm-mq.xml lässt TCP-Port 1414 zu (Sie müssen das Script bearbeiten, wenn Sie einen anderen Port verwenden)
- Verwendet das System SELinux in einem anderen Modus als permissive, müssen Sie den folgenden Befehl ausführen:

```
semanage permissive -a drbd_t
```

## Netzvoraussetzungen

Es wird empfohlen, die Knoten zu lokalisieren, die für die Wiederherstellung nach einem Katastrophenfall in verschiedenen Rechenzentren verwendet werden.

Sie sollten sich die folgenden Einschränkungen bewusst sein:

- Die Leistung verschlechtert sich schnell mit zunehmender Latenzzeit zwischen Rechenzentren. IBM unterstützt eine Latenzzeit von bis zu 5 ms für die synchrone Replikation und 100 ms für die asynchrone Replikation.
- Die über die Replikationsverbindung gesendeten Daten unterliegen keiner zusätzlichen Verschlüsselung, die über die evtl. durch die Verwendung von IBM MQ AMS vorgegebene Verschlüsselung hinausgeht.
- Die Konfiguration eines RDQM-Warteschlangenmanagers für die Wiederherstellung nach einem Katastrophenfall verursacht einen Systemaufwand aufgrund der Anforderung, Daten zwischen den beiden RDQM-Knoten zu replizieren. Die synchrone Replikation hat einen höheren Systemaufwand als die asynchrone Replikation. Wenn die synchrone Replikation verwendet wird, werden Platten-E/A-Operationen blockiert, bis die Daten auf beide Knoten geschrieben werden. Wenn die asynchrone Replikation ver-

wendet wird, müssen die Daten nur auf den Primärknoten geschrieben werden, bevor die Verarbeitung fortgesetzt werden kann.

## Benutzeranforderungen für die Arbeit mit Warteschlangenmanagern

Um replizierte Datenwarteschlangenmanager (RDQMs) zu erstellen, zu löschen oder zu konfigurieren, müssen Sie entweder der Rootbenutzer sein oder eine Benutzer-ID haben, die zur Gruppe `mqm` gehört, die die Berechtigung 'sudo' für die folgenden Befehle erteilt:

- **`crtmqm`**
- **`dltmqm`**
- **`rdqmdr`**

Ein Benutzer, der zu der Gruppe `mqm` gehört, kann den Status und den Status eines DR RDQM-Systems mit den folgenden Befehlen anzeigen:

- **`dspmq`**
- **`rdqmstatus`**

Der Benutzer `mqm` muss auf beiden Servern dieselbe UID haben und die Gruppe `mqm` muss auf beiden Servern dieselbe GID haben.

# Linux *RDQM für die Wiederherstellung nach einem Katastrophenfall erstellen*

Mit dem Befehl **`crtmqm`** können Sie einen replizierten Datenwarteschlangenmanager (RDQM) erstellen, der als primärer oder sekundärer Warteschlangenmanager in einer Disaster-Recovery-Konfiguration fungiert.

## Informationen zu diesem Vorgang

Sie können einen replizierten Datenwarteschlangenmanager (RDQM) als Benutzer in der `mqm` -Gruppe erstellen, wenn der Benutzer sudo verwenden kann. Andernfalls müssen Sie den RDQM als Root erstellen.

Sie müssen einen primären RDQM DR-Warteschlangenmanager auf einem Knoten erstellen. Anschließend müssen Sie eine sekundäre Instanz desselben Warteschlangenmanagers auf einem anderen Knoten erstellen. Die primären und sekundären Instanzen müssen denselben Namen haben und die gleiche Menge an Speicher zugeordnet werden.

Die folgenden Punkte enthalten Anleitungen zur Dimensionierung des Dateisystems des Warteschlangenmanagers:

1. Wenn Sie einen RDQM-Warteschlangenmanager erstellen, wird ein Dateisystem zum Speichern von Warteschlangenmanagerdaten und -protokollen zugeordnet. Es ist wichtig, die Größe dieses Dateisystems entsprechend zu ändern, damit der Warteschlangenmanager fortlaufende Aktivitäten in seinen Protokollen aufzeichnen und Anwendungsnachrichten in Warteschlangen speichern kann. Berücksichtigen Sie bei der Dimensionierung des Dateisystems die Anforderungen für Spitzennachrichten, das künftige Workloadwachstum und Anwendungsausfälle, die dazu führen können, dass Nachrichten in Warteschlangen erstellt werden. Informationen zur Berechnung der Größe des Wiederherstellungsprotokolls des Warteschlangenmanagers finden Sie unter „Wie groß sollte ich mein Protokolldateisystem machen?" auf Seite 703. Bei der Berechnung des Speicherbedarfs für Anwendungsnachrichten müssen die Größe und Anzahl der Nachrichten sowie deren MQMD-Header und Nachrichteneigenschaften berücksichtigt werden.
2. Die Größe von Dateisystemen des RDQM-Warteschlangenmanagers kann nicht dynamisch geändert werden. Wenn dies erforderlich ist, müssen Sie einen RDQM-Warteschlangenmanager mit einem größeren Dateisystem sichern und wiederherstellen (siehe „Größe des Dateisystems für einen HA-RDQM-Warteschlangenmanager ändern" auf Seite 625).
3. Sie können die Größe einzelner Warteschlangen auf Platte begrenzen, indem Sie lokale Warteschlangenattribute wie MAXDEPTH und MAXFSIZE verwenden. Siehe Warteschlangendateien von IBM MQ ändern.

4. Sie sollten Ihre laufende Plattenbelegung überwachen und entsprechend reagieren, wenn die Plattenbelegung zunimmt, bevor die Dateisystembelegung kritisch wird. Die Dateisystemnutzung kann mithilfe von Plattform-/Betriebssystemfunktionen überwacht werden oder indem Metriken abonniert werden, die in IBM MQ -Systemthemen veröffentlicht werden, die unter In den Systemthemen veröffentlichte Metrikenbeschrieben sind.

## Prozedur

- Gehen Sie wie folgt vor, um einen primären DR RDQM zu
  a) Geben Sie den folgenden Befehl ein:

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

  Dabei gilt:

  **-rr p**
  Gibt an, dass Sie die primäre Instanz des Warteschlangenmanagers erstellen.

  **-rt a | s**
  **-rt s** gibt an, dass die DR-Konfiguration die synchrone Replikation verwendet, **-rt a** gibt an, dass die DR-Konfiguration die asynchrone Replikation verwendet. Die asynchrone Replikation ist die Standardeinstellung.

  **-rl *Local_IP***
  Gibt die lokale IP-Adresse an, die für die DR-Replikation dieses Warteschlangenmanagers verwendet werden soll.

  **-ri *Recovery_IP***
  Gibt die IP-Adresse der Schnittstelle an, die für die Replikation auf dem Server verwendet wird, auf dem sich die sekundäre Instanz des Warteschlangenmanagers befindet.

  **-rn *Recovery_Name***
  Gibt den Namen des Systems an, auf dem sich die sekundäre Instanz des Warteschlangenmanagers befindet. Der Name ist der Wert, der zurückgegeben wird, wenn Sie `uname -n` auf diesem Server ausführen. Sie müssen auf diesem Server explizit einen sekundären WS-Manager erstellen.

  **-rp *Port***
  Gibt den Port an, der für die DR-Replikation verwendet werden soll

  ***other_crtmqm_options***
  Sie können optional eine oder mehrere der folgenden allgemeinen **crtmqm** Optionen angeben:

  - -z
  - -q
  - -c *Text*
  - -d *DefaultTransmissionQueue*
  - -h *MaxHandles*
  - -g *ApplicationGroup*
  - -oa *user* | *group*
  - -t *TrigInt*
  - -u *DeadQ*
  - -x *MaxUMsgs*
  - -lp *LogPri*
  - -ls *LogSec*
  - -lc | -l
  - -lla | -lln

– -lf *LogFileSize*

– -p *Port*

**-fs *size***
(Optional) Gibt die Größe des Dateisystems an, das für den Warteschlangenmanager erstellt werden soll, d. B. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird. Ein weiteres logisches Volumen dieser Größe wird erstellt, um die Zurücksetzung auf die Momentaufnahmeoperation zu unterstützen, so dass der Gesamtspeicher für den DR RDQM etwas mehr als doppelt so groß ist wie hier angegeben.

*Größe* ist ein numerischer Wert, der in GB angegeben wird. Sie können einen Wert in MB angeben, indem Sie den Wert gefolgt vom Zeichen Meingeben. Geben Sie beispielsweise 3 ein, um eine Dateisystemgröße von 3 GB anzugeben. Geben Sie 1024Mein, um eine Dateisystemgröße von 1024 MB anzugeben. (Sie können auch ein G-Suffix hinzufügen, um explizit GB anzugeben.)

***QMNAME***
Gibt den Namen des replizierten Datenwarteschlangenmanagers an. Bei dem Namen muss die Groß-/Kleinschreibung beachtet werden

Nach Beendigung des Befehls wird der Befehl ausgegeben, den Sie auf dem Sekundärknoten eingeben müssen, um die sekundäre Instanz des Warteschlangenmanagers zu erstellen. Sie können auch den Befehl **rdqmdr** auf Ihrem Primärknoten verwenden, um den Befehl **crtmqm** abzurufen, den Sie auf dem Sekundärknoten ausführen müssen, um den sekundären Warteschlangenmanager zu erstellen (siehe „Primäre und sekundäre Merkmale von DR-RDQMs verwalten” auf Seite 655).

- Gehen Sie wie folgt vor, um einen sekundären DR RDQM zu erstellen:

  a) Geben Sie den folgenden Befehl auf dem Knoten ein, der sekundäre Instanzen von RDQM enthalten soll:

```
crtmqm  -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name  -rp Port
[other_crtmqm_options] [-fs size] QMname
```

  Dabei gilt Folgendes:

  **-rr s**
  Gibt an, dass Sie die sekundäre Instanz des Warteschlangenmanagers erstellen.

  **-rt a | s**
  **-rt s** gibt an, dass die DR-Konfiguration die synchrone Replikation verwendet, **-rt a** gibt an, dass die DR-Konfiguration die asynchrone Replikation verwendet.

  **-rl *Local_IP***
  Gibt die lokale IP-Adresse an, die für die DR-Replikation dieses Warteschlangenmanagers verwendet werden soll.

  **-ri *Primary_IP***
  Gibt die IP-Adresse der Schnittstelle an, die für die Replikation auf dem Server verwendet wird, auf dem sich die primäre Instanz des WS-Managers befindet.

  **-rn *Primary_Name***
  Gibt den Namen des Systems an, auf dem sich die primäre Instanz des Warteschlangenmanagers befindet. Der Name ist der Wert, der zurückgegeben wird, wenn Sie uname -n auf diesem Server ausführen.

  **-rp *Port***
  Gibt den Port an, der für die DR-Replikation verwendet werden soll

  ***other_crtmqm_options***
  Sie können optional eine oder mehrere der folgenden allgemeinen **crtmqm** Optionen angeben:

  – -z

  **-fs *size***
  Gibt die Größe des Dateisystems an, das für den WS-Manager erstellt werden soll, d. B. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird. Wenn Sie

bei der Erstellung des primären Warteschlangenmanagers eine andere als die Standardgröße angegeben haben, müssen Sie hier denselben Wert angeben.

*Größe* ist ein numerischer Wert, der in GB angegeben wird. Sie können einen Wert in MB angeben, indem Sie den Wert gefolgt vom Zeichen Meingeben. Geben Sie beispielsweise 3 ein, um eine Dateisystemgröße von 3 GB anzugeben. Geben Sie 1024Mein, um eine Dateisystemgröße von 1024 MB anzugeben. (Sie können auch ein G-Suffix hinzufügen, um explizit GB anzugeben.)

***QMNAME***
Gibt den Namen des replizierten Datenwarteschlangenmanagers an. Dieser Name muss mit dem Namen identisch sein, den Sie für die primäre Instanz des Warteschlangenmanagers angegeben haben. Beachten Sie, dass bei dem Namen die Groß-/Kleinschreibung beachtet werden

## Nächste Schritte

Nachdem Sie die primären und sekundären Instanzen Ihres Warteschlangenmanagers erstellt haben, müssen Sie den Status auf beiden Knoten überprüfen, um beide zu überprüfen. Verwenden Sie den Befehl **rdqmstatus** auf beiden Knoten. Die Knoten sollten den normalen Status wie in „DR-RDQM-Status anzeigen" auf Seite 657 beschrieben anzeigen. Wenn sie diesen Status nicht anzeigen, löschen Sie die sekundäre Instanz, und erstellen Sie sie erneut, und achten Sie darauf, die richtigen Argumente zu verwenden.

**Zugehörige Verweise**

crtmqm

Linux *DR RDQM löschen*

Sie verwenden den Befehl **dltmqm** , um einen Disaster-Recovery-Warteschlangenmanager für replizierte Daten (RDQM) zu löschen.

## Informationen zu diesem Vorgang

Sie müssen den Befehl ausführen, um RDQM auf den primären und sekundären RDQM-Knoten des RDQM zu löschen. RDQM muss zuerst beendet werden. Sie können den Befehl als mqm-Benutzer ausführen, wenn dieser Benutzer über die erforderlichen Zugriffsrechte für "sudo" verfügt. Andernfalls müssen Sie den Befehl als Root ausführen.

## Prozedur

- Geben Sie den folgenden Befehl ein, um einen DR RDQM zu löschen:

```
dltmqm RDQM_name
```

**Zugehörige Verweise**

dltmqm

Linux MQ Adv. *Warteschlangenmanager als DR-RDQM-Warteschlangenmanager migrieren*

Sie können einen vorhandenen Warteschlangen migrieren, um einen replizierten Datenwarteschlangenmanager (RDQM) zur Wiederherstellung nach einem Katastrophenfall (DR) zu erhalten, indem Sie die zugehörigen persistenten Daten sichern und diese Daten anschließend in einem neu erstellten RDQM-Warteschlangenmanager mit dem gleichen Namen wiederherstellen.

## Informationen zu diesem Vorgang

Für replizierte Datenwarteschlangenmanager zur Wiederherstellung nach einem Katastrophenfall ist ein dedizierter logischer Datenträger (Dateisystem) und die Konfiguration der Plattenreplikation und HA-Steuerung erforderlich. Diese Komponenten werden nur konfiguriert, wenn ein neuer Warteschlangenmanager erstellt wird. Ein vorhandener Warteschlangenmanager kann migriert werden, damit ein RDQM verwendet wird, indem die zugehörigen persistenten Daten gesichert und anschließend in einem neu erstellten RDQM-Warteschlangenmanager mit dem gleichen Namen wiederhergestellt werden. Durch diese Vorge-

hensweise bleiben die Konfiguration, der Status und die persistenten Nachrichten des Warteschlangenmanagers so erhalten, wie sie zum Zeitpunkt der Sicherungserstellung waren.

**Anmerkung:** Sie können einen Warteschlangenmanager nur aus einer Version von IBM MQ migrieren, die identisch oder niedriger ist als die Version, auf der RDQM installiert ist. Das Betriebssystem und die Architektur müssen ebenfalls identisch sein. Andernfalls müssen sie einen neuen Warteschlangenmanager auf Ihrer Zielplattform erstellen; siehe Warteschlangenmanager in ein anderes Betriebssystem verschieben.

Die folgenden Bedingungen sollten vor der Migration eines Warteschlangenmanagers erfüllt sein:

- Bewerten Sie Ihre Anforderungen für die Wiederherstellung nach einem Katastrophenfall und lesen Sie den Abschnitt „RDQM-Notfallwiederherstellung” auf Seite 640.
- Überprüfen Sie die Anwendungen und Warteschlangenmanager, die eine Verbindung zum Warteschlangenmanager herstellen. Beachten Sie dabei die Änderungen, die für die Weiterleitung der Verbindungen an den RDQM-Knoten erforderlich sind, auf dem der Warteschlangenmanager ausgeführt wird.
- Stellen Sie RDQM-Knoten für die von Ihnen ausgewählte Konfiguration bereit oder geben Sie diese an. Weitere Informationen zu den Systemvoraussetzungen für RDQM finden Sie unter „Voraussetzungen für RDQM-DR-Lösung” auf Seite 643.
- Installieren Sie IBM MQ Advanced mit der integrierten RDQM-Funktion auf jedem Knoten.
- Überprüfen Sie optional die RDQM-Konfiguration mit einem Testwarteschlangenmanager, der anschließend gelöscht werden kann. Das Überprüfen der Konfiguration wird empfohlen, um mögliche Probleme vor der Migration des Warteschlangenmanagers zu ermitteln und zu beheben.
- Überprüfen Sie die Sicherheitskonfiguration für den Warteschlangenmanager und replizieren Sie anschließend die erforderlichen lokalen Benutzer und Gruppen auf jedem RDQM-Knoten.
- Überprüfen Sie die Warteschlangenmanager- und Kanalkonfiguration, um zu ermitteln, ob API-Exits, Kanalexits oder Exits zur Datenkonvertierung verwendet werden. Installieren Sie die erforderlichen Exits auf jedem RDQM-Knoten.
- Überprüfen Sie alle Warteschlangenmanagerservices, die definiert wurden, und installieren und konfigurieren Sie anschließend die erforderlichen Prozesse auf jedem RDQM-Knoten.

## Vorgehensweise

1. Sichern Sie den vorhandenen Warteschlangenmanager:
   a) Stoppen Sie den vorhandenen Warteschlangenmanager, indem Sie den Befehl `endmqm -w` ausgeben, mit dem das Beenden verzögert wird, oder indem Sie den Befehl `endmqm -i` für die sofortige Beendigung ausgeben. Dieser Schritt ist wichtig, um sicherzustellen, dass die Daten in der Sicherung konsistent sind.
   b) Ermitteln Sie die Position des Datenverzeichnisses des Warteschlangenmanagers, indem Sie die IBM MQ-Konfigurationsdatei `mqs.ini` anzeigen. Unter Linux befindet sich diese Datei im Verzeichnis `/var/mqm`. Weitere Informationen zu `mqs.ini` finden Sie unter „IBM MQ-Konfigurationsdatei, mqs.ini” auf Seite 91.

      Suchen Sie die Zeilengruppe `QueueManager` für den Warteschlangenmanager in der Datei. Wenn die Zeilengruppe einen Schlüssel mit der Bezeichnung `DataPath` enthält, handelt es sich bei dem zugehörigen Wert um das Datenverzeichnis für den Warteschlangenmanager. Wenn der Schlüssel nicht vorhanden ist, kann das Datenverzeichnis für den Warteschlangenmanager mithilfe der Werte für die Schlüssel `Prefix` und `Directory` ermittelt werden. Das Datenverzeichnis für den Warteschlangenmanager ist eine Verknüpfung dieser Werte im Format *Präfix*/qmgrs/*Verzeichnis*. Weitere Informationen zur Zeilengruppe 'QueueManager' finden Sie unter „Zeilengruppe 'QueueManager' in der Datei 'mqs.ini'” auf Seite 102.
   c) Erstellen Sie eine Sicherung des Warteschlangenmanager-Datenverzeichnisses. Unter Linux können Sie dies mit dem Befehl **tar** tun. Wenn Sie zum Beispiel das Datenverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

      ```
      tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
      ```

d) Ermitteln Sie die Position der Warteschlangenmanager-Protokolldatei, indem Sie sich die IBM MQ Warteschlangenmanager-Konfigurationsdatei `qm.ini` ansehen. Diese Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers. Weitere Informationen zur Datei finden Sie unter „Warteschlangenmanagerkonfigurationsdateien, qm.ini" auf Seite 104.

Das Protokollverzeichnis für den Warteschlangenmanager wird als Wert des Schlüssels `LogPath` in der Zeilengruppe `Log` definiert. Weitere Informationen zu der Zeilengruppe finden Sie unter „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140.

e) Erstellen Sie eine Sicherung des Warteschlangenmanager-Protokollverzeichnisses. Unter Linux können Sie dies mit dem Befehl 'tar' tun. Wenn Sie z. B. das Protokollverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

f) Erstellen Sie eine Sicherungskopie aller Zertifikatsrepositorys, die vom Warteschlangenmanager verwendet werden, wenn diese sich nicht im Datenverzeichnis für den Warteschlangenmanager befinden. Stellen Sie sicher, dass die Schlüsseldatenbankdatei und die Kennwortstashdatei gesichert werden. Weitere Informationen zum Schlüsselrepository für den Warteschlangenmanager finden Sie in den Abschnitten Das SSL/TLS-Schlüsselrepository und Schlüsselrepository für einen Warteschlangenmanager suchen. Weitere Informationen zum Suchen des AMS-Schlüsselspeichers, wenn der Warteschlangenmanager für die Verwendung des Abfangprozesses für den Nachrichtenkanalagenten (Message Channel Agent, MCA) in AMS konfiguriert ist, finden Sie unter Überwachung des Nachrichtenkanalagenten (MCA).

g) Der vorhandene Warteschlangenmanager ist nicht mehr erforderlich und kann deshalb gelöscht werden. Wo dies möglich ist, sollten Sie den vorhandenen Warteschlangenmanager allerdings erst dann löschen, wenn er auf dem Zielsystem erfolgreich wiederhergestellt wurde. Durch das Zurückstellen des Löschvorgangs wird sichergestellt, dass der Warteschlangenmanager erneut gestartet werden kann, wenn der Migrationsprozess nicht erfolgreich abgeschlossen wird.

**Anmerkung:** Wenn Sie das Löschen des vorhandenen Warteschlangenmanagers zurückstellen, starten Sie ihn nicht erneut. Es ist wichtig, dass der Warteschlangenmanager beendet bleibt, da weitere Änderungen an der Konfiguration oder dem Status während der Migration verloren gehen.

2. Bereiten Sie den primären RDQM-Knoten vor:

a) Erstellen Sie einen neuen RDQM-Warteschlangenmanager mit dem gleichen Namen wie der Warteschlangenmanager, den Sie gesichert haben. Stellen Sie sicher, dass das Dateisystem, das dem RDQM-Warteschlangenmanager durch **`crtmqm`** zugeordnet wurde, ausreichend ist für die Daten, primären und sekundären Protokolle für den vorhandenen Warteschlangenmanager sowie für einen zusätzlichen Speicherbereich für zukünftige Erweiterungen. Informationen zum Erstellen eines RDQM-Warteschlangenmanagers finden Sie unter „RDQM für die Wiederherstellung nach einem Katastrophenfall erstellen" auf Seite 645.

b) Bestimmen Sie den primären RDQM-Knoten für den Warteschlangenmanager. Informationen zum Ermitteln des Primärknotens finden Sie im Abschnitt rdqmstatus (RDQM-Status anzeigen).

c) Falls der RDQM-Warteschlangenmanager gestartet ist, stoppen Sie ihn auf dem primären RDQM-Knoten mit dem Befehl `endmqm -w` oder `endmqm -i`.

d) Ermitteln Sie die Daten- und Protokollverzeichnisse für den RDQM-Warteschlangenmanager (verwenden Sie die in den Schritten 1b und 1d beschriebenen Methoden).

e) Löschen Sie die Inhalte der Daten- und Protokollverzeichnisse für den RDQM-Warteschlangenmanager, aber nicht die Verzeichnisse selbst.

3. Stellen Sie den Warteschlangenmanager auf dem primären RDQM-Knoten wieder her:

a) Kopieren Sie die Sicherungsdateien aus den Daten- und Protokollverzeichnissen des Warteschlangenmanagers auf den primären RDQM-Knoten ebenso wie alle separaten Sicherungen von Zertifikatsrepositorys, die vom Warteschlangenmanager verwendet werden.

b) Stellen Sie die Sicherung des Datenverzeichnisses des Warteschlangenmanagers im leeren Datenverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass das Eigentumsrecht und die Berechtigungen der Dateien beibehalten werden. Wenn die Si-

cherungsdateien mithilfe des Beispiels für den Befehl 'tar' in Schritt 1c erstellt wurde, kann der folgende Befehl vom Rootbenutzer für die Wiederherstellung verwendet werden:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

c) Stellen Sie die Sicherung des Protokollverzeichnisses des Warteschlangenmanagers im leeren Protokollverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass das Eigentumsrecht und die Berechtigungen der Dateien beibehalten werden. Wenn die Sicherungsdateien mithilfe des Beispiels für den Befehl 'tar' in Schritt 1e erstellt wurde, kann der folgende Befehl vom Rootbenutzer für die Wiederherstellung verwendet werden:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

d) Bearbeiten Sie die zurückgeschriebene Konfigurationsdatei des Warteschlangenmanagers `qm.ini` im Datenverzeichnis für den RDQM-Warteschlangenmanager. Aktualisieren Sie den Wert des Schlüssels `LogPath` in der Zeilengruppe `Log`, um das Protokollverzeichnis für den RDQM-Warteschlangenmanager anzugeben.

Überprüfen Sie weitere in der Konfigurationsdatei definierte Dateipfade und aktualisieren Sie diese, falls erforderlich. Sie müssen z. B. möglicherweise die folgenden Pfade aktualisieren:

- Den Pfad für Fehlerprotokolldateien, die von Diagnosenachrichtenservices generiert werden.
- Den Pfad für Exits, die vom Warteschlangenmanager benötigt werden.
- Den Pfad für Switchloaddateien, wenn der Warteschlangenmanager ein XA-Transaktionskoordinator ist.

e) Wenn der Warteschlangenmanager für die Verwendung des Abfangprozesses für den Nachrichtenkanalagenten (MCA) in AMS konfiguriert ist, kopieren Sie den AMS-Schlüsselspeicher in die neue RDQM-Installation und prüfen und aktualisieren anschließend die Konfiguration. Der Schlüsselspeicher muss in jedem RDQM-Knoten verfügbar sein; wenn er sich also nicht im replizierten Dateisystem für den Warteschlangenmanager befindet, muss er stattdessen auf jeden Knoten kopiert werden. Weitere Informationen finden Sie unter Überwachung des Nachrichtenkanalagenten (MCA).

f) Stellen Sie sicher, dass der Warteschlangenmanager mit dem Befehl **dspmq** angezeigt wird und dass für ihn der Status 'beendet' gemeldet wird. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-DR-Warteschlangenmanager:

```
$ dspmq -o status -o dr

QMNAME(QM1) STATUS(Ended normally) DRROLE(Primary)
```

g) Stellen Sie sicher, dass die wiederhergestellten Warteschlangenmanagerdaten auf den sekundären RDQM-Knoten repliziert wurden; verwenden Sie dazu den Befehl **rdqmstatus**, um den Status des Warteschlangenmanagers anzuzeigen. Der DR-Status sollte auf jedem der Knoten als `Normal` gemeldet werden. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-DR-Warteschlangenmanager:

```
$ rdqmstatus -m QM1
Queue manager status:                   Ended normally
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

h) Starten Sie den Warteschlangenmanager auf dem primären RDQM-Knoten.

i) Stellen Sie eine Verbindung zum Warteschlangenmanager her und aktualisieren Sie den Wert des Warteschlangenmanagerattributs SSLKEYR, um die neue Position des Zertifikatsrepositorys für den Warteschlangenmanager anzugeben. Standardmäßig ist der Wert dieses Attributs auf *queue_manager_data_directory*/ssl/keygesetzt. Das Zertifikatsrepository muss sich auf jedem RDQM-Knoten an der gleichen Position befinden. Wenn sich das Repository nicht im replizierten Datei-

system für den Warteschlangenmanager befindet, muss es stattdessen auf jeden Knoten kopiert werden.

j) Überprüfen Sie die IBM MQ-Objektdefinitionen für den Warteschlangenmanager und aktualisieren Sie den Wert der Objektattribute, die auf geänderte Netzeinstellungen, das IBM MQ-Installationsverzeichnis oder das Datenverzeichnis des Warteschlangenmanagers verweisen, einschließlich der folgenden Objekte:

- Lokale IP-Adressen, die von Listener verwendet werden (Attribut IPADDR).
- Lokale IP-Adressen, die von Kanälen verwendet werden (Attribut LOCLADDR).
- Lokale IP-Adressen, die für Clusterempfängerkanäle definiert werden (Attribut CONNAME).
- Lokale IP-Adressen, die für Informationsobjekte zur Kommunikation definiert werden (AttributGRPADDR).
- Systempfade, die für Prozess- und Serviceobjektdefinitionen definiert werden.

k) Stoppen Sie den Warteschlangenmanager und starten Sie ihn erneut, um sicherzustellen, dass die Änderungen wirksam werden.

l) Wiederholen Sie Schritt 3j für ferne Warteschlangenmanager und nehmen Sie entsprechende Einstellungen für Anwendungen vor, die eine Verbindung zum migrierten Warteschlangenmanager herstellen, einschließlich der Folgenden:

- Kanalverbindungsnamen (Attribut CONNAME).
- Kanalauthentifizierungsregeln, die eingehende Verbindungen vom Warteschlangenmanager auf Basis der IP-Adresse oder des Hostnamens einschränken.
- Definitionstabellen für den Clientkanal (CCDTs), Einstellungen des Domänennamens (DNS), Netzweiterleitung oder entsprechende Verbindungsinformationen.

m) Führen Sie für jeden RDQM-Knoten eine gesteuerte Funktionsübernahme des Warteschlangenmanagers durch, um sicherzustellen, dass die erforderliche Konfiguration erfolgreich eingerichtet wurde, siehe „Wechsel zu einem Wiederherstellungsknoten” auf Seite 662.

*Größe des Dateisystems für einen DR-RDQM-Warteschlangenmanager ändern*
Um die Größe des Dateisystems für einen vorhandenen DR-RDQM-Warteschlangenmanager mit replizierten Daten (DR = Disaster-Recovery) zu ändern, sichern Sie seine persistenten Daten und stellen anschließend die Daten in einem neu erstellten RDQM-Warteschlangenmanager wieder her, der denselben Namen, aber ein Dateisystem mit einer anderen Größe hat.

## Informationen zu diesem Vorgang

DR-Warteschlangenmanager mit replizierten Daten (RDQM) benötigen einen dedizierten logischen Datenträger (Dateisystem) und die Konfiguration der Plattenreplikation. Diese Komponenten werden nur konfiguriert, wenn ein neuer Warteschlangenmanager erstellt wird. Das Dateisystem kann nach seiner Erstellung nicht geändert werden, da es auf jedem Knoten die gleiche Größe haben muss. Um die Größe des Dateisystems für einen vorhandenen Warteschlangenmanager mit replizierten Daten (RDQM) zu ändern, können Sie seine persistenten Daten sichern und anschließend die Daten in einem neu erstellten RDQM-Warteschlangenmanager wiederherstellen, der denselben Namen, aber ein Dateisystem mit einer anderen Größe hat. Durch diese Vorgehensweise bleiben die Konfiguration, der Status und die persistenten Nachrichten des Warteschlangenmanagers so erhalten, wie sie zum Zeitpunkt der Sicherungserstellung waren.

## Vorgehensweise

1. Sichern Sie den vorhandenen RDQM-WS-Manager auf dem primären RDQM-Knoten:
   a) Bestimmen Sie den primären RDQM-Knoten für den Warteschlangenmanager. Informationen zum Ermitteln des Primärknotens finden Sie im Abschnitt rdqmstatus (RDQM-Status anzeigen).
   b) Falls der RDQM-Warteschlangenmanager gestartet ist, stoppen Sie ihn auf dem primären RDQM-Knoten mit dem Befehl **endmqm -w** oder **endmqm -i**.

c) Ermitteln Sie die Position des Datenverzeichnisses des Warteschlangenmanagers, indem Sie die IBM MQ-Konfigurationsdatei `mqs.ini` anzeigen. Unter Linux befindet sich diese Datei im Verzeichnis `/var/mqm`. Weitere Informationen zu `mqs.ini` finden Sie unter „IBM MQ-Konfigurationsdatei, mqs.ini" auf Seite 91.

Suchen Sie die Zeilengruppe `QueueManager` für den Warteschlangenmanager in der Datei. Das Warteschlangenmanager-Datenverzeichnis ist der Wert des Schlüssels mit dem Namen `DataPath`. Weitere Informationen zu der Zeilengruppe `QueueManager` finden Sie unter „Zeilengruppe 'QueueManager' in der Datei 'mqs.ini'" auf Seite 102.

d) Erstellen Sie eine Sicherung des Warteschlangenmanager-Datenverzeichnisses. Unter Linux können Sie dies mit dem Befehl **tar** tun. Wenn Sie zum Beispiel das Datenverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

e) Ermitteln Sie die Position der Warteschlangenmanager-Protokolldatei, indem Sie sich die IBM MQ Warteschlangenmanager-Konfigurationsdatei `qm.ini` ansehen. Diese Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers. Weitere Informationen zur Datei finden Sie unter „Warteschlangenmanagerkonfigurationsdateien, qm.ini" auf Seite 104.

Das Warteschlangenmanager-Protokollverzeichnis ist als Wert des Schlüssels `LogPath` in der Zeilengruppe 'Log' definiert. Weitere Informationen zu der Zeilengruppe finden Sie unter „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140.

f) Erstellen Sie eine Sicherung des Warteschlangenmanager-Protokollverzeichnisses. Unter Linux können Sie dies mit dem Befehl **tar** tun. Wenn Sie z. B. das Protokollverzeichnis für einen Warteschlangenmanager sichern möchten, können Sie den folgenden Befehl verwenden. Beachten Sie den letzten Parameter des Befehls, bei dem es sich um einen einzelnen Punkt (.) handelt:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

g) Löschen Sie den vorhandenen RDQM-Warteschlangenmanager.

2. Stellen Sie den Warteschlangenmanager mit einem Dateisystem mit der erforderlichen Größe wieder her:

a) Erstellen Sie einen neuen RDQM-Warteschlangenmanager mit dem gleichen Namen wie der Warteschlangenmanager, den Sie gesichert haben. Stellen Sie sicher, dass das Dateisystem, das dem RDQM-Warteschlangenmanager durch **crtmqm** zugeordnet wurde, die benötigte Größe hat und groß genug ist, um die Daten, primären und sekundären Protokolle für den vorhandenen Warteschlangenmanager aufzunehmen sowie zusätzlichen Speicherplatz für zukünftige Erweiterungen zu bieten. Informationen zum Erstellen eines RDQM-Warteschlangenmanagers finden Sie unter „RDQM für die Wiederherstellung nach einem Katastrophenfall erstellen" auf Seite 645.

b) Bestimmen Sie den primären RDQM-Knoten für den Warteschlangenmanager. Informationen zum Ermitteln des Primärknotens finden Sie im Abschnitt rdqmstatus (RDQM-Status anzeigen).

c) Wenn der RDQM-Warteschlangenmanager auf dem primären RDQM-Knoten gestartet ist, stoppen Sie ihn mit dem Befehl **endmqm -w** oder **endmqm -i** .

d) Bestimmen Sie auf dem primären RDQM-Knoten die neue Position der Daten- und Protokollverzeichnisse für den RDQM-Warteschlangenmanager (verwenden Sie die in den Schritten 1c und 1e beschriebenen Methoden).

e) Löschen Sie auf dem primären RDQM-Knoten den Inhalt der RDQM-Warteschlangenmanager-Daten- und -Protokollverzeichnisse, aber nicht die Verzeichnisse selbst.

f) Stellen Sie auf dem primären RDQM-Knoten die Sicherung des Warteschlangenmanager-Datenverzeichnisses in dem leeren Datenverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass die Dateieigentumsrechte und die Berechtigungen beibehalten werden. Wenn die Sicherung mit dem in Schritt 1d gezeigten Beispielbefehl **tar** erstellt wurde, kann vom Rootbenutzer für die Wiederherstellung der folgende Befehl verwendet werden:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

g) Stellen Sie auf dem primären RDQM-Knoten die Sicherung des Warteschlangenmanager-Protokollverzeichnisses in dem leeren Protokollverzeichnis für den neuen RDQM-Warteschlangenmanager wieder her und stellen Sie dabei sicher, dass die Dateieigentumsrechte und die Berechtigungen beibehalten werden. Wenn die Sicherung mit dem in Schritt 1f gezeigten Beispielbefehl **tar** erstellt wurde, kann vom Rootbenutzer für die Wiederherstellung der folgende Befehl verwendet werden:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

h) Bearbeiten Sie auf dem primären RDQM-Knoten die zurückgeschriebene Konfigurationsdatei des Warteschlangenmanagers `qm.ini` im Datenverzeichnis für den neuen RDQM-Warteschlangenmanager. Aktualisieren Sie den Wert des Schlüssels `LogPath` in der Zeilengruppe `Log`, so dass dort das in Schritt 2d bestimmte Protokollverzeichnis für den neuen RDQM-Warteschlangenmanager angegeben ist. Überprüfen Sie weitere in der Konfigurationsdatei definierte Dateipfade und aktualisieren Sie diese, falls erforderlich. Sie müssen z. B. möglicherweise die folgenden Pfade aktualisieren:

- Den Pfad für Fehlerprotokolldateien, die von Diagnosenachrichtenservices generiert werden.
- Den Pfad für Exits, die vom Warteschlangenmanager benötigt werden.
- Den Pfad für Switchloaddateien, wenn der Warteschlangenmanager ein XA-Transaktionskoordinator ist.

i) Überprüfen Sie, ob der WS-Manager vom **dspmq** -Befehl angezeigt wird und sein Status als `ended` gemeldet wird. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-DR-Warteschlangenmanager:

```
$ dspmq -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DR(Primary)
```

j) Stellen Sie sicher, dass die wiederhergestellten Warteschlangenmanagerdaten auf dem sekundären RDQM-Knoten repliziert wurden; verwenden Sie dazu den Befehl **rdqmstatus**, um den Status des Warteschlangenmanagers anzuzeigen. Der DR-Status sollte auf jedem der Knoten als `Normal` gemeldet werden. Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-DR-Warteschlangenmanager auf dem primären Knoten:

```
$ rdqmstatus -m QM1
Queue manager status:                 Running
CPU:                                  0.00
Memory:                               123MB
Queue manager file system:            51MB used, 1.0GB allocated [5%]
DR role:                              Primary
DR status:                            Normal
DR type:                              Synchronous
DR port:                              3000
DR local IP address:                  192.168.20.1
DR remote IP address:                 192.168.20.2
```

Das folgende Beispiel zeigt eine Beispielausgabe für einen RDQM-DR-Warteschlangenmanager auf dem Recovery-Knoten:

```
Queue manager status:                 Ended immediately
DR role:                              Secondary
DR status:                            Normal
DR port:                              3000
DR local IP address:                  192.168.20.2
DR remote IP address:                 192.168.20.1
```

k) Starten Sie den Warteschlangenmanager auf dem primären RDQM-Knoten.

l) Führen Sie für den Recovery-Knoten eine Umschaltung des Warteschlangenmanagers durch, um sicherzustellen, dass die erforderliche Konfiguration erfolgreich eingerichtet wurde, siehe „Wechsel zu einem Wiederherstellungsknoten” auf Seite 662.

*Persistenten Anwendungsstatus speichern*
Persistente Statusinformationen zu Anwendungen können zusammen mit anderen Warteschlangenmanagerdaten gespeichert werden.

Jeder IBM MQ-Warteschlangenmanager verfügt über ein dediziertes Dateisystem für seinen persistenten Status, was sowohl seine Warteschlangendaten als auch das Wiederherstellungsprotokoll einschließt. In einer RDQM-Konfiguration wird das Dateisystem durch einen logischen Datenträger gesichert, der zwischen den Linux-Systemen (Knoten) repliziert wird. Das Dateisystem enthält ein `userdata`-Verzeichnis, das Sie zum Speichern persistenter Statusinformationen für Ihre Anwendungen verwenden können. Wenn ein RDQM (Replicated Data Queue Manager) auf einen anderen Knoten in Ihrer RDQM-Konfiguration verschoben wird, verfügen Sie auf diese Weise sowohl über den Anwendungs- als auch den Warteschlangenmanagerkontext. Weitere Informationen finden Sie im Abschnitt Verzeichnisinhalt auf Unix- und Linux-Systemen.

Wenn Sie den Anwendungsstatus im Verzeichnis `userdata` speichern, müssen Sie sich darüber im Klaren sein, dass Daten, die an diese Position geschrieben werden, möglicherweise den verfügbaren Plattenspeicherplatz belegen, der dem Warteschlangenmanager zugeordnet ist. Sie müssen sicherstellen, dass für den Warteschlangenmanager genügend Plattenspeicherplatz zum Speichern von Warteschlangendaten, Protokollen und anderen persistenten Statusinformationen verfügbar ist.

Das Verzeichnis `userdata` hat den Benutzer 'mqm' und das Gruppeneigentum, und es ist weltweit lesbar, sodass Benutzer darauf zugreifen können, ohne sich in der Administratorgruppe von IBM MQ (d. h. mqm) befinden zu müssen. Sie können die Berechtigungen des Verzeichnisses `userdata` nicht ändern, aber Sie können Inhalte in ihr erstellen, unabhängig davon, welche Eigentümer und Berechtigungen Sie benötigen.

Bei der Übernahme eines RDQM-Warteschlangenmanagers wird der Warteschlangenmanager beendet und sein Dateisystem wird auf seinem aktuellen RDQM-Knoten abgehängt. Das Dateisystem wird anschließend auf einem anderen Knoten in der RDQM-Konfiguration angehängt und der Warteschlangenmanager dort erneut gestartet. Ein Dateisystem kann nicht abgehängt werden, solange es einen Prozess mit einer offenen Kennung für eine seiner Dateien gibt. Um sicherzustellen, dass eine Warteschlangenmanagerübernahme auch dann abgeschlossen werden kann, wenn das Dateisystem des Warteschlangenmanagers nicht abgehängt werden kann, wird an alle Prozesse mit einer offenen Dateikennung ein SIGTERM-Signal gesendet, gefolgt von einem SIGKILL, falls die offenen Kennungen nicht freigegeben werden. Ihre Anwendungen müssen so konzipiert sein, dass sie korrekt auf SIGTERM reagieren. Wenn Anwendungen oder Prozesse als Warteschlangenmanagerservice konfiguriert sind, können sie während einer verwalteten Übernahme beim Herunterfahren des Warteschlangenmanagers beendet werden, bevor das Dateisystem abgehängt wird. Wenn eine Anwendung oder ein Prozess nicht als Warteschlangenmanagerservice konfiguriert ist oder eine nicht verwaltete Übernahme erfolgt, z. B. nach einem Verlust des Quorums, ist es wahrscheinlich, dass Signale gesendet werden, um das Dateisystem freizugeben.

## Linux *Primäre und sekundäre Merkmale von DR-RDQMs verwalten*

Sie können einen sekundären Wiederherstellungsdatenwarteschlangenmanager (DR RDQM) in eine primäre DR RDQM-Datei ändern. Sie können auch eine primäre Instanz in eine sekundäre Instanz ändern.

### Informationen zu diesem Vorgang

Mit dem Befehl **`rdqmdr`** können Sie eine sekundäre Instanz eines RDQM in die primäre Instanz ändern. Möglicherweise müssen Sie diese Aktion ausführen, wenn Sie Ihre primäre Instanz aus einem bestimmten Grund verlieren. Anschließend können Sie den Warteschlangenmanager starten und die Ausführung des Warteschlangenmanagers auf dem Wiederherstellungsknoten ausführen.

Mit dem Befehl **`rdqmdr`** können Sie auch eine primäre Instanz eines RDQM in die sekundäre Instanz ändern. Möglicherweise müssen Sie diese Aktion ausführen, z. B. wenn Sie Ihr System neu konfigurieren.

Sie können auch **`rdqmdr`** auf einem primären Warteschlangenmanager verwenden, um den genauen Befehl abzurufen, den Sie benötigen, um eine sekundäre Instanz dieses Warteschlangenmanagers auf Ihrem Wiederherstellungsknoten zu erstellen.

Sie können den Befehl **`rdqmdr`** als Benutzer in der Gruppe `mqm` verwenden, wenn der Benutzer sudo verwenden kann. Andernfalls müssen Sie als Root angemeldet sein.

## Prozedur

- Geben Sie den folgenden Befehl ein, um eine sekundäre Instanz von DR RDQM in eine primäre Instanz zu ändern:

  ```
  rdqmdr -m QMname -p
  ```

  Dieser Befehl schlägt fehl, wenn die primäre Instanz des Warteschlangenmanagers noch aktiv ist und die DR-Replikationsverbindung noch funktioniert.
- Geben Sie den folgenden Befehl ein, um eine primäre Instanz des Warteschlangenmanagers in eine sekundäre Instanz zu ändern:

  ```
  rdqmdr -m QMname -s
  ```

- Geben Sie den folgenden Befehl auf Ihrem Primärknoten ein, um den Befehl **crtmqm** anzuzeigen, der zum Konfigurieren der sekundären Instanz eines Warteschlangenmanagers erforderlich ist:

  ```
  rdqmdr -d -m QMname
  ```

  Sie können den zurückgegebenen Befehl **crtmqm** auf Ihrem sekundären Knoten eingeben, um die sekundäre Instanz von RD RDQM zu erstellen.

## Linux *Starten, Stoppen und Anzeigen des Status eines DR RDQM*

Zum Starten und Stoppen eines Disaster-Recovery-Warteschlangenmanagers mit replizierten Daten (Disaster Recovery Replicated Data Queue Manager, DR RDQM) sowie zum Anzeigen seines aktuellen Status werden Varianten der standardmäßigen IBM MQ-Steuerbefehle verwendet.

## Informationen zu diesem Vorgang

Sie müssen die Befehle ausführen, die den aktuellen Status eines replizierten Datenwarteschlangenmanagers (RDQM) als Benutzer, der zu der Gruppe `mqm` gehört, starten, stoppen und anzeigen.

Sie müssen die Befehle ausführen, um einen WS-Manager auf dem Primärknoten für diesen Warteschlangenmanager zu starten und zu stoppen (d. B. der Knoten, auf dem der Warteschlangenmanager derzeit ausgeführt wird).

## Prozedur

- Geben Sie zum Starten eines DR RDQM den folgenden Befehl auf dem Primärknoten des RDQM-Systems ein:

  ```
  strmqm qmname
  ```

  Hierbei steht *qmname* für den Namen des RDQM, den Sie starten wollen.
- Geben Sie zum Stoppen eines RDQM den folgenden Befehl auf dem Primärknoten des RDQM-Systems ein:

  ```
  endmqm qmname
  ```

  Hierbei steht *qmname* für den Namen des RDQM, den Sie stoppen wollen.
- Geben Sie den folgenden Befehl ein, um den Status eines RDQM anzuzeigen:

  ```
  dspmq -m QMname
  ```

  Die Statusinformationen, die ausgegeben werden, hängen davon ab, ob Sie den Befehl auf dem Primär- oder Sekundärknoten des RDQM ausführen. Wenn die Ausführung auf dem Primärknoten erfolgt, wird eine der normalen Statusnachrichten angezeigt, die von **dspmq** zurückgegeben werden. Wenn Sie den Befehl auf einem Sekundärknoten ausführen, wird der Status `Ended immediately` angezeigt.

Wird **dspmq** beispielsweise auf dem Knoten RDQM7 ausgeführt wird, werden möglicherweise folgende Informationen zurückgegeben:

```
QMNAME(DRQM8)                           STATUS(Ended immediately)
QMNAME(DRQM7)                           STATUS(Running)
```

Sie können Argumente mit dspmq verwenden, um festzustellen, ob ein RDQM für Disaster-Recovery konfiguriert ist und ob es sich aktuell um die Primär- oder Sekundärinstanz handelt:

```
dspmq -m QMname -o (dr | DR)
```

Eine der folgenden Antworten wird angezeigt:

**DRROLE()**
Gibt an, dass der Warteschlangenmanager nicht für Disaster-Recovery konfiguriert ist.

**DRROLE(Primary)**
Gibt an, dass der Warteschlangenmanager als der DR-Primärmanager konfiguriert ist.

**DRROLE(Secondary)**
Gibt an, dass der Warteschlangenmanager als der DR-Sekundärmanager konfiguriert ist.

**Zugehörige Verweise**
dspmq
endmqm
strmqm

## Linux *DR-RDQM-Status anzeigen*

Sie können den Status aller replizierten Datenwarteschlangenmanager für Notfallwiederherstellung (DR RDQMs) auf einem Knoten oder detaillierte Informationen zu einem angegebenen DR RDQM anzeigen.

### Informationen zu diesem Vorgang

Sie verwenden den Befehl **rdqmstatus** , um den Status aller DR RDQMs oder einzelner RDQMs anzuzeigen.

V 9.3.0 Der Zusammenfassungsstatus für einen Knoten zeigt auch Informationen über das DRBD-Kernelmodul an, auf das RDQM angewiesen ist. Wenn Sie RDQM aktualisieren, ist es wichtig, sicherzustellen, dass die korrekte Version des DRBD Kernelmoduls installiert ist für die Version des RHEL Kernels, der auf dem System läuft. Der Status zeigt die Version des Betriebssystem-Kernels, die Kernelversion, für die das DRBD-Modul erstellt wurde, die DRBD-Version und den Status des DRBD-Kernelmoduls an.

Sie müssen ein Benutzer in der Gruppe mqm sein, um den Befehl **rdqmstatus** ausführen zu können. Sie können den Befehl auf jedem Knoten des DR-RDQM-Paars ausführen.

### Prozedur

- Wenn Sie den Zusammenfassungsstatus aller DR-RDQMs auf einem Knoten anzeigen möchten, führen Sie den folgenden Befehl auf diesem Knoten aus:

  ```
  rdqmstatus
  ```

  Der Status der DR-RDQMs auf dem Knoten wird angezeigt, z. B.:

  ```
  Node:                                 mqhavm07.exampleco.com
  OS kernel version:                    3.10.0-1160.15.2
  DRBD OS kernel version:               3.10.0-1160
  DRBD version:                         9.1.1
  DRBD kernel module status:            Loaded

  Queue manager name:                   DRQM8
  Queue manager status:                 Ended immediately
  DR role:                              Secondary

  Queue manager name:                   DRQM7
  ```

```
Queue manager status:                   Running
DR role:                                Primary
```

V 9.3.0 Der Status des DRBD-Kernelmoduls ist einer der folgenden Werte:

**Geladen**
Gibt an, dass das DRBD-Modul geladen wurde.

**Teilweise geladen**
Kann auftreten, wenn das DRBD-Modul geladen wurde, aber aufgrund einer Abweichung nicht ordnungsgemäß funktioniert.

**Nicht geladen**

Das DRBD-Modul ist nicht geladen. Diese Option kann auf einer neu installierten Konfiguration angezeigt werden, wenn noch keine RDQM-Warteschlangenmanager erstellt wurden.

**Nicht installiert**
Gibt an, dass das DRBD-Modul nicht installiert ist. oder dass IBM MQ die Betriebssystem-Kernelversion des DRBD-Moduls nicht bestimmen konnte.

**Zuvor installierte Version ist noch geladen**
Dieser Status kann auftreten, wenn ein neues DRBD-Modul installiert wird, während das vorhandene DRBD-Modul ausgeführt wird (d. h. ein RDQM-Warteschlangenmanager wird ausgeführt). Das neu installierte Modul wird im Status gemeldet, ist aber nicht das Modul, das tatsächlich ausgeführt wird.

- Geben Sie den folgenden Befehl ein, um den Status eines bestimmten RDQM anzuzeigen:

```
rdqmstatus -m  qmname
```

In der folgenden Tabelle sind die Informationen zusammengefasst, die zurückgegeben werden.

*Tabelle 34. Status-Attribute*

| **Statusattribut** | **Mögliche Werte** | **Wenn angezeigt** |
|---|---|---|
| Status des Warteschlangenmanagers | Status (wie von dspmq angezeigt) | Immer |
| CPU | *n*. *nn*% | Nur angezeigt, wenn RDQM auf dem aktuellen Knoten die primäre Rolle hat |
| Hauptspeicher | *nnn* MB | Nur angezeigt, wenn RDQM auf dem aktuellen Knoten die primäre Rolle hat |
| Queue manager file system | *nnn* MB verwendet ,*n*. *n* GB zugeordnet [ *n*%] | Nur angezeigt, wenn RDQM auf dem aktuellen Knoten die primäre Rolle hat |
| DR role | Primär<br>Sekundär<br>Unbekannt | Immer |
| DR-Status | Normal | Normaler Betrieb |
| | Synchronization in progress | Die Synchronisation ist in Bearbeitung. |
| | Partitioniert | Der WS-Manager wurde auf beiden Knoten gestartet, während das DR-Replikationsnetz nicht verfügbar ist. |

*Tabelle 34. Status-Attribute (Forts.)*

| Statusattribut | Mögliche Werte | Wenn angezeigt |
|---|---|---|
| | Fernes System nicht verfügbar | Die Verbindung zum anderen Knoten ist verloren gegangen. |
| | Inkonsistent | Es wurde eine Synchronisation in Bearbeitung, wurde aber unterbrochen |
| | Zurücksetzen auf Momentaufnahme | Der Benutzer hat die Zurücksetzung auf die Momentaufnahme ausgewählt, die ausgeführt wurde, als der WS-Manager in den Status Inkonsistent eingegeben wurde. |
| | Das ferne System wurde nicht konfiguriert. | Die primäre Instanz von RDQM wurde konfiguriert, es wurde jedoch keine sekundäre Instanz konfiguriert. |
| | Fehlgeschlagene Vereinbarung | Einer der Knoten wurde für die synchrone Replikation und die andere für die asynchrone Replikation festgelegt. |
| DR-Typ | Synchron oder asynchron | Immer |
| DR port | *port_number* (der TCP/IP-Port, der für die Replikation der Daten dieses Warteschlangenmanagers verwendet wird) | Immer |
| DR local IP address | Die lokale IP-Adresse, die dieser WS-Manager von für DR repliziert | Immer |
| Ferne IP-IP-Adresse | Die ferne IP-Adresse, die dieser WS-Manager für DR repliziert | Immer |
| DR out of sync data | *n* KB | Wird angezeigt, wenn der ferne Knoten nicht verfügbar oder inkonsistent |
| DR synchronization progress | *n*% | Wird angezeigt, wenn die Synchronisation in Bearbeitung ist. |
| DR estimated time to completion | *YYYY-MM-DD HH*: *MM*: *SS* | Wird angezeigt, wenn die Synchronisation in Bearbeitung ist. |
| Snapshot reversion progress | *n*% | Wird angezeigt, wenn der DR-Status `Reverting to snapshot` ist. Der Status zählt nach unten, also 0% zeigt die Fertigstellung an. |
| V 9.3.0 V 9.3.0 DR last in sync | *YYYY-MM-DD HH*: *MM*: *SS* | Wird angezeigt, wenn die DR-Daten nicht synchron sind (nach der Erstsynchronisation). Gibt Zeit und Datum an, als die Daten zuletzt synchron waren. |

**Beispiel**

Beispiel für einen normalen Status auf dem Primärknoten:

```
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

Beispiel für einen normalen Status auf einem Sekundärknoten:

```
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.20.2
DR remote IP address:                   192.168.20.1
```

Beispiel für den Status auf dem Primärknoten, wenn die Synchronisation in Bearbeitung ist:

```
Queue manager status:                   Running
CPU:                                    0.53
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Synchronization in progress
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
DR synchronization progress:            11.0%
DR estimated time to completion:        2017-09-06 14:55:05
```

Beispiel für einen Primärknoten, der anzeigt, dass er partitioniert ist:

```
Queue manager status:                   Running
CPU:                                    0.02
Memory:                                 124MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Partitioned
DR type:                                Synchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
```

V 9.3.0 Beispiel eines Primärknotens, das zeigt, dass er nicht mit dem Sekundärknoten synchron ist:

```
Queue manager status:                   Running
CPU:                                    0.00
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
DR role:                                Primary
DR status:                              Remote unavailable
DR type:                                Asynchronous
DR port:                                3000
DR local IP address:                    192.168.20.1
DR remote IP address:                   192.168.20.2
DR out of sync data:                    15932KB
DR last in sync:                        2020-07-27 16:01:47
```

V 9.3.0 Beispiel für einen Zusammenfassungsstatus, der eine Diskrepanz zwischen der Betriebssystem-Kernelversion (RHEL 7.9) und dem DRBD-Kernelmodul (für RHEL 7.8) zeigt. Obwohl der Status meldet, dass das DRBD-Kernelmodul geladen ist und der erwartete Warteschlangenmanager aktiv ist, sollten Sie das DRBD-Kernelmodul mit der Version aktualisieren, die für den aktiven Betriebssystemkern in dieser Situation vorgesehen ist.

```
Node:                                   mqhavm07.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     DRQM8
Queue manager status:                   Ended immediately
DR role:                                Secondary

Queue manager name:                     DRQM7
Queue manager status:                   Running
DR role:                                Primary
```

V 9.3.0 Beispiel für einen Zusammenfassungsstatus, der eine Diskrepanz zwischen der Betriebssystem-Kernelversion (RHEL 7.9) und dem DRBD-Kernelmodul (für RHEL 7.6) zeigt. In diesem Beispiel ist die Versionsabweichung schwerwiegender, und das DRBD-Kernelmodul kann nicht erfolgreich geladen werden. QM3 ist ein DR-Warteschlangenmanager und soll die primäre Instanz sein. Da das DRBD-Kernelmodul jedoch nicht vollständig geladen wurde, wird als sekundärer Warteschlangenmanager mit dem DR-Status Unknowngemeldet. Um diesen Fehler zu beheben, muss das DRBD-Kernelmodul mit dem Versionsziel für den aktiven Betriebssystemkern aktualisiert werden.

```
Node:                                   mqhavm57.exampleco.com
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM3
Queue manager status:                   Status not available
DR role:                                Secondary
DR status:                              Unknown
```

**Zugehörige Verweise**

Linux rdqmstatus

## Linux *Betrieb in einer Disaster-Recovery-Umgebung*

Es gibt eine Reihe von Situationen, in denen Sie möglicherweise in eine Konfiguration zur Wiederherstellung nach einem Katastrophenfall auf den sekundären Warteschlangenmanager umschalten möchten.

**Wiederherstellung nach einem Katastrophenfall**

Nach dem vollständigen Verlust des primären Warteschlangenmanagers am Hauptstandort wird der sekundäre WS-Manager am Wiederherstellungsstandort gestartet. Anwendungen stellen die Verbindung zum WS-Manager auf der Wiederherstellungssite wieder her und der sekundäre Warteschlangenmanager verarbeitet Anwendungsnachrichten. Die Schritte, die zum Zurücksetzen auf die vorherige Konfiguration ausgeführt wurden, hängen von der Ursache des Fehlers ab. Beispiel: vollständiger Verlust des Hauptknotens im Vergleich zum temporären Verlust.

Informationen zu den Schritten, die nach einem temporären Verlust des Hauptstandortes ausgeführt werden müssen, finden Sie in „Wechsel zu einem Wiederherstellungsknoten” auf Seite 662. Informationen zu den Schritten, die nach einem permanenten Fehler ausgeführt werden müssen, finden Sie in „Fehlgeschlagenen Knoten in einer Wiederherstellung nach einem Katastrophenfall ersetzen” auf Seite 663.

**Testunterstützung für Disaster Recovery**

Sie können die Konfiguration für die Notfallwiederherstellung testen, indem Sie vorübergehend auf die sekundäre Instanz umschalten und überprüfen, ob Anwendungen erfolgreich eine Verbindung herstellen können. Sie befolgen die gleiche Vorgehensweise wie beim Umschalten nach einem temporären Ausfall des Primärknotens, siehe „Wechsel zu einem Wiederherstellungsknoten” auf Seite 662.

**Zurücksetzen auf Momentaufnahme**

Wenn Sie während einer Synchronisierung im Primärknoten einen Fehler erleiden, können Sie die Momentaufnahme der sekundären WS-Manager-Daten unmittelbar vor dem Start der Synchronisation zurücksetzen. Der sekundäre Status wird dann in einen konsistenten Status zurückgeschrieben und kann als primärer Status ausgeführt werden. Zum Zurücksetzen auf die Momentaufnahme machen

Sie den sekundären zum primären Status, wie in „Wechsel zu einem Wiederherstellungsknoten” auf Seite 662 beschrieben. Sie müssen überprüfen, ob die Zurücksetzung auf die Momentaufnahme abgeschlossen ist (mit dem Befehl **rdqmstatus** ), bevor Sie den Warteschlangenmanager starten.

Linux *Wechsel zu einem Wiederherstellungsknoten*

Wenn an Ihrem Hauptstandort ein Katastrophenfall auftritt, führen Sie die Schritte aus, um zu Ihrer Wiederherstellungssite umzuschalten.

## Informationen zu diesem Vorgang

Nach dem Verlust des primären WS-Managers am Hauptstandort machen Sie den sekundären Warteschlangenmanager am Wiederherstellungsstandort in den Primärwarteschlangenmanager und starten ihn. Anwendungen stellen die Verbindung zum WS-Manager auf der Wiederherstellungssite wieder her und der Warteschlangenmanager verarbeitet Anwendungsnachrichten. Sie können diese Prozedur auch verwenden, um Ihren Wiederherstellungsknoten zu testen.

**Wichtig:** Sie müssen sicherstellen, dass die primäre Instanz eines Warteschlangenmanagers entweder nicht ausgeführt oder gestoppt und zu einer sekundären Instanz gemacht wurde, bevor Sie die ursprüngliche sekundäre Instanz hochstufen. Andernfalls können sich partitionierte Daten ansammeln.

Sie müssen entweder als Root angemeldet oder als Benutzer angemeldet sein, der zur Gruppe `mqm` gehört und über die erforderliche sudo-Konfiguration verfügt.

## Vorgehensweise

1. Wenn Sie diese Prozedur zum Testen Ihres sekundären Warteschlangenmanagers verwenden (d. B. die primäre Instanz ist noch aktiv), müssen Sie die primäre Instanz stoppen und sie als sekundäre Instanz neu definieren:

   ```
   endmqm qmname
   rdqmdr -m qmname -s
   ```

2. Machen Sie den sekundären Warteschlangenmanager in den Primärwarteschlangenmanager, indem Sie den folgenden Befehl auf dem Wiederherstellungsknoten eingeben:

   ```
   rdqmdr -m qmname -p
   ```

3. Starten Sie den Warteschlangenmanager, indem Sie den folgenden Befehl eingeben:

   ```
   strmqm qmname
   ```

4. Stellen Sie sicher, dass Ihre Anwendungen die Verbindung zum Warteschlangenmanager auf dem Wiederherstellungswarteschlangenmanager wiederherstellen. Wenn Sie die Kanäle mit einer Liste alternativer Verbindungsnamen definiert haben, die Ihre primären und sekundären Warteschlangenmanager angeben, werden Ihre Anwendungen automatisch eine Verbindung zum neuen primären Warteschlangenmanager herstellen.

## Nächste Schritte

Wenn der ausgefallene Knoten wiederhergestellt wird, vorausgesetzt, dass die Verbindung zwischen den beiden Knoten funktioniert, kann der Warteschlangenmanager auf diesem Knoten nicht gestartet werden, da er auf dem Wiederherstellungsknoten ausgeführt wird, auf dem Sie die sekundäre Warteschlangenmanagerinstanz hochgestuft haben. Wenn Sie zur normalen Operation zurückkehren möchten, müssen Sie den Warteschlangenmanager auf dem Wiederherstellungsknoten stoppen und anschließend den Warteschlangenmanager auf dem ursprünglichen Knoten wieder in die primäre Rolle hochstufen.

**Zugehörige Verweise**

strmqm

rdqmdr

*RDQM-Warteschlangenmanager für Wiederherstellung testen*
Sie können testen, ob die Wiederherstellungsinstanz eines Warteschlangenmanagers in einer RDQM-Disaster-Recovery-Konfiguration ordnungsgemäß funktioniert, ohne den Hauptstandort zu unterbrechen.

## Informationen zu diesem Vorgang

Sie testen den Wiederherstellungswarteschlangenmanager, indem Sie die Schnittstelle zwischen Haupt- und Wiederherstellungsknoten inaktivieren. Sie können den sekundären Warteschlangenmanager zum primären Warteschlangenmanager machen und dann den eigenständigen Warteschlangenmanager testen. Nach Abschluss des Tests stellen Sie die Schnittstelle wieder her und löschen den Testwarteschlangenmanager. Anschließend erstellen Sie den Warteschlangenmanager als sekundären Warteschlangenmanager in der Disaster-Recovery-Konfiguration erneut.

## Vorgehensweise

1. Inaktivieren Sie die Netzverbindung zwischen dem Hauptknoten und dem Wiederherstellungsknoten.
2. Legen Sie auf dem Wiederherstellungsknoten den Warteschlangenmanager als primären Warteschlangenmanager fest:

   ```
   rdqmdr -m QMname -p
   ```

   Dabei ist *QMname* der Name des Warteschlangenmanagers.
3. Starten Sie den Warteschlangenmanager:

   ```
   strmqm QMname
   ```
4. Verbinden Sie Anwendungen mit dem Warteschlangenmanager und testen Sie, ob sie wie erwartet funktionieren.
5. Beenden Sie den Warteschlangenmanager:

   ```
   endmqm QMname
   ```
6. Löschen Sie den Warteschlangenmanager:

   ```
   dltmqm QMname
   ```
7. Stellen Sie die Netzverbindung zwischen der Haupt- und der Wiederherstellungsappliance wieder her.
8. Führen Sie auf dem Hauptknoten den folgenden Befehl zum Abrufen des Befehls **crtmqm** aus, den Sie beim ersten Konfigurieren der Disaster-Recovery verwendet haben.

   ```
   rdqmdr -d -m QMname
   ```
9. Führen Sie den resultierenden Befehl **crtmqm** auf dem Wiederherstellungsknoten aus, um den sekundären Warteschlangenmanager erneut zu erstellen. Der primäre Warteschlangenmanager auf dem Hauptknoten synchronisiert seine Daten mit dem sekundären Warteschlangenmanager, um ihn auf den neuesten Stand zu bringen.

Linux *Fehlgeschlagenen Knoten in einer Wiederherstellung nach einem Katastrophenfall ersetzen*
Wenn Sie einen der Knoten in einer Konfiguration zur Wiederherstellung nach einem Katastrophenfall verloren haben, können Sie den Knoten ersetzen und die Konfiguration für die Wiederherstellung nach einem Katastrophenfall wiederherstellen, indem Sie die folgende Prozedur befolgen.

## Informationen zu diesem Vorgang

Wenn eine solche Katastrophe eintritt, dass der Knoten in der Hauptsite nicht repariert werden kann, können Sie den fehlgeschlagenen Knoten ersetzen, während der Warteschlangenmanager auf dem Wiederherstellungsknoten ausgeführt wird, und anschließend die ursprüngliche Konfiguration zur Wiederher-

stellung nach einem Katastrophenfall wiederherstellen. Der Ersatzknoten muss die Identität des fehlgeschlagenen Knotens annehmen: der Name und die IP-Adresse müssen identisch sein.

Sie müssen entweder als Root angemeldet oder als Benutzer angemeldet sein, der zur Gruppe `mqm` gehört und über die erforderliche sudo-Konfiguration verfügt.

## Vorgehensweise

Führen Sie nach dem Verlust des Warteschlangenmanagers auf der Hauptseite die folgenden Schritte aus:

1. Führen Sie auf dem Wiederherstellungsknoten die folgenden Befehle aus, damit der sekundäre Warteschlangenmanager die primäre Rolle übernimmt:

   ```
   rdqmdr -m QMname -p
   ```

   Dabei ist *QMname* der Name des Warteschlangenmanagers.
2. Rufen Sie den Befehl ab, den Sie auf dem Ersatzprimärknoten ausführen müssen, um die Wiederherstellung nach einem Katastrophenfall zu rekonfigurieren:

   ```
   rdqmdr -m QMname -d
   ```

   Kopieren Sie die Ausgabe dieses Befehls.
3. Führen Sie den folgenden Befehl aus, um den WS-Manager zu starten:

   ```
   strmqm QMname
   ```
4. Stellen Sie sicher, dass Ihre Anwendungen die Verbindung zum WS-Manager auf dem Wiederherstellungsknoten herstellen. Wenn Sie die Kanäle mit einer Liste alternativer Verbindungsnamen definiert haben, die Ihre primären und sekundären Warteschlangenmanager angeben, werden Ihre Anwendungen automatisch eine Verbindung zum neuen primären Warteschlangenmanager herstellen.
5. Ersetzen Sie den ausgefallenen Knoten auf Ihrer Hauptseite, und konfigurieren Sie ihn so, dass er denselben Namen und dieselbe IP-Adresse hat, die Sie für die Wiederherstellung nach einem Katastrophenfall auf dem ursprünglichen Knoten verwendet haben. Konfigurieren Sie anschließend die Disaster-Recovery, indem Sie den in Schritt 2 kopierten Befehl **crtmqm** ausführen. Sie haben jetzt eine sekundäre Instanz des Warteschlangenmanagers, und die primäre Instanz synchronisiert ihre Daten mit der sekundären Instanz.
6. Beenden Sie die aktuelle primäre Instanz.
7. Nachdem die Synchronisation abgeschlossen ist, machen Sie die primäre Instanz, die auf dem Wiederherstellungsknoten ausgeführt wird, wieder in die sekundäre Instanz:

   ```
   rdqmdr -m QMname -s
   ```
8. Stellen Sie auf dem Ersatzprimärknoten die sekundäre Instanz des Warteschlangenmanagers in die primäre Instanz ein:

   ```
   rdqmdr -m QMname -p
   ```
9. Starten Sie den WS-Manager auf dem Ersatzprimärknoten:

   ```
   strmqm QMname
   ```

   Sie haben jetzt die Konfiguration wiederhergestellt, wie sie vor dem Ausfall am Hauptstandort war.

**Zugehörige Verweise**
strmqm
rdqmdr
endmqm

*Inkonsistenz-Problem bei DR-RDQM beheben*
Der DR-Status `inconsistent` kann gemeldet werden, wenn die Synchronisation zwischen den primären und sekundären Instanzen eines Warteschlangenmanagers fehlschlägt.

## Informationen zu diesem Vorgang

Ein inkonsistenter Status wird in der sekundären Instanz eines Warteschlangenmanagers gemeldet, weil die Replikationsverbindung mit der primären Instanz während einer Synchronisationsoperation verloren geht. Möglicherweise müssen Sie entsprechende Maßnahmen ergreifen, um diese Situation zu beheben. Stellen Sie sich die folgende Abfolge von Ereignissen vor:

1. Primärer DR-Warteschlangenmanager ist mit sekundärem DR-Warteschlangenmanager synchronisiert
2. Replikationsverbindung zwischen primär und sekundär verloren
3. Replikationsverbindung zwischen primär und sekundär wiederhergestellt
4. Es erfolgt eine Resynchronisation, bei der der sekundäre DR-Warteschlangenmanager an den primären DR-Warteschlangenmanager angeglichen wird. Während dieser Zeit wird der DR-Status von `synchronization in progress` für beide Warteschlangenmanager gemeldet.
5. Wenn die Replikation während der Resynchronisation wieder verloren geht, wird der Status auf dem sekundären DR-Datenträger als `Inconsistent`gemeldet.

Wenn der Knoten, auf dem sich der primäre Warteschlangenmanager befindet, noch betriebsbereit ist und die Replikationsverbindung wiederhergestellt werden kann, wird die Resynchronisation automatisch durchgeführt. Der inkonsistente Status wird behoben, ohne dass Sie eingreifen mussten.

Wenn der Knoten, auf dem sich der primäre Warteschlangenmanager befindet, nicht mehr betriebsbereit ist, können Sie den inkonsistenten Status beheben, indem Sie auf dem sekundären Warteschlangenmanager eine Zurücksetzung auf eine Momentaufnahme durchführen. Mit dieser Operation werden die Daten auf den zuletzt bekannten intakten Status zurückgesetzt.

## Vorgehensweise

So beheben Sie einen inkonsistenten Status:

1. Stellen Sie auf dem Wiederherstellungsknoten die sekundäre Instanz auf die primäre Instanz um:

   ```
   rdqmdr -m qmname -p
   ```

   Die Zurücksetzung auf die Momentaufnahme wird gestartet.
2. Überprüfen Sie auf dem Wiederherstellungsknoten den Status des Warteschlangenmanagers, um festzustellen, wann die Zurücksetzung auf die Momentaufnahme abgeschlossen ist:

   ```
   rdqmstatus -m qmname
   ```
3. Wenn der Warteschlangenmanager-Status `Normal` ist, starten Sie den Warteschlangenmanager:

   ```
   strmqm qmname
   ```

*Problem mit Partitionierung (Spaltung) bei DR-RDQM beheben*
Ein Partitionierungs-Problem kann auftreten, wenn beide Warteschlangenmanager eines Disaster-Recovery-Paares gleichzeitig in der primären Rolle ausgeführt werden.

## Informationen zu diesem Vorgang

Wenn Sie die sekundäre Instanz eines Warteschlangenmanagers auf dem Wiederherstellungsknoten hochgestuft haben, während die ursprüngliche primäre Instanz weiterhin auf dem Hauptknoten ausgeführt wurde, sind bei Ihnen effektiv zwei Versionen desselben Warteschlangenmanagers aktiv, die jeweils ihre eigene Ansicht der Warteschlangenmanagerdaten haben. Der DR-Status für den Warteschlangenmanager auf jedem Knoten wird als `Partitioned`gemeldet.

Sie müssen entscheiden, welcher der beiden Warteschlangenmanager über die richtige Ansicht der Daten verfügt, und Sie müssen diesen Satz beibehalten, während Sie den anderen verwerfen. Zum Durchführen dieser Operation verwenden Sie den Befehl **rdqmdr**.

Es gibt zwei Prozeduren. In der ersten wird beschrieben, wie die Daten vom Hauptknoten beibehalten werden, in der zweiten wird beschrieben, wie die Daten vom Wiederherstellungsknoten beibehalten werden.

## Prozedur

- Gehen Sie wie folgt vor, um die Daten von dem Warteschlangenmanager auf dem Hauptknoten beizubehalten:
  a) Stellen Sie sicher, dass beide Warteschlangenmanager-Instanzen gestoppt sind.
  b) Geben Sie an, dass der Warteschlangenmanager auf dem Wiederherstellungsknoten der sekundäre Warteschlangenmanager ist:

  ```
  rdqmdr -m qmname -s
  ```

  c) Geben Sie an, dass der Warteschlangenmanager auf dem Hauptknoten der primäre Warteschlangenmanager ist:

  ```
  rdqmdr -m qmname -p
  ```

  Die Synchronisation beginnt, wobei die Daten von dem Warteschlangenmanager auf dem Hauptknoten auf den Wiederherstellungsknoten kopiert werden.
  d) Überprüfen Sie den Status der Synchronisation:

  ```
  rdqmstatus -m qmname
  ```

  e) Wenn die Synchronisation abgeschlossen ist, starten Sie den Warteschlangenmanager auf dem Hauptknoten:

  ```
  strmqm qmname
  ```

- Gehen Sie wie folgt vor, um die Daten von dem Warteschlangenmanager auf dem Wiederherstellungsknoten beizubehalten:
  a) Stellen Sie sicher, dass beide Warteschlangenmanager-Instanzen gestoppt sind.
  b) Geben Sie an, dass der Warteschlangenmanager auf dem Hauptknoten der sekundäre Warteschlangenmanager ist:

  ```
  rdqmdr -m qmname -s
  ```

  c) Geben Sie an, dass der Warteschlangenmanager auf dem Wiederherstellungsknoten der primäre Warteschlangenmanager ist:

  ```
  rdqmdr -m qmname -p
  ```

  Die Synchronisation beginnt, wobei die Daten von dem Warteschlangenmanager auf dem Wiederherstellungsknoten auf den Hauptknoten kopiert werden.
  d) Überprüfen Sie den Status der Synchronisation:

  ```
  rdqmstatus -m qmname
  ```

  e) Wenn die Synchronisation abgeschlossen ist, stufen Sie den Warteschlangenmanager auf dem Wiederherstellungsknoten herab:

  ```
  rdqmdr -m qmname -s
  ```

  f) Stufen Sie den Warteschlangenmanager auf dem Hauptknoten hoch und starten Sie ihn:

```
rdqmdr -m qmname -p
strmqm qmname
```

*IP-Adressen in Disaster-Recovery-Konfigurationen ändern*
Wenn Sie die IP-Adressen einer der Schnittstellen in einer Disaster-Recovery-Konfiguration ändern, ist zwischen den beiden Knoten keine Replikation mehr möglich.

Wenn Sie die IP-Adressen für die Replikationsschnittstelle einer Ihrer DR-Knoten ändern müssen, gehen Sie dazu folgendermaßen vor:

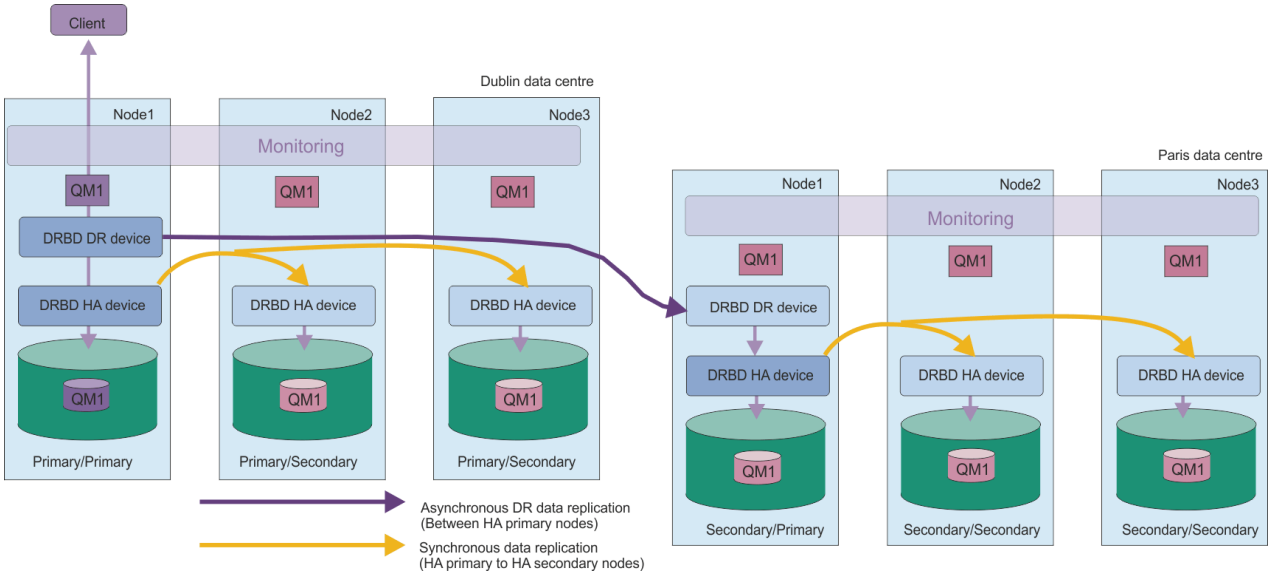
1. Sichern Sie auf dem Primärknoten die DR-Warteschlangenmanager und löschen Sie sie. Löschen Sie auf dem Wiederherstellungsknoten die Warteschlangenmanager. Siehe „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen” auf Seite 723 und „DR RDQM löschen” auf Seite 648.
2. Erstellen Sie die DR-Warteschlangenmanager erneut, wobei Sie die neuen IP-Adressen angeben, und stellen Sie die Sicherungskopien wieder her (siehe „RDQM für die Wiederherstellung nach einem Katastrophenfall erstellen” auf Seite 645 und „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen” auf Seite 723).

## Linux Disaster-Recovery- und Hochverfügbarkeits-RDQM

Sie können einen RDQM (Replicated Data Queue Manager) konfigurieren, der in einer Hochverfügbarkeitsgruppe an einem Standort ausgeführt wird, aber in eine andere Hochverfügbarkeitsgruppe (HA-Gruppe) an einem anderen Standort übernommen werden kann, falls die erste Gruppe nach einem Störfall nicht mehr verfügbar ist. Dieser Warteschlangenmanager wird als DR/HA-RDQM bezeichnet.

Ein DR/HA-RDQM vereint in sich die Funktionen eines Hochverfügbarkeits-RDQM (siehe „RDQM-Hochverfügbarkeit” auf Seite 609) und eines Disaster-Recovery-RDQM (siehe „RDQM-Notfallwiederherstellung” auf Seite 640).

Das folgende Diagramm zeigt ein Beispiel für DR/HA RDQM.




Die Replikation zwischen den DR/HA-RDQMs am Hauptstandort und am Disaster-Recovery-Standort erfolgt immer asynchron. Bei der asynchronen Replikation werden Operationen wie IBM MQ PUT oder GET abgeschlossen und die Steuerung an die Anwendung zurückgegeben, bevor das Ereignis auf den Sekundär-Warteschlangenmanager repliziert wird.

Falls erforderlich, können Sie statt eines Haupt- und eines Wiederherstellungsstandorts ('main' und 'recovery') zwei aktive Standorte einrichten, sodass im normalen Betrieb einige Ihrer DR/HA-RDQMs an dem einen Standort und einige an dem anderen Standort ausgeführt werden. Wenn nach einem Störfall einer der Standorte nicht mehr verfügbar ist, werden alle DR/HA-RDQMs in derselben HA-Gruppe am selben Standort ausgeführt.

Jede HA-Gruppe wird wie eine gewöhnliche HA-Gruppe konfiguriert. Sie können variable IP-Adressen für einen DR/HA-RDQM in jeder HA-Gruppe definieren. Die variable IP-Adresse kann für jede HA-Gruppe dieselbe oder eine andere sein.

Es ist nicht möglich, aus einem vorhandenen RDQM per Upgrade einen DR/HA-RDQM zu machen. Sie müssen einen DR/HA-RDQM erstellen. (Falls erforderlich, können Sie die Daten eines vorhandenen RDQM sichern, ihn löschen, als DR/HA-RDQM neu erstellen und anschließend die Daten wiederherstellen (siehe „IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen" auf Seite 723).

Um DR/HA-RDQMs zu konfigurieren, müssen Sie die folgenden Hauptschritte ausführen:

1. Konfigurieren Sie eine HA-Gruppe am Hauptstandort ('main').
2. Konfigurieren Sie eine HA-Gruppe am Wiederherstellungsstandort ('recovery').
3. Erstellen Sie einen Primär/Primär-DR/HA-RDQM auf einem Knoten der HA-Gruppe am Hauptstandort.
4. Erstellen Sie Primär/Sekundär-DR/HA-RDQMs auf den anderen beiden Knoten am Hauptstandort.
5. Definieren Sie eine variable IP-Adresse für eine Anwendung für den Zugriff auf den DR/HA-RDQM, wenn er auf einem der Knoten der HA-Gruppe am Hauptstandort aktiv ist.
6. Erstellen Sie einen Sekundär/Primär-DR/HA-RDQM auf einem Knoten der HA-Gruppe am Wiederherstellungsstandort.
7. Erstellen Sie Sekundär/Sekundär-DR/HA-RDQMs auf den anderen beiden Knoten am Wiederherstellungsstandort.
8. Definieren Sie eine variable IP-Adresse für eine Anwendung für den Zugriff auf den DR/HA-RDQM, wenn er auf einem der Knoten der HA-Gruppe am Wiederherstellungsstandort aktiv ist.

Details zu jedem dieser Schritte finden Sie in den folgenden Abschnitten.

## Linux *Voraussetzungen für eine DR/HA-RDQM-Lösung*

Die Voraussetzungen für die DR/HA-RDQM-Lösung sind dieselben wie für die HA-RDQM-Lösung und die DR-RDQM-Lösung.

Details zu den Voraussetzungen für die HA-Teile der Konfiguration finden Sie im Abschnitt „Voraussetzungen für RDQM HA-Lösung" auf Seite 611.

Details zum DR-Teil der Konfiguration finden Sie im Abschnitt „Voraussetzungen für RDQM-DR-Lösung" auf Seite 643.

## Linux *HA-Gruppen für DR/HA-RDQMs konfigurieren*

Sie müssen sowohl am Hauptstandort als auch am Wiederherstellungsstandort eine HA-Gruppe (Hochverfügbarkeitsgruppe) erstellen. Wenn es an einem der beiden Standorte bereits eine HA-Gruppe gibt, können Sie DR/HA-RDQMs in dieser HA-Gruppe erstellen. (Der Betrieb vorhandener RDQMs läuft weiter wie bisher.)

Die Vorgehensweise ist dieselbe wie dir für RDQM-Hochverfügbarkeit (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)" auf Seite 615).

Wenn Sie eine Gruppe mit hoher Verfügbarkeit definieren, geben Sie die IP-Adressen an, die für die Überwachung und Replikation von jedem Knoten in der `rdqm.ini`-Datei verwendet werden. Wenn Sie eine HA-Gruppe zur Unterstützung von DR/HA-RDQMs erstellen, können Sie auch die IP-Adressen, die von der HA-Gruppe, die Sie definieren, für die DR-Replikation verwendet werden, und die IP-Adressen, die von den Knoten in der anderen HA-Gruppe des DR-Paars für die DR-Replikation verwendet werden, angeben. (Wenn Sie in der Datei `rdqm.ini` die IP-Adressen der DR-Replikation nicht angeben, können Sie sie in der Befehlszeile angeben, wenn Sie eine DR/HA-RDQM-Datei erstellen.)

Wenn Sie eine vorhandene HA-Gruppe konfigurieren, können Sie der vorhandenen `rdqm.ini`-Datei IP-Adressen für die DR-Replikation hinzufügen. Nachdem Sie `rdqm.ini` aktualisiert haben, müssen Sie **`rdqmadm`** nicht erneut ausführen, aber Sie müssen `rdqm.ini` aktualisieren, bevor Sie DR/HA-RDQMs erstellen.

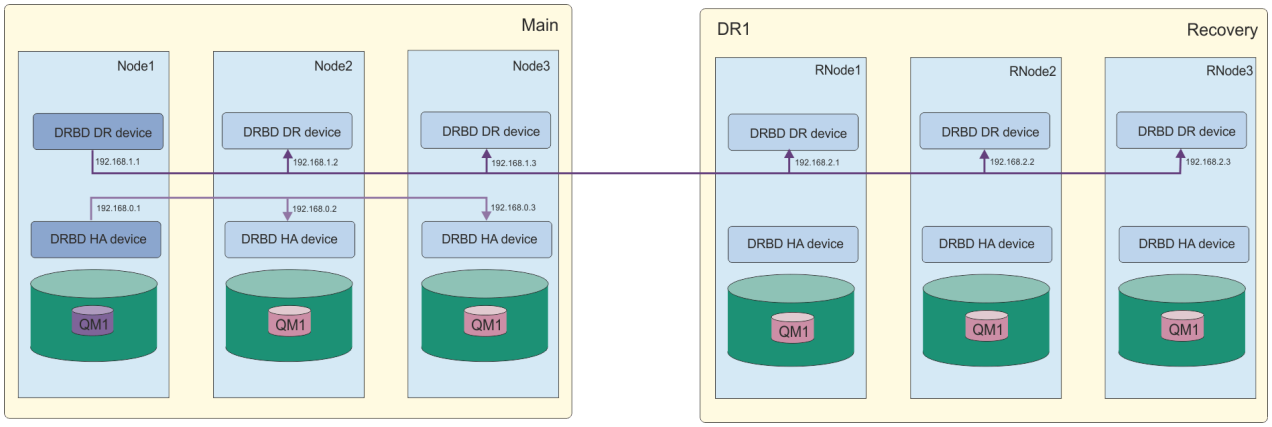
Geben Sie mit dem Attribut DR_Replication in den Node-Zeilengruppen die DR-Replikationsschnittstellen in der HA-Gruppe an, die Sie definieren, z. B.:

```
Node:
  Name=Node1
  HA_Replication=192.168.0.1
  DR_Replication=192.168.1.1
Node:
  Name=Node2
  HA_Replication=192.168.0.2
  DR_Replication=192.168.1.2
Node:
  Name=Node3
  HA_Replication=192.168.0.3
  DR_Replication=192.168.1.3
```

Geben Sie in der Zeilengruppe DRGroup die DR-Replikationsadressen der fernen HA-Gruppe an, z. B.:

```
DRGroup:
  Name=DR1
  DR_Replication=192.168.2.1
  DR_Replication=192.168.2.2
  DR_Replication=192.168.2.3
```

Das folgende Diagramm veranschaulicht diese Konfiguration:




Wenn Sie keine IP-Adressen für die DR-Replikation für die Knoten in der lokalen HA-Gruppe angeben, entweder in der Datei rdqm.ini oder in der Befehlszeile beim Erstellen eines DR/HA-RDQM, dann werden die HA_Replication-Schnittstellen, die für jeden Knoten definiert sind, für die DR-Replikation verwendet. Sie müssen DR-Replikationsadressen der fernen HA-Gruppe entweder in der Datei rdqm.ini oder in der **crtmqm**-Befehlszeile angeben.

## Linux *DR/HA-RDQMs erstellen*

Verwenden Sie den Befehl **crtmqm**, um einen RDQM (Replicated Data Queue Manager) in einer DR/HA-Konfiguration zu erstellen.

### Informationen zu diesem Vorgang

Sie können einen DR/HA-RDQM als Benutzer in der mqm-Gruppe erstellen, wenn der Benutzer 'sudo' verwenden kann. Andernfalls müssen Sie den RDQM als Root erstellen.

Sie müssen eine Reihe von DR/HA-RDQMs erstellen:

- In der HA-Gruppe am Hauptstandort ('main'):
  - Erstellen Sie auf dem Knoten, auf dem der Warteschlangenmanager unter normalen Bedingungen ausgeführt werden soll, den Primär/Primär-DR/HA-RDQM.
  - Erstellen Sie auf jedem der anderen beiden Knoten in der HA-Gruppe einen Primär/Sekundär-DR/HA-RDQM.
- In der HA-Gruppe am Wiederherstellungsstandort ('recovery'):

– Erstellen Sie auf dem Knoten, auf dem der Warteschlangenmanager nach einer Übernahme ausgeführt wird, den Sekundär/Primär-DR/HA-RDQM. Sie können die Befehlsausgabe verwenden, die bei der Erstellung des Primär/Primär-Warteschlangenmanagers am Hauptstandort erfolgte.
– Erstellen Sie auf jedem der anderen beiden Knoten in der HA-Gruppe einen Sekundär/Sekundär-DR/HA-RDQM.

Alle Warteschlangenmanagerinstanzen müssen denselben Namen haben und es muss ihnen der gleiche Speicherplatz zugeordnet werden.

Die folgenden Punkte enthalten Anleitungen zur Dimensionierung des Dateisystems des Warteschlangenmanagers:

1. Wenn Sie einen RDQM-Warteschlangenmanager erstellen, wird ein Dateisystem zum Speichern von Warteschlangenmanagerdaten und -protokollen zugeordnet. Es ist wichtig, die Größe dieses Dateisystems entsprechend zu ändern, damit der Warteschlangenmanager fortlaufende Aktivitäten in seinen Protokollen aufzeichnen und Anwendungsnachrichten in Warteschlangen speichern kann. Berücksichtigen Sie bei der Dimensionierung des Dateisystems die Anforderungen für Spitzennachrichten, das künftige Workloadwachstum und Anwendungsausfälle, die dazu führen können, dass Nachrichten in Warteschlangen erstellt werden. Informationen zur Berechnung der Größe des Wiederherstellungsprotokolls des Warteschlangenmanagers finden Sie unter „Wie groß sollte ich mein Protokolldateisystem machen?" auf Seite 703. Bei der Berechnung des Speicherbedarfs für Anwendungsnachrichten müssen die Größe und Anzahl der Nachrichten sowie deren MQMD-Header und Nachrichteneigenschaften berücksichtigt werden.
2. Die Größe von Dateisystemen des RDQM-Warteschlangenmanagers kann nicht dynamisch geändert werden. Wenn dies erforderlich ist, müssen Sie einen RDQM-Warteschlangenmanager mit einem größeren Dateisystem sichern und wiederherstellen (siehe „Größe des Dateisystems für einen HA-RDQM-Warteschlangenmanager ändern" auf Seite 625).
3. Sie können die Größe einzelner Warteschlangen auf Platte begrenzen, indem Sie lokale Warteschlangenattribute wie MAXDEPTH und MAXFSIZE verwenden. Siehe Warteschlangendateien von IBM MQ ändern.
4. Sie sollten Ihre laufende Plattenbelegung überwachen und entsprechend reagieren, wenn die Plattenbelegung zunimmt, bevor die Dateisystembelegung kritisch wird. Die Dateisystemnutzung kann mithilfe von Plattform-/Betriebssystemfunktionen überwacht werden oder indem Metriken abonniert werden, die in IBM MQ -Systemthemen veröffentlicht werden, die unter In den Systemthemen veröffentlichte Metrikenbeschrieben sind.

## Prozedur

- Gehen Sie wie folgt vor, um den Primär/Primär-DR/HA-RDQM zu erstellen:
  a) Geben Sie den folgenden Befehl ein:

```
crtmqm -sx -rr p
              [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
              (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
              -rp DRPort
              [-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]
              [-g ApplicationGroup] [-oa user|group]
              [-t TrigInt] [-u DeadQ] [-x MaxUMsgs]
              [-lp LogPri] [-ls LogSec]
              [-lc | -ll | -lla | -lln] [-lf LogFileSize]
              [-p Port] [-fs FilesystemSize] QMgrName
```

  Dabei gilt Folgendes:

  **-sx**
  Gibt 'Primär' als ursprüngliche HA-Rolle an.

  **-rr p**
  Gibt 'Primär' als ursprüngliche DR-Rolle an.

**-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3***
Geben Sie optional die IP-Adressen der DR-Schnittstellen auf den drei Knoten am lokalen Standort (d. h. am Hauptstandort 'main') an. Wenn nicht angegeben, werden die in der `rdqm.ini`-Datei angegebenen IP-Adressen verwendet.

**-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3***
Geben Sie die IP-Adressen der DR-Schnittstellen auf den drei Knoten am fernen Standort (d. h. am Wiederherstellungsstandort 'recovery') an. Sie müssen entweder diesen Parameter oder den Parameter `-rn` angeben.

**-rn *GroupName***
Geben Sie den Namen der remote HA-Gruppe wie in der `rdqm.ini`-Datei angegeben an. Sie müssen `-ri` oder `-rn` angeben.

**-rp *Port***
Gibt den Port an, der für die DR-Replikation verwendet werden soll

***other_crtmqm_options***
Sie können optional eine oder mehrere der folgenden allgemeinen **`crtmqm`** Optionen angeben:

- -z
- -q
- -c *Text*
- -d *DefaultTransmissionQueue*
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa user | group
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -lc | -l
- -lla | -lln
- -lf *LogFileSize*
- -p *Port*

**-fs *size***
Gibt optional die Größe des Dateisystems an, das für den Warteschlangenmanager erstellt werden soll, d. h. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird. Ein weiteres logisches Volumen dieser Größe wird erstellt, um die Zurücksetzung auf die Momentaufnahmeoperation zu unterstützen, so dass der Gesamtspeicher für den DR RDQM etwas mehr als doppelt so groß ist wie hier angegeben.

*Größe* ist ein numerischer Wert, der in GB angegeben wird. Sie können einen Wert in MB angeben, indem Sie den Wert gefolgt vom Zeichen Meingeben. Geben Sie beispielsweise 3 ein, um eine Dateisystemgröße von 3 GB anzugeben. Geben Sie 1024Mein, um eine Dateisystemgröße von 1024 MB anzugeben. (Sie können auch ein G-Suffix hinzufügen, um explizit GB anzugeben.)

***QMNAME***
Gibt den Namen des replizierten Datenwarteschlangenmanagers an. Bei dem Namen muss die Groß-/Kleinschreibung beachtet werden

Nach Beendigung des Befehls wird der Befehl ausgegeben, den Sie am Wiederherstellungsstandort eingeben können, um die Sekundär/Primär-Instanz des Warteschlangenmanagers zu erstellen.

- Gehen Sie wie folgt vor, um einen Primär/Sekundär-DR/HA-RDQM auf den anderen beiden Knoten in der HA-Gruppe zu erstellen:

a) Geben Sie auf jedem Knoten folgenden Befehl ein:

```
crtmqm -sxs -rr p
             [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
             (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
             -rp DRPort
             [-fs FilesystemSize] QMgrName
```

Dabei gilt Folgendes:

**-sxs**
Gibt 'Sekundär' als ursprüngliche HA-Rolle an.

**-rr p**
Gibt 'Primär' als ursprüngliche DR-Rolle an.

**-rl *DRLocalIP1*, *DRLocalIP2*, *DRLocalIP3***
Geben Sie optional die IP-Adressen der DR-Schnittstellen auf den drei Knoten am lokalen Standort (d. h. am Hauptstandort 'main') an. Wenn nicht angegeben, werden die in der `rdqm.ini`-Datei angegebenen IP-Adressen verwendet.

**-ri *DRRemoteIP1*, *DRRemoteIP2*, *DRRemoteIP3***
Geben Sie die IP-Adressen der DR-Schnittstellen auf den drei Knoten am fernen Standort (d. h. am Wiederherstellungsstandort 'recovery') an. Sie müssen entweder diesen Parameter oder den Parameter `-rn` angeben.

**-rn *GroupName***
Geben Sie den Namen der remote HA-Gruppe wie in der `rdqm.ini`-Datei angegeben an. Sie müssen `-ri` oder `-rn` angeben.

**-rp *Port***
Gibt den Port an, der für die DR-Replikation verwendet werden soll

**-fs *size***
Gibt die Größe des Dateisystems an, das für den Warteschlangenmanager erstellt werden soll, d. h. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird. Wenn Sie bei der Erstellung des Primär/Primär-RDQM eine andere als die Standardgröße angegeben haben, müssen Sie hier denselben Wert angeben.

*Größe* ist ein numerischer Wert, der in GB angegeben wird. Sie können einen Wert in MB angeben, indem Sie den Wert gefolgt vom Zeichen Meingeben. Geben Sie beispielsweise 3 ein, um eine Dateisystemgröße von 3 GB anzugeben. Geben Sie 1024Mein, um eine Dateisystemgröße von 1024 MB anzugeben. (Sie können auch ein G-Suffix hinzufügen, um explizit GB anzugeben.)

***QMNAME***
Gibt den Namen des Primär/Sekundär-RDQM an. Dieser muss mit dem Namen identisch sein, den Sie für die Primär/Primär-Instanz des RDQM angegeben haben. Beachten Sie, dass bei dem Namen die Groß-/Kleinschreibung beachtet werden

- Gehen Sie wie folgt vor, um einen Sekundär/Primär-DR/HA-RDQM auf dem Knoten zu erstellen, auf dem der Warteschlangenmanager nach einer Übernahme ausgeführt wird:

  a) Verwenden Sie die Befehlsausgabe, die bei der Erstellung des Primär/Primär-DR/HA-RDQM am Hauptstandort erfolgte, oder geben Sie folgenden Befehl ein:

```
crtmqm -sx -rr s
             [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
             (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
             -rp DRPort
             [-fs FilesystemSize] QMgrName
```

**-sx**
Gibt 'Primär' als ursprüngliche HA-Rolle an.

**-rr s**
Gibt 'Sekundär' als ursprüngliche DR-Rolle an.

**-rl *DRLocalIP1, DRLocalIP2, DRLocalIP3***
Geben Sie optional die IP-Adressen der DR-Schnittstellen auf den drei Knoten am lokalen Standort (d. h. am Wiederherstellungsstandort 'recovery') an. Wenn nicht angegeben, werden die in der `rdqm.ini`-Datei angegebenen IP-Adressen verwendet.

**-ri *DRRemoteIP1, DRRemoteIP2, DRRemoteIP3***
Geben Sie die IP-Adressen der DR-Schnittstellen auf den drei Knoten am fernen Standort (d. h. am Hauptstandort 'main') an. Sie müssen entweder diesen Parameter oder den Parameter `-rn` angeben.

**-rn *GroupName***
Geben Sie den Namen der remote HA-Gruppe wie in der `rdqm.ini`-Datei angegeben an. Sie müssen `-ri` oder `-rn` angeben.

**-rp *Port***
Gibt den Port an, der für die DR-Replikation verwendet werden soll

**-fs *size***
Gibt optional die Größe des Dateisystems an, das für den Warteschlangenmanager erstellt werden soll, d. h. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird. Ein weiteres logisches Volumen dieser Größe wird erstellt, um die Zurücksetzung auf die Momentaufnahmeoperation zu unterstützen, so dass der Gesamtspeicher für den DR RDQM etwas mehr als doppelt so groß ist wie hier angegeben.

***QMNAME***
Gibt den Namen des replizierten Datenwarteschlangenmanagers an. Bei dem Namen muss die Groß-/Kleinschreibung beachtet werden

- Gehen Sie wie folgt vor, um einen Sekundär/Sekundär-DR/HA-RDQM auf den anderen beiden Knoten am Wiederherstellungsstandort zu erstellen:

  a) Geben Sie auf jedem Knoten folgenden Befehl ein:

```
crtmqm -sxs -rr s
             [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
             (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
             -rp DRPort
             [-fs FilesystemSize] QMgrName
```

**-sxs**
Gibt 'Primär' als ursprüngliche HA-Rolle an.

**-rr s**
Gibt 'Sekundär' als ursprüngliche DR-Rolle an.

**-rl *DRLocalIP1, DRLocalIP2, DRLocalIP3***
Geben Sie optional die IP-Adressen der DR-Schnittstellen auf den drei Knoten am lokalen Standort an. Wenn nicht angegeben, werden die in der `rdqm.ini`-Datei angegebenen IP-Adressen verwendet.

**-ri *DRRemoteIP1, DRRemoteIP2, DRRemoteIP3***
Geben Sie die IP-Adressen der DR-Schnittstellen auf den drei Knoten am fernen Standort an. Sie müssen entweder diesen Parameter oder den Parameter `-rn` angeben.

**-rn *GroupName***
Geben Sie den Namen der remote HA-Gruppe wie in der `rdqm.ini`-Datei angegeben an. Sie müssen `-ri` oder `-rn` angeben.

**-rp *Port***
Gibt den Port an, der für die DR-Replikation verwendet werden soll

**-fs *size***
Gibt optional die Größe des Dateisystems an, das für den Warteschlangenmanager erstellt werden soll, d. h. die Größe des logischen Datenträgers, der in der Datenträgergruppe drbdpool erstellt wird. Ein weiteres logisches Volumen dieser Größe wird erstellt, um die Zurücksetzung auf die Momentaufnahmeoperation zu unterstützen, so dass der Gesamtspeicher für den DR RDQM etwas mehr als doppelt so groß ist wie hier angegeben.

***QMNAME***
Gibt den Namen des replizierten Datenwarteschlangenmanagers an. Bei dem Namen muss die Groß-/Kleinschreibung beachtet werden

**Anmerkung:** Wenn Sie eine RDQM erstellen, wird die nächste freie Portnummer über 7000 für die HA-Replikationsverbindung zugeordnet. Wenn festgestellt wird, dass der ausgewählte Port von einer anderen Anwendung verwendet wird, schlägt der Befehl **crtmqm** mit der Fehlermeldung AMQ6543 fehl und dieser Port wird einer Ausschlussliste hinzugefügt. Sie müssen die sekundären Instanzen des Warteschlangenmanagers löschen und den Befehl **crtmqm** anschließend erneut ausführen.

## Nächste Schritte

Nachdem Sie alle DR/HA-RDQMs erstellt haben, müssen Sie anhand des Status der Primär/Primär- und der Sekundär/Primär-Instanz überprüfen, ob alle Instanzen korrekt sind. Führen Sie auf den Knoten den Befehl **rdqmstatus** aus. Die Knoten sollten den normalen Status wie in „Status des DR/HA-RDQM und der HA-Gruppe anzeigen" auf Seite 676 beschrieben anzeigen. Wenn sie diesen Status nicht anzeigen, löschen Sie die Sekundär/Primär-Instanz und erstellen Sie sie erneut. Achten Sie dabei darauf, die richtigen Argumente anzugeben.

**Zugehörige Tasks**

„DR/HA-RDQMs erstellen" auf Seite 669
Verwenden Sie den Befehl **crtmqm**, um einen RDQM (Replicated Data Queue Manager) in einer DR/HA-Konfiguration zu erstellen.

**Zugehörige Verweise**

crtmqm

Linux *Löschen eines DR/HA-RDQM*

Verwenden Sie zum Löschen eines DR/HA-RDQM (Replicated Data Queue Manager) den Befehl **dltmqm**.

## Informationen zu diesem Vorgang

Sie müssen den Befehl zum Löschen des RDQM sowohl auf dem Primär/Primär-Knoten als auch auf dem Sekundär/Primär-Knoten ausführen. RDQM muss zuerst beendet werden. Sie können den Befehl als mqm-Benutzer ausführen, wenn dieser Benutzer über die erforderlichen Zugriffsrechte für "sudo" verfügt. Andernfalls müssen Sie den Befehl als Root ausführen.

## Prozedur

- Geben Sie folgenden Befehl ein, um einen DR/HA-RDQM zu löschen:

```
dltmqm RDQM_name
```

**Zugehörige Verweise**

dltmqm

Linux ***Variable IP-Adresse erstellen***

Sie können variable IP-Adressen für jede Ihrer HA-Gruppen in einer DR/HA-RDQM-Konfiguration erstellen.

Eine variable IP-Adresse ermöglicht einem Client die Verwendung derselben IP-Adresse für einen DR/HA-RDQM, unabhängig davon, welcher Knoten in einer HA-Gruppe ausgeführt wird. Wenn die Anwendungskonnektivität Ihrer beiden HA-Gruppen auf privaten/isolierten Netzen basiert, kann für beide Gruppen dieselbe variable IP-Adresse definiert werden. Sie müssen die betreffende variable IP-Adresse zwar immer noch zweimal definieren, aber nur einmal in jeder HA-Gruppe.

Variable IP-Adressen werden auf dieselbe Weise erstellt und gelöscht wie ein HA-RDQM. Weitere Informationen finden Sie unter „Variable IP-Adresse erstellen und löschen" auf Seite 628.

## Linux *Starten, Stoppen und Anzeigen des Status eines DR/HA-RDQM*

Zum Starten und Stoppen eines DR/HA-RDQM sowie zum Anzeigen seines aktuellen Status werden Varianten der standardmäßigen IBM MQ-Steuerbefehle verwendet.

### Informationen zu diesem Vorgang

Sie müssen die Befehle zum Starten und Stoppen eines DR/HA-RDQM und zum Anzeigen seines aktuellen Status als ein Benutzer ausführen, der sowohl zur Gruppe `mqm` als auch zur Gruppe `haclient` gehört.

Sie müssen die Befehle ausführen, um einen Warteschlangenmanager auf dem Primärknoten für diesen Warteschlangenmanager zu starten und zu stoppen.

### Prozedur

- Geben Sie zum Starten eines RDQM folgenden Befehl auf dem Primärknoten des RDQM ein:

  ```
  strmqm qmname
  ```

  Dabei steht *qmname* für den Namen des DR/HA-RDQM, der gestartet werden soll.

  RDQM wird gestartet, und Pacemaker beginnt mit der Verwaltung des RDQM. Sie müssen die Option `-ns` mit `strmqm` angeben, wenn Sie andere `strmqm` -Optionen angeben möchten.
- Geben Sie zum Stoppen eines RDQM folgenden Befehl auf dem Primärknoten des DR/HA-RDQM ein:

  ```
  endmqm qmname
  ```

  Hierbei steht *qmname* für den Namen des RDQM, den Sie stoppen wollen.

  Der Pacemaker wird für die Verwaltung von RDQM eingestellt, und der RDQM wird beendet. Alle anderen **endmqm** -Parameter können beim Stoppen eines RDQM verwendet werden.
- Geben Sie den folgenden Befehl ein, um den Status eines RDQM anzuzeigen:

  ```
  dspmq -m QMname
  ```

  Die Statusinformationen, die ausgegeben werden, hängen davon ab, ob Sie den Befehl auf dem Primär- oder Sekundärknoten des RDQM ausführen. Wenn die Ausführung auf dem Primärknoten erfolgt, wird eine der normalen Statusnachrichten angezeigt, die von **dspmq** zurückgegeben werden. Wenn Sie den Befehl auf einem Sekundärknoten ausführen, wird der Status `Ended immediately` angezeigt. Wird **dspmq** beispielsweise auf dem Knoten RDQM7 ausgeführt wird, werden möglicherweise folgende Informationen zurückgegeben:

  ```
  QMNAME(DRQM8)                           STATUS(Ended immediately)
  QMNAME(DRQM7)                           STATUS(Running)
  ```

  Sie können Argumente mit `dspmq` verwenden, um festzustellen, ob ein RDQM für Disaster-Recovery konfiguriert ist und ob es sich aktuell um die Primär- oder Sekundärinstanz handelt:

  ```
  dspmq -m QMname -o (dr | DR)
  ```

  Eine der folgenden Antworten wird angezeigt:

  **DRROLE()**
  Gibt an, dass der Warteschlangenmanager nicht für Disaster-Recovery konfiguriert ist.

  **DRROLE(Primary)**
  Gibt an, dass der Warteschlangenmanager als der DR-Primärmanager konfiguriert ist.

  **DRROLE(Secondary)**
  Gibt an, dass der Warteschlangenmanager als der DR-Sekundärmanager konfiguriert ist.

Verwenden Sie den Befehl **dspmq -o all**, um die Disaster-Recovery- und Hochverfügbarkeitsinformationen für DR/HA-RDQMs anzuzeigen. Wenn Sie **dspmq -o all** beispielsweise auf dem Knoten ausführen, auf dem der DR/HA-RDQM aktiv ist, werden folgende Statusinformationen angezeigt:

```
QMNAME(TESTQM1)                                           STATUS(Running) HA(Replicated)
DRROLE(Primary)
```

**Zugehörige Verweise**

dspmq (Warteschlangenmanager anzeigen)
endmqm (Warteschlangenmanager beenden)
strmqm (Warteschlangenmanager starten)

### V 9.3.0 *Fehlgeschlagene Ressourcenaktionen in DR/HA-Konfigurationen*

Fehlgeschlagene Ressourcenaktionen treten auf, wenn die Pacemaker-Komponente einer RDQM-Hochverfügbarkeitskonfiguration Probleme mit einer Ressource auf einem der Knoten in einer HA-Gruppe ermittelt.

Fehlgeschlagene Ressourcenaktion können in jeder HA-Konfiguration in einer RDQM-DR/HA-Konfiguration auftreten. Mit dem Befehl **rdqmstatus** können Sie fehlgeschlagene Ressourcenaktionen anzeigen und sie mit dem Befehl **rdqmclean** löschen (nachdem die Ursache des Fehlers beseitigt wurde). Der Prozess entspricht dem für RDQM-HA-Konfigurationen ohne die DR-Komponente. Weitere Informationen finden Sie in „Fehlgeschlagene Ressourcenaktionen” auf Seite 631.

**Zugehörige Tasks**

„Status des DR/HA-RDQM und der HA-Gruppe anzeigen” auf Seite 676
Der HA-Status und die DR-Rolle von DR/HA-RDQMs (Replicated Data Queue Managers) können angezeigt werden.

„RDQM-und HA-Gruppenstatus anzeigen” auf Seite 632
Sie können den Status der HA-Gruppe und von einzelnen replizierten Datenwarteschlangenmanagern (RDQMs) anzeigen.

**Zugehörige Verweise**

rdqmclean
rdqmstatus

### Linux *Status des DR/HA-RDQM und der HA-Gruppe anzeigen*

Der HA-Status und die DR-Rolle von DR/HA-RDQMs (Replicated Data Queue Managers) können angezeigt werden.

## Informationen zu diesem Vorgang

Verwenden Sie den Befehl **rdqmstatus**, um den Status einzelner RDQMs anzuzeigen oder eine Übersicht über den Status aller RDQMs, die der HA-Gruppe bekannt sind, abzurufen.

V 9.3.0 Der Zusammenfassungsstatus für einen Knoten zeigt auch Informationen über das DRBD-Kernelmodul an, auf das RDQM angewiesen ist. Wenn Sie RDQM aktualisieren, ist es wichtig, sicherzustellen, dass die korrekte Version des DRBD Kernelmoduls installiert ist für die Version des RHEL Kernels, der auf dem System läuft. Der Status zeigt die Version des Betriebssystem-Kernels, die Kernelversion, für die das DRBD-Modul erstellt wurde, die DRBD-Version und den Status des DRBD-Kernelmoduls an.

**Anmerkung:** Beachten Sie, dass die DR-Konfiguration in einer HA/DR-Konfiguration immer asynchrone Replikation verwendet, während die HA-Konfiguration immer synchrone Replikation verwendet. Diese Werte werden in der Ausgabe des Befehls `rdqmstatus -m qmgr` in einer kombinierten HA/DR-Konfiguration nicht angezeigt.

Sie müssen ein Benutzer in den Gruppen `mqm` und `haclient` sein, um den Befehl **rdqmstatus** ausführen zu können. Sie können den Befehl auf jedem der Knoten in einer der HA-Gruppen ausführen.

## Prozedur

- Gehen Sie wie folgt vor, um den Zusammenfassungsstatus eines Knotens und der RDQMs anzuzeigen, die Teil der HA-Konfiguration sind:

```
rdqmstatus
```

Es werden die Identität des Knotens, auf dem Sie den Befehl ausgeführt haben, und der Status der RDQMs in der HA-Konfiguration sowie deren aktuelle DR-Rolle angezeigt, z. B.:

```
Node:                                 main-alice
OS kernel version:                    3.10.0-1160.15.2
DRBD OS kernel version:               3.10.0-1160
DRBD version:                         9.1.1
DRBD kernel module status:            Loaded

Queue manager name:                   RDQM1
Queue manager status:                 Running elsewhere
HA current location:                  main-charlie
HA preferred location:                main-charlie
HA blocked location:                  None

Queue manager name:                   RDQM9
Queue manager status:                 Running elsewhere
HA current location:                  main-bob
HA preferred location:                main-bob
HA blocked location:                  None
DR role:                              Primary

Queue manager name:                   RDQM7
Queue manager status:                 Running
HA current location:                  This node
HA preferred location:                This node
HA blocked location:                  None
DR role:                              Primary
```

In diesem Beispiel handelt es sich bei RDQM7 und RDQM8 um DR/HA-RDQMs, während RDQM1 ein HA-RDQM ist, der nicht so konfiguriert ist, dass er auf einen Disaster-Recovery-Standort umschalten kann.

V 9.3.0 Der Status des DRBD-Kernelmoduls ist einer der folgenden Werte:

**Geladen**
Gibt an, dass das DRBD-Modul geladen wurde.

**Teilweise geladen**
Kann auftreten, wenn das DRBD-Modul geladen wurde, aber aufgrund einer Abweichung nicht ordnungsgemäß funktioniert.

**Nicht geladen**
Das DRBD-Modul ist nicht geladen. Diese Option kann auf einer neu installierten Konfiguration angezeigt werden, wenn noch keine RDQM-Warteschlangenmanager erstellt wurden.

**Nicht installiert**
Gibt an, dass das DRBD-Modul nicht installiert ist. oder dass IBM MQ die Betriebssystem-Kernelversion des DRBD-Moduls nicht bestimmen konnte.

**Zuvor installierte Version ist noch geladen**
Dieser Status kann auftreten, wenn ein neues DRBD-Modul installiert wird, während das vorhandene DRBD-Modul ausgeführt wird (d. h. ein RDQM-Warteschlangenmanager wird ausgeführt). Das neu installierte Modul wird im Status gemeldet, ist aber nicht das Modul, das tatsächlich ausgeführt wird.

- Geben Sie folgenden Befehl ein, um den Status eines bestimmten Warteschlangenmanagers auf allen Knoten in der HA-Gruppe anzuzeigen:

```
rdqmstatus -m qmname
```

Dabei steht *qmname* für den Namen des RDQM, dessen Status angezeigt werden soll. Der Status des RDQM-Knotens auf dem aktuellen Knoten wird angezeigt, gefolgt von einer Zusammenfassung des Status der anderen beiden Knoten aus der Perspektive des aktuellen Knotens.

- V 9.3.0

  Geben Sie folgenden Befehl ein, um den Status eines bestimmten Warteschlangenmanagers auf allen Knoten in der HA-Gruppe einschließlich der Einzelheiten zu fehlgeschlagenen Ressourcenaktionen anzuzeigen:

  ```
  rdqmstatus -m qmname -a
  ```

  Dabei steht *qmname* für den Namen des RDQM, dessen Status angezeigt werden soll. Der Status des RDQM-Knotens auf dem aktuellen Knoten wird angezeigt, gefolgt von einer Zusammenfassung des Status der anderen beiden Knoten aus der Perspektive des aktuellen Knotens. Anschließend werden die Einzelheiten zu fehlgeschlagenen Ressourcenaktionen angezeigt, die dem RDQM zugeordnet sind.

  In der folgenden Tabelle sind die Informationen zum aktuellen Knoten zusammengefasst, die vom `rdqmstatus -m qmname`-Befehl für einen RDQM zurückgegeben werden können.

*Tabelle 35. Aktueller Knotenstatus*

| **Statusattribut** | **Mögliche Werte** | **Wird wann angezeigt** |
|---|---|---|
| Knotenname | *Knotenname* | Immer |
| Status des Warteschlangenmanagers | Status des Warteschlangenmanagers (ein Status, der für den Befehl **dspmq** gültig ist) | Immer |
| CPU | *n.nn*% | Nur wenn RDQM auf diesem Knoten aktiv ist |
| Hauptspeicher | *nnn*MB used | Nur wenn RDQM auf diesem Knoten aktiv ist |
| Queue manager file system | *nnn*MB used, *y.y*GB allocated [*z*%] | Nur wenn RDQM auf diesem Knoten aktiv ist |
| HA-Rolle | Primär<br>Sekundär<br>Unbekannt | Immer |
| HA status | Alle Knoten im Standby-Modus<br>This node in standby<br>Remote nodes in standby<br>Gemischt | Alle Knoten im Standby-Modus<br>Aktueller Knoten im Standby-Modus<br>Beide fernen Knoten im Standby-Modus<br>Anderer Status für jeden fernen Knoten |
| HA control | Enabled<br>Inaktiviert<br>Unbekannt | Immer. Zeigt an, ob RDQM von Pacemaker gesteuert wird. |
| HA preferred location | --<br>This node<br>Unbekannt<br>*Knotenname* | Immer |

| *Tabelle 35. Aktueller Knotenstatus (Forts.)* | | |
|---|---|---|
| **Statusattribut** | **Mögliche Werte** | **Wird wann angezeigt** |
| V 9.3.0 V 9.3.0 HA blocked location | None - Die Ausführung des Warteschlangenmanagers auf einem anderen Knoten ist nicht blockiert.<br>This node - Die Ausführung des Warteschlangenmanagers auf dem aktuellen Knoten ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen blockiert.<br>*Knotenname* - Die Ausführung des Warteschlangenmanagers auf *Knotenname* ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen blockiert.<br>*Knotenname1*, *Knotenname2* - Die Ausführung des Warteschlangenmanagers auf *Knotenname1* und *Knotenname2* ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen blockiert.<br>All nodes - Die Ausführung des Warteschlangenmanagers ist aufgrund einer oder mehrerer fehlgeschlagener Ressourcenaktionen auf allen Knoten blockiert. | Immer |
| HA floating IP interface | *Schnittstellenname* | Immer |
| HA floating IP address | *IPV4_address* | Immer |
| DR role | Primär<br>Sekundär<br>Sekundär anstehend<br>Unbekannt | Immer |

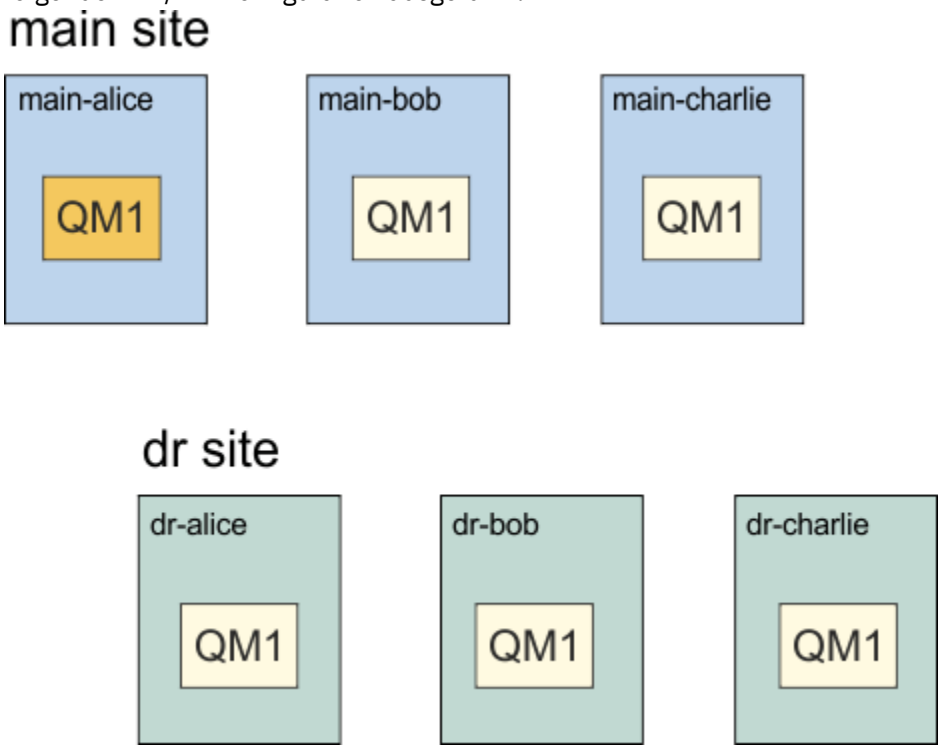
*Tabelle 35. Aktueller Knotenstatus (Forts.)*

| **Statusattribut** | **Mögliche Werte** | **Wird wann angezeigt** |
|---|---|---|
| DR status | Normal<br>Synchronization in progress<br>Partitioniert<br><br><br><br>Fernes System nicht verfügbar<br><br>Inkonsistent<br><br>Zurücksetzen auf Momentaufnahme<br><br><br><br>Das ferne System wurde nicht konfiguriert.<br><br>Negotiation failed | Alles in Ordnung.<br>Synchronisation läuft.<br>Der Benutzer hat die Warteschlange<br>auf jedem Knoten gestartet, während<br>das DR-Replikationsnetz nicht verfügbar war.<br>Die Verbindung zum anderen Knoten<br>ist verloren gegangen.<br>Eine Synchronisation lief, wurde aber unterbrochen.<br>Der Benutzer hat ausgewählt, dass er zurückkehren will zu<br>der Momentaufnahme, die ausgeführt wurde, als<br>der Warteschlangenmanager in den inkonsistenten Status übergetreten ist.<br>Der Primärknoten wurde konfiguriert,<br>aber der Sekundärknoten nicht.<br>Die Anfangsvereinbarung zwischen Primär- und Sekundärknoten ist fehlgeschlagen. Mögliche Ursache: Inkompatible Replikationstypen oder der Sekundärknoten ist mit einer kleineren Dateisystemgröße konfiguriert. |
| DR status (auf HA-Sekundärknoten) | See *HA-Primärknoten* | Wird auf den HA-Sekundärknoten angezeigt, weil der DR-Status nur auf dem HA-Primärknoten bekannt ist. |
| DR port | Der TCP/IP-Port für die Replikation der Daten für diesen Warteschlangenmanager. | Immer. |
| DR local IP address | Die lokale IP-Adresse, die dieser Warteschlangenmanager für die DR-Replikation verwendet. | Immer. |
| DR remote IP address list | Die fernen IP-Adressen, die dieser Warteschlangenmanager für die DR-Replikation verwendet. Eine durch Kommas getrennte Liste mit drei IP-Adressen. | Immer. |
| DR current remote IP address | Die aktuelle ferne IP-Adresse, mit der dieser Warteschlangenmanager für die DR-Replikation verbunden wird. | Für einen HA-Primärknoten mit einer aktiven DR-Verbindung. |

| *Tabelle 35. Aktueller Knotenstatus (Forts.)* | | |
|---|---|---|
| **Statusattribut** | **Mögliche Werte** | **Wird wann angezeigt** |
| DR current remote IP address (auf HA-Sekundärknoten) | See *HA-Primärknoten* | Wird auf einem HA-Sekundärknoten angezeigt, da die DR-Verbindung nur auf dem HA-Primärknoten vorhanden ist. |
| DR out of sync data | *x*KB | Wird angezeigt, wenn der ferne Knoten nicht verfügbar oder inkonsistent ist. |
| DR synchronization progress | *y*% | Wird angezeigt, wenn eine Synchronisation läuft. |
| DR estimated time to completion | *jjjj-mm-tt hh:mm:ss* | Wird angezeigt, wenn eine Synchronisation läuft. |
| Snapshot reversion progress | *y*% | Wird angezeigt, wenn der DR-Status "Reverting to snapshot" ist. |
| V 9.3.0 V 9.3.0 DR last in sync | *jjjj-mm-tt hh:mm:ss* | Wird angezeigt, wenn die DR-Daten nicht synchron sind (nach der Erstsynchronisation). Gibt Zeit und Datum an, als die Daten zuletzt synchron waren. |

**Beispiel**

Diese Beispiele veranschaulichen den Befehl **`rdqmstatus -m qm1`** wird auf verschiedenen Knoten der folgenden DR/HA-Konfiguration ausgeführt:




Beispiel für Status 'Normal' auf einem Knoten, der der DR-Primärknoten und der HA-Primärknoten ist:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
```

```
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

Beispiel für Status 'Normal' auf einem Knoten, der der DR-Primärknoten und ein HA-Sekundärknoten ist:

```
Node:                                   main-bob
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    main-alice
HA preferred location:                  main-alice
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              See main-alice
DR port:                                3000
DR local IP address:                    192.168.1.2
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           See main-alice

Node:                                   main-alice
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

Beispiel für Status 'Normal' auf einem Knoten, der der DR-Sekundärknoten und ein HA-Primärknoten ist:

```
Node:                                   dr-alice
Queue manager status:                   Ended immediately
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Secondary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.2.1
DR remote IP address list:              192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address:           192.168.1.1

Node:                                   dr-bob
HA status:                              Normal

Node:                                   dr-charlie
HA status:                              Normal
```

Beispiel für Status 'Normal' auf einem Knoten, der der DR-Sekundärknoten und ein HA-Sekundärknoten ist:

```
Node:                                   dr-bob
Queue manager status:                   Ended immediately
```

```
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    dr-alice
HA preferred location:                  dr-alice
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Secondary
DR status:                              See dr-alice
DR port:                                3000
DR local IP address:                    192.168.2.2
DR remote IP address list:              192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address:           See dr-alice

Node:                                   dr-alice
HA status:                              Normal

Node:                                   dr-charlie
HA status:                              Normal
```

Beispiel für laufende DR-Synchronisation auf einem Knoten, der ein DR-Primärknoten und HA-Primärknoten ist:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1
DR synchronization progress:            11.0%
DR estimated time to completion:        2018-09-06 14:55:05

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

Beispiel für DR-Status 'Partitioned' auf einem Knoten, der ein DR-Primärknoten und HA-Primärknoten ist:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Partitioned
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1
DR out of sync data:                    372KB

Node:                                   main-bob
HA status:                              Normal
```

```
Node:                                   main-charlie
HA status:                              Normal
```

V 9.3.0 Beispiel für nicht synchrone DR auf einem Knoten, der ein DR-Primärknoten und HA-Primärknoten ist:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Remote unavailable
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           Unknown
DR out of sync data:                    372KB
DR last in sync:                        2020-02-02 20:22:02

Node:                                   main-bob
HA status:                              Normal

Node:                                   main-charlie
HA status:                              Normal
```

V 9.3.0 Beispiel für nicht synchrone HA auf einem Knoten, der ein DR-Primärknoten und HA-Primärknoten ist:

```
Node:                                   main-alice
Queue manager status:                   Running
CPU:                                    0.00%
Memory:                                 123MB
Queue manager file system:              51MB used, 1.0GB allocated [5%]
HA role:                                Primary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
HA floating IP interface:               None
HA floating IP address:                 None
DR role:                                Primary
DR status:                              Normal
DR port:                                3000
DR local IP address:                    192.168.1.1
DR remote IP address list:              192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address:           192.168.2.1

Node:                                   main-bob
HA status:                              Inconsistent
HA out of sync data:                    15932KB
HA last in sync:                        2020-02-02 20:22:02

Node:                                   main-charlie
HA status:                              Normal
```

V 9.3.0 Beispiel für einen Zusammenfassungsstatus, der eine Diskrepanz zwischen der Betriebssystem-Kernelversion (RHEL 7.9) und dem DRBD-Kernelmodul (für RHEL 7.8) zeigt. Obwohl der Status meldet, dass DRBD-Kernelmodul geladen ist und der Warteschlangenmanager aktiv ist, sollten Sie das DRBD-Kernelmodul mit der Version aktualisieren, die für den aktiven Betriebssystemkern in dieser Situation vorgesehen ist.

```
Node:                                   main-alice
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-1127
```

```
DRBD version:                           9.1.1
DRBD kernel module status:              Loaded

Queue manager name:                     QM1
Queue manager status:                   Running
HA current location:                    This node
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
```

V 9.3.0 Beispiel für einen Zusammenfassungsstatus, der eine Diskrepanz zwischen der Betriebssystem-Kernelversion (RHEL 7.9) und dem DRBD-Kernelmodul (für RHEL 7.6) zeigt. In diesem Beispiel ist die Versionsabweichung schwerwiegender, und das DRBD-Kernelmodul kann nicht erfolgreich geladen werden. QM1 ist ein HA/DR-Warteschlangenmanager und wird auf einen anderen Knoten verschoben, dessen HA-Status unbekannt ist und dessen DR-Status unbekannt ist. Um diesen Fehler zu beheben, muss das DRBD-Kernelmodul mit dem Versionsziel für den aktiven Betriebssystemkern aktualisiert werden.

```
Node:                                   main-alice
OS kernel version:                      3.10.0-1160.15.2
DRBD OS kernel version:                 3.10.0-957
DRBD version:                           9.1.2+ptf.3
DRBD kernel module status:              Partially loaded

Queue manager name:                     QM1
Queue manager status:                   Running elsewhere
HA status:                              Unknown
HA current location:                    main-bob
HA preferred location:                  This node
HA blocked location:                    None
DR role:                                Primary
DR status:                              Unknown
```

**Zugehörige Verweise**

Linux rdqmstatus

## Linux *Betrieb in einer DR/HA-Umgebung*

Beim Betrieb in einer DR/HA-Umgebung sind hinsichtlich Hochverfügbarkeit und Disaster-Recovery unterschiedliche Überlegungen anzustellen.

Wenn der Knoten, auf dem ein DR/HA-RDQM aktiv ist, ausfällt, wird der RDQM automatisch von einem anderen Knoten in derselben HA-Gruppe übernommen. Wenn der gesamte Standort ausfällt, müssen Sie den RDQM manuell auf dem bevorzugten Knoten in der HA-Gruppe am Wiederherstellungsstandort starten. Hier sind dieselben Überlegungen anzustellen wie für einen gewöhnlichen DR-RDQM (siehe „Betrieb in einer Disaster-Recovery-Umgebung” auf Seite 661).

Wenn einer der Knoten vollständig ausfällt und ersetzt werden muss, beachten Sie die Anweisungen in den Abschnitten „Fehlgeschlagenen Knoten in einer Wiederherstellung nach einem Katastrophenfall ersetzen” auf Seite 663 und „Fehlgeschlagenen Knoten in einer Hochverfügbarkeitskonfiguration ersetzen” auf Seite 639.

## Linux *Ausgefallenen Knoten in einer DR/HA-Konfiguration ersetzen*

Wenn einer der Knoten in einer Ihrer HA-Gruppen ausfällt, können Sie ihn ersetzen.

### Informationen zu diesem Vorgang

Die Vorgehensweise ist davon abhängig, ob der Knoten, den Sie ersetzen, in der DR-Konfiguration ein Primär- oder Sekundärknoten ist. In beiden Fällen muss die Konfiguration des neuen Knotens mit der des zu ersetzenden Knotens identisch sein, d. h., er muss denselben Hostnamen, dieselben IP-Adressen usw. haben.

Es kann auch die Situation eintreten, dass die HA-Gruppe am Haupt- oder Wiederherstellungsstandort vollständig verloren gegangen ist und die gesamte HA-Gruppe ersetzt werden muss.

## Prozedur

- Führen Sie auf dem neuen Knoten folgende Schritte aus, um einen Ersatzknoten als Primärknoten in der DR-Konfiguration zu erstellen:
  a) Erstellen Sie eine `rdqm.ini`-Datei, die mit den Dateien auf den anderen Knoten übereinstimmt, und führen Sie dann den Befehl `rdqmadm -c` aus (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)" auf Seite 615).
  b) Führen Sie den Befehl `crtmqm -sxs -rr p` *`qmanager`* aus, um jeden DR/HA-RDQM erneut zu erstellen (siehe „DR/HA-RDQMs erstellen" auf Seite 669).
- Führen Sie auf dem neuen Knoten folgende Schritte aus, um einen Ersatzknoten als Sekundärknoten in der DR-Konfiguration zu erstellen:
  a) Erstellen Sie eine `rdqm.ini`-Datei, die mit den Dateien auf den anderen Knoten übereinstimmt, und führen Sie dann den Befehl `rdqmadm -c` aus (siehe „Definieren des Pacemaker-Clusters (HA-Gruppe)" auf Seite 615).
  b) Führen Sie den `crtmqm -sx -rr s` -Befehl *qmanager* aus, um jeden DR/HA-RDQM erneut zu erstellen (siehe „DR/HA-RDQMs erstellen" auf Seite 669).
- Gehen Sie wie folgt vor, um eine gesamte HA-Gruppe zu ersetzen:
  a) Wenn die gesamte HA-Gruppe am DR-Primärstandort (d. h. am Hauptstandort) verloren geht, müssen Sie die Schritte zur Durchführung einer verwalteten Übernahme an den DR-Sekundärstandort ausführen, um den Betrieb der DR/HA-RDQMs aufrechtzuerhalten (siehe „Betrieb in einer Disaster-Recovery-Umgebung" auf Seite 661). (Wenn eine gesamte HA-Gruppe am Wiederherstellungsstandort verloren geht, bleiben die DR/HA-RDQMs auf dem Hauptstandort aktiv, ohne dass Sie eingreifen müssen.)
  b) Erstellen Sie die HA-Gruppe auf den drei Ersatzknoten erneut, wie im Abschnitt „HA-Gruppen für DR/HA-RDQMs konfigurieren" auf Seite 668 beschrieben.
  c) Erstellen Sie die DR/HA-RDQMs in der neuen HA-Gruppe erneut, wie im Abschnitt „DR/HA-RDQMs erstellen" auf Seite 669 beschrieben.
  d) Führen Sie, falls erforderlich, eine verwaltete Übernahme vom Wiederherstellungsstandort zurück an den Hauptstandort durch.

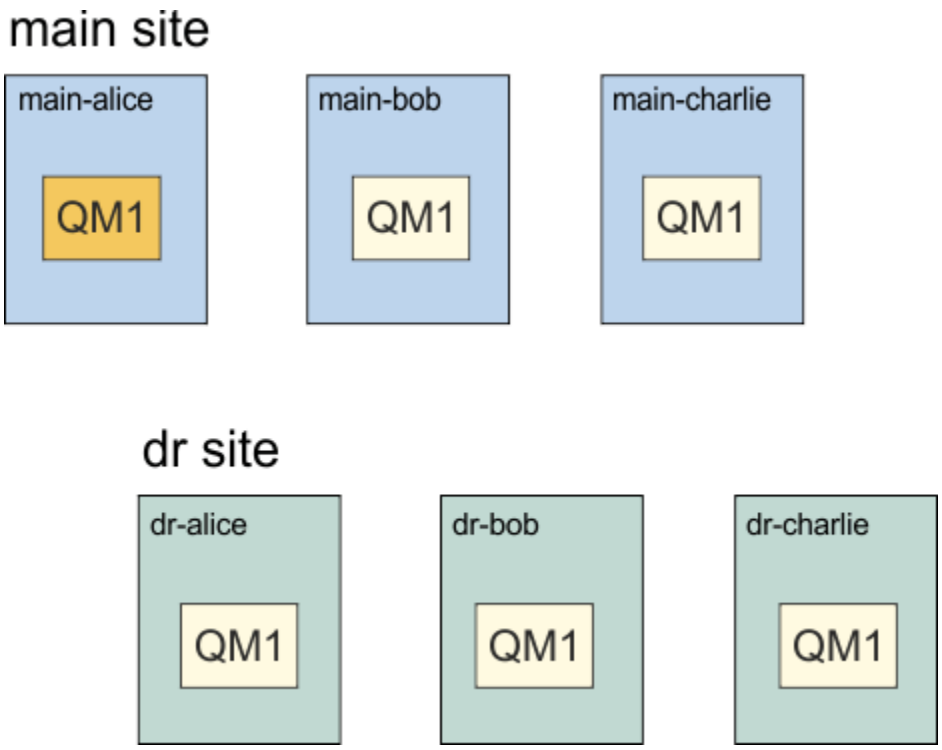
### Linux *Lösungsbeispiel für DR/HA-RDQM*

Dieses Beispiel zeigt, wie ein DR/HA-RDQM (Replicated Data Queue Manager) erstellt und gelöscht wird.

## DR/HA-RDQM erstellen

Die Beispielkonfiguration besteht aus zwei Standorten mit den Namen 'main' und 'dr'. An jedem Standort gibt es drei Knoten namens 'alice', 'bob' und 'charlie'. Die vollständigen Namen der Knoten setzen sich aus dem Standortnamen und dem jeweiligen Knotennamen zusammen, also 'main-alice', 'dr-alice' usw.

Mit den folgenden Schritten wird ein DR/HA-RDQM mit dem Namen QM1 erstellt, der auf dem Knoten main-alice ausgeführt wird. Der Haupt-Alice-Knoten ist die HA- und die DR-Primärknoten.




Wenn die lokalen und remote DR-IP-Adressen in der `rdqm.ini`-Datei angegeben sind, müssen keine IP-Adressen in der Befehlszeile angegeben werden, und ein DR/HA RDQM mit dem Namen QM1 kann erstellt werden, indem der folgende Befehl auf dem Haupt-Alice ausgeführt wird:

```
crtmqm -sx -rr p -rn DR1 -rp 7001 QM1
```

Wenn die lokalen DR-IP-Adressen in der `rdqm.ini`-Datei angegeben werden, können die remote DR-IP-Adressen in der Befehlszeile angegeben werden:

```
crtmqm -sx -rr p -ri 192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

Wenn in der `rdqm.ini`-Datei keine DR-IP-Adressen angegeben sind, können sowohl remote als auch lokale DR-IP-Adressen in der Befehlszeile angegeben werden:

```
crtmqm -sx -rr p -rl 192.168.1.1,192.168.1.2,192.168.1.3 -ri
192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

Die Ausgabe als Antwort auf die Erstellung von QM1 wird im folgenden Beispiel gezeigt:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'main-bob'.
Secondary queue manager created on 'main-charlie'.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qm1/qmgr/qm1' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QM1'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA secondary queue manager:
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

Kopieren Sie den Befehl aus der Nachricht, um die DR-Sekundärinstanz von QM1 auf dr-alice zu erstellen:

```
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

Auf dr-alice wird folgende Nachricht ausgegeben:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr-bob'.
Secondary queue manager created on 'dr-charlie'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
```

## Test der DR-Sekundärinstanz

Führen Sie zum Testen der Disaster-Recovery-Funktionen von QM1 folgenden Befehl auf main-alice aus, um QM1 zur DR-Sekundärinstanz zu machen:

```
rdqmdr -m QM1 -s
Queue manager 'QM1' has been made the DR secondary on this node.
```

Führen Sie folgenden Befehl auf dr-alice aus, um QM1 auf diesem Knoten zur DR-Primärinstanz zu machen:

```
rdqmdr -m QM1 -p
Queue manager 'QM1' has been made the DR primary on this node.
```

## Löschen eines DR/HA-RDQM

Um den DR/HA-RDQM mit dem Namen QM1 zu löschen, müssen Sie zunächst den Warteschlangenmanager auf main-alice beenden:

```
endmqm -w QM1
Replicated data queue manager disabled.
Waiting for queue manager 'QM1' to end.
IBM MQ queue manager 'QM1' ended.
```

Führen Sie dann folgenden Befehl auf main-alice aus, um QM1 zu löschen:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'main-bob'.
Secondary queue manager deleted on 'main-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

Schließlich müssen Sie QM1 auf dr-alice löschen:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'dr-bob'.
Secondary queue manager deleted on 'dr-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

**Zugehörige Konzepte**

„Betrieb in einer DR/HA-Umgebung” auf Seite 685
Beim Betrieb in einer DR/HA-Umgebung sind hinsichtlich Hochverfügbarkeit und Disaster-Recovery unterschiedliche Überlegungen anzustellen.

**Zugehörige Tasks**

„DR/HA-RDQMs erstellen” auf Seite 669
Verwenden Sie den Befehl **crtmqm**, um einen RDQM (Replicated Data Queue Manager) in einer DR/HA-Konfiguration zu erstellen.

„Löschen eines DR/HA-RDQM” auf Seite 674
Verwenden Sie zum Löschen eines DR/HA-RDQM (Replicated Data Queue Manager) den Befehl **dltmqm**.

## CP4I MQ Adv. Native HA

Die native Hochverfügbarkeit ist eine Hochverfügbarkeitslösung, die in Containerbereitstellungen von IBM MQverfügbar ist.

Eine native HA-Konfiguration besteht aus drei Knoten (z. B. drei Kubernetes -Pods) mit jeweils einer Instanz des Warteschlangenmanagers. Eine Instanz ist der aktive Warteschlangenmanager, der Nachrichten verarbeitet und in sein Protokoll schreibt. Immer wenn das Protokoll geschrieben wird, sendet der aktive Warteschlangenmanager die Daten an die beiden anderen Instanzen, die als 'Replikate' bezeichnet werden. Jedes Replikat schreibt in sein eigenes Protokoll, bestätigt die Daten und aktualisiert dann seine eigenen Warteschlangendaten aus dem replizierten Protokoll. Wenn der Knoten fehlschlägt, auf dem der aktive Warteschlangenmanager ausgeführt wird, übernimmt eine der Replikatinstanzen des Warteschlangenmanagers die aktive Rolle und verfügt über aktuelle Daten, mit denen gearbeitet werden kann.

Eine detaillierte Übersicht finden Sie unter Native HA im Abschnitt "Container" dieser Dokumentation.

Die folgende Abbildung zeigt eine typische Implementierung mit drei Instanzen eines Warteschlangenmanagers in drei Containern.

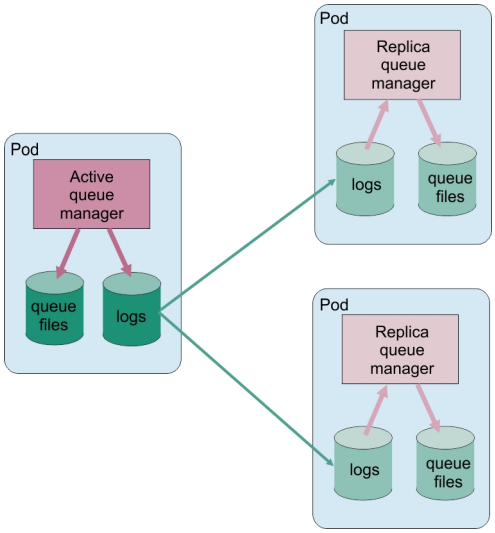



*Abbildung 83. Beispiel einer Native HA-Konfiguration*

## CP4I MQ Adv. *Native HA-Lösung erstellen*

Die empfohlene Methode zum Erstellen einer nativen HA-Lösung ist die Verwendung von IBM MQ Operator. Alternativ können Sie eigene Container erstellen und die native Hochverfügbarkeit manuell konfigurieren.

**Anmerkung:** Diese Informationen gelten nur für Containerumgebungen.

Wenn Sie eine native HA-Lösung mit IBM MQ Operatorerstellen möchten, finden Sie im Abschnitt Native HA eine Übersicht und ausführliche Anweisungen unter Beispiel: Native HA-Warteschlangenmanager konfigurieren .

Informationen zum Erstellen eigener Container und zum manuellen Konfigurieren der nativen Hochverfügbarkeit finden Sie unter Native HA-Gruppe erstellen, wenn eigene Container erstellt werden.

## CP4I MQ Adv. *Beenden von nativen HA-Warteschlangenmanagern*

Bei IBM MQ in Containern können Sie mit dem Befehl **endmqm** einen aktiven Warteschlangenmanager oder einen Replikatwarteschlangenmanager beenden, der Teil einer nativen HA-Gruppe ist.

### Informationen zu diesem Vorgang

**Anmerkung:** Diese Informationen gelten nur für Containerumgebungen.

Die Vorgehensweise zum Stoppen eines Warteschlangenmanagers, der zu einer nativen HA-Gruppe gehört, hängt davon ab, ob es sich um eine aktive oder eine Replikatinstanz handelt. Wenn Sie einen der Instanztypen beenden, wird eine Prüfung durchgeführt, um sicherzustellen, dass die Beendigung der Instanz das Quorum der nativen HA-Gruppe nicht unterbricht. Wenn das Quorum unterbrochen wird, schlägt der Befehl **endmqm** fehl.

Wenn Sie den Befehl **endmqm** absetzen, werden die anderen Instanzen in der Gruppe gewarnt, dass dies geschieht, damit sie keine Fehler melden, wenn die Verbindung unterbrochen wird.

Wenn eine aktive Instanz das Quorum verliert, weil zu viele Replikatinstanzen beendet oder getrennt werden, wartet die aktive Instanz einen konfigurierbaren Zeitraum, bevor sie vollständig beendet wird. Dies ermöglicht eine gewisse Zeit, um die Verarbeitung ordnungsgemäß zu beenden, anstatt dass die Verbindungen von Anwendungen unterbrochen werden. Dieser Zeitlimitwert kann mit dem Attribut `QuorumConnectivityTimeout` in der Zeilengruppe `NativeHALocalInstance` der Datei `qm.ini` angegeben werden. Der Standardwert ist 0 Sekunden.

## Prozedur

- Um die aktive Instanz eines Warteschlangenmanagers zu beenden, geben Sie den folgenden Befehl auf dem Knoten aus, auf dem die aktive Instanz ausgeführt wird:

  ```
  endmqm -s QMgrName
  ```

  – Geben Sie die Option `-r` an, damit Clientanwendungen eine Verbindung zu einer anderen Instanz wiederherstellen können.
  – Falls es sich bei dieser Instanz nicht um die aktive Instanz in der nativen HA-Gruppe handelt, schlägt der Befehl fehl.
  – Falls durch das Beenden dieser aktiven Instanz das Quorum der Gruppe fehlschlägt, schlägt der Befehl fehl. (Wenn andere Instanzen während der Ausführung dieses Befehls beendet werden oder nicht mehr verfügbar sind, wird dies bei der Überprüfung des Quorums möglicherweise nicht erkannt. Die native HA-Gruppe wird beendet und kann nur erneut gestartet werden, wenn genügend Instanzen verfügbar sind.)

  Wenn der aktive Warteschlangenmanager endet, übernimmt eine der Replikatinstanzen die aktive Rolle. Sie können nicht angeben, welches Replikat übernommen wird. Dies wird durch Vereinbarung innerhalb der Gruppe bestimmt und hängt davon ab, welches über die aktuellsten Transaktionsprotokolle verfügt.
- Geben Sie den folgenden Befehl aus, um eine Replikatinstanz eines Warteschlangenmanagers zu beenden:

  ```
  endmqm -x QMgrName
  ```

  – Wenn es sich bei dieser Instanz um die aktive Instanz handelt, schlägt der Befehl fehl.
  – Falls durch das Beenden dieser Replikatsinstanz das Quorum der Gruppe fehlschlägt, schlägt der Befehl fehl. (Wenn andere Instanzen während der Ausführung dieses Befehls beendet werden oder nicht mehr verfügbar sind, wird dies bei der Überprüfung des Quorums möglicherweise nicht erkannt. Die native HA-Gruppe wird beendet und kann nur erneut gestartet werden, wenn genügend Instanzen verfügbar sind.)

  **Anmerkung:** Sie können auch die Switches -c, -i, -p oder -w mit dem Befehl **endmqm** für native HA-Instanzen verwenden, unabhängig davon, in welcher Rolle sie sich befinden. Die Warteschlangenmanagerinstanz wird beendet, wobei die Auswirkung auf das Gruppenquorum ignoriert wird. Informationen werden jedoch weiterhin mit den anderen Instanzen in der Gruppe gemeinsam genutzt. Sie können diese Switches zusammen mit dem Parameter -s für die aktive Instanz verwenden. Sie können diese Switches nicht zusammen mit dem Switch -x für Replikatinstanzen verwenden.

**Zugehörige Verweise**

endmqm (Warteschlangenmanager beenden)

# Protokollierung: Stellen Sie sicher, dass die Nachrichten nicht verloren gehen.

IBM MQ zeichnet in einem Wiederherstellungsprotokoll alle signifikanten Änderungen an den persistenten Daten auf, die vom WS-Manager gesteuert werden.

Dies umfasst das Erstellen und Löschen von Objekten, persistenten Nachrichtenaktualisierungen, Transaktionsstatus, Änderungen an Objektattributen und Kanalaktivitäten. Das Protokoll enthält die Informationen, die Sie benötigen, um alle Aktualisierungen für Nachrichtenwarteschlangen wiederherzustellen, indem Sie folgende Schritte ausführen:

- Datensätze von WS-Manager-Änderungen werden beibehalten
- Datensätze von Warteschlangenaktualisierungen für die Verwendung durch den Neustartprozess beibehalten
- Wiederherstellung von Daten nach einem Hardware-oder Softwarefehler

IBM MQ stützt sich jedoch auch auf das Plattensystem, auf dem sich seine Dateien befinden, einschließlich der Protokolldateien. Wenn das Plattensystem selbst unzuverlässig ist, können Informationen, einschließlich Protokollinformationen, immer noch verloren gehen.

**Vorsicht:** Sie können die Wiederherstellungsprotokolle nicht in ein anderes Betriebssystem versetzen.

## Wie Logs aussehen

Protokolle bestehen aus primären und sekundären Dateien und einer Steuerdatei. Sie definieren die Anzahl und Größe der Protokolldateien und werden dort gespeichert, wo sie im Dateisystem gespeichert werden.

Ein IBM MQ-Protokoll besteht aus zwei Komponenten:

1. Eine oder mehrere Dateien mit Protokolldaten.
2. Eine Protokollsteuerdatei

Eine Datei mit Protokolldaten wird auch als Protokollspeicherbereich bezeichnet.

Es gibt eine Reihe von Protokollerweiterungen, die die Daten enthalten, die aufgezeichnet werden. Sie können die Anzahl und die Größe (wie in „Zeilengruppe 'LogDefaults' in der Datei 'mqs.ini'" auf Seite 100 erläutert) definieren oder den Systemstandardwert von drei primären und zwei sekundären Speicherbereichen übernehmen.

Jeder der drei primären und zwei sekundären Speicherbereiche nimmt standardmäßig den Wert 16 MB an.

Wenn Sie einen Warteschlangenmanager erstellen, ist die Anzahl der vorab zugeordneten Protokollextents die Anzahl der zugeordneten *primären* Protokollbereiche. Wenn Sie keine Zahl angeben, wird der Standardwert verwendet.

IBM MQ verwendet zwei Arten der Protokollierung:

- Umlauf
- Linear

  Die Anzahl der Protokollspeicherbereiche, die mit der linearen Protokollierung verwendet werden, kann sehr groß sein, abhängig von der Häufigkeit der Aufzeichnung der Datenträgerimages.

Weitere Informationen finden Sie unter „Typen der Protokollierung" auf Seite 692.

ALW Wenn Sie in IBM MQ for AIX or Linux-Systemen den Protokollpfad nicht geändert haben, werden Protokollspeicherbereiche unter dem folgenden Verzeichnis erstellt:

```
/var/mqm/log/QMgrName
```

Windows Wenn Sie in IBM MQ for Windows den Protokollpfad nicht geändert haben, werden die Protokollerweiterungen unter dem folgenden Verzeichnis erstellt:

```
C:\ProgramData\IBM\MQ\log\QMgrName
```

IBM MQ beginnt mit diesen primären Protokollspeicherbereichen, aber wenn der primäre Protokollspeicherbereich nicht ausreichend ist, ordnet er *sekundäre* Protokollspeicherbereiche zu. Dies macht sie dynamisch und entfernt sie, wenn die Nachfrage nach Protokollspeicherbereich reduziert wird. Standardmäßig können bis zu zwei sekundäre Protokollextents zugeordnet werden. Sie können diese Standardzuordnung wie in „IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern” auf Seite 90 beschrieben ändern.

Protokollerweiterungen werden entweder mit dem Buchstaben S oder mit dem Buchstaben R als Präfix festgelegt. Aktive, inaktive und überflüssige Speicherbereiche werden mit S vorfixiert, während die Wiederverwendung von Speicherbereichen mit R vorfixiert ist.

Wenn Sie den Warteschlangenmanager sichern oder wiederherstellen, sichern Sie alle aktiven, inaktiven und überflüssigen Extents zusammen mit der Protokollsteuerdatei zurück und stellen Sie diese wieder her.

**Anmerkung:** Sie müssen keine Wiederverwendungsbereiche sichern und wiederherstellen.

### *Die Protokollsteuerdatei*

Die Protokollsteuerdatei enthält Informationen, die zum Beschreiben des Status von Protokollspeicherbereichen erforderlich sind, z. B. Größe und Position des Protokolls sowie der Name des nächsten verfügbaren Speicherbereichs.

**Wichtig:** Die Protokollsteuerdatei dient nur der internen Verwendung des Warteschlangenmanagers.

Der WS-Manager verwaltet die Steuerdaten, die dem Status des Wiederherstellungsprotokolls in der Protokollsteuerdatei zugeordnet sind, und Sie dürfen den Inhalt der Protokollsteuerdatei nicht ändern.

Die Protokollsteuerdatei befindet sich im Protokollpfad und wird als `amqhlctl.lfh` bezeichnet. Stellen Sie beim Sichern oder Wiederherstellen des Warteschlangenmanagers sicher, dass die Protokollsteuerdatei zusammen mit Ihren Protokollspeicherbereichen gesichert und zurückgeschrieben wird.

## Typen der Protokollierung

In IBM MQ gibt es zwei Möglichkeiten, Aufzeichnungen von WS-Manager-Aktivitäten zu verwalten: Umlaufprotokollierung und lineare Protokollierung. Ein dritter Protokollierungstyp (Replizierung) wird nur von nativen HA-Konfigurationen verwendet.

### Umlaufprotokollierung

Verwenden Sie die Umlaufprotokollierung, wenn Sie möchten, dass die Wiederherstellung erneut gestartet wird. Verwenden Sie das Protokoll, um Transaktionen rückgängig zu machen, die sich in Bearbeitung befanden, als das System gestoppt wurde.

Die Umlaufprotokollierung speichert alle Neustartdaten in einem Ring von Protokolldateien. Die Protokollierung füllt die erste Datei im Ring aus und wird dann in die nächste Datei verschoben, und so weiter, bis alle Dateien voll sind. Anschließend geht es zurück in die erste Datei im Ring und beginnt erneut. Dies wird solange fortgesetzt, wie das Produkt im Gebrauch ist, und hat den Vorteil, dass Sie keine Protokolldateien mehr ausführen.

IBM MQ speichert die Protokolleinträge, die erforderlich sind, um den WS-Manager ohne Datenverlust erneut zu starten, bis diese nicht mehr benötigt werden, um die Wiederherstellung des Warteschlangenmanagers sicherzustellen. Der Mechanismus zum Freigeben von Protokolldateien für die Wiederverwendung wird in „Verwenden des Prüfpunktprogramms zur Sicherstellung einer vollständigen Wiederherstellung” auf Seite 695 beschrieben.

## Lineare Protokollierung

Verwenden Sie die lineare Protokollierung, wenn Sie sowohl die Wiederherstellung als auch die Datenträgerwiederherstellung (Wiederherstellung verlorener oder beschädigter Daten durch Wiedergabe des Inhalts des Protokolls) verwenden möchten. Bei der linearen Protokollierung werden die Protokolldaten in einer fortlaufenden Folge von Protokolldateien aufbewahrt.

Die Protokolldateien können optional wie folgt sein:

- Wiederverwendet, aber nur dann, wenn sie nicht mehr benötigt werden, um eine Wiederherstellung nach einem Neustart oder eine Wiederherstellung des Datenträgers zu starten.
- Manuell archiviert für längerfristige Speicherung und Analyse.

Die Häufigkeit von Datenträgerimages bestimmt, wann lineare Protokolldateien wiederverwendet werden können, und ist ein wichtiger Faktor, wie viel Plattenspeicherplatz für die linearen Protokolldateien verfügbar sein muss.

Sie können den Warteschlangenmanager so konfigurieren, dass er automatisch regelmäßige Medienimages auf der Basis der Zeit-oder Protokollverwendung verwendet oder Sie können Medienimages manuell planen.

Ihr Administrator entscheidet, welche Richtlinie implementiert werden soll, und die Auswirkungen auf die Plattenspeicherplatzbelegung. Protokolldateien, die für die Neustartwiederherstellung benötigt werden, müssen immer verfügbar sein, während Protokolldateien, die nur für die Datenträgerwiederherstellung benötigt werden, in einem längerfristig erstellbaren Speicher archiviert werden können, z. B. Band.

Wenn Ihr Administrator automatische Protokollverwaltung und automatische Medienimages aktiviert, verhält sich die lineare Protokollierung ähnlich wie ein sehr großes Umlaufprotokoll, aber mit der verbesserten Redundanz gegen Datenträgerfehler, die durch die Datenträgerwiederherstellung aktiviert sind.

Ab IBM MQ 9.1.0 können Sie einen vorhandenen Protokolltyp für einen Warteschlangenmanager mit dem Befehl migmqlog von linear zu umlaufend bzw. von umlaufend zu linear ändern.

## Replizierte Protokollierung

CP4I

Verwenden Sie die replizierte Protokollierung zum Erstellen einer nativen HA-Konfiguration. Beim Erstellen einer nativen HA-Gruppe erstellen Sie drei Warteschlangenmanager auf verschiedenen Knoten. Sie geben den Typ der replizierten Protokollierung zusammen mit einem eindeutigen Instanznamen für jeden der Warteschlangenmanager an. Die native HA-Konfiguration stellt eine Hochverfügbarkeitslösung bereit, bei der eine aktive Instanz Protokolldaten auf zwei Replikatinstanzen repliziert. Wenn die aktive Instanz ausfällt, übernimmt eine der Replikatsinstanzen die aktive Rolle. Die Protokollreplikation stellt sicher, dass, wenn überhaupt, nur wenige Daten verloren gehen. Weitere Informationen finden Sie unter „Native HA” auf Seite 689. Ein repliziertes Protokoll entspricht einem linearen Protokoll mit aktivierter automatischer Protokollverwaltung und automatischen Medienimages.

## Nicht aktive lineare Protokollextents

Multi

Wenn Sie in IBM MQ 9.1.0 oder höher die automatische Protokollverwaltung, einschließlich der Archivierung, verwenden, protokolliert die Protokollfunktion die linearen Protokolloberbereiche, die nicht aktiv sind.

**Achtung:** Wenn Sie die automatische Protokollverwaltung verwenden, ohne die Archivierung zu archivieren, wird die Verwendung eines Sicherungswarteschlangenmanagers für diesen Prozess nicht unterstützt.

ALW Wenn ein Protokollspeicherbereich für die Wiederherstellung nicht mehr benötigt wird und, falls erforderlich, archiviert wird, wird die Protokollfunktion an einem geeigneten Punkt entweder die Protokollspeicherausdehnung löschen oder sie wiederverwenden.

Ein wiederverwenter Protokollspeicherbereich wird umbenannt, damit er der nächste in der Protokollfolge ist. Die Nachricht AMQ7490 wird in regelmäßigen Abständen geschrieben, die angibt, wie viele Speicherbereiche erstellt, gelöscht oder wiederverwendet wurden.

Die Protokollfunktion wählt die Anzahl der Speicherbereiche aus, die für die Wiederverwendung bereitgehalten werden sollen, und wann diese Extents gelöscht werden sollen.

## Aktives Protokoll

Es gibt eine Reihe von Dateien, die sowohl in der linearen als auch in der Umlaufprotokollierung als *aktiv* angegeben sind. Das aktive Protokoll ist die maximale Größe des Protokollspeicherbereichs, unabhängig davon, ob Sie die Umlaufprotokollierung oder die lineare Protokollierung verwenden, die durch die Neustartwiederherstellung referenziert werden kann.

Die Anzahl der aktiven Protokolldateien ist in der Regel kleiner als die Anzahl primärer Protokolldateien, die in den Konfigurationsdateien definiert sind. (Informationen zum Definieren der Anzahl finden Sie im Abschnitt „Berechnen der Größe des Protokolls” auf Seite 698.)

Beachten Sie, dass der Speicherbereich für aktive Protokolldateien nicht den für die Datenträgerwiederherstellung erforderlichen Speicherbereich enthält und dass die Anzahl der Protokolldateien, die mit der linearen Protokollierung verwendet werden, sehr groß sein kann, abhängig von Ihrem Nachrichtenfluss und der Häufigkeit von Medienimages.

## Inaktives Protokoll

Wenn eine Protokolldatei für die Wiederanlaufwiederherstellung nicht mehr benötigt wird, wird sie *inaktiv* . Protokolldateien, die weder für die Wiederherstellung nach einem Neustart noch für die Datenträgerwiederherstellung erforderlich sind, können als überflüssige Protokolldateien betrachtet werden.

Wenn Sie die automatische Protokollverwaltung verwenden, steuert der Warteschlangenmanager die Verarbeitung dieser überflüssigen Protokolldateien. Wenn Sie das manuelle Protokollmanagement ausgewählt haben, wird es in die Zuständigkeit Ihres Administrators, überflüssige Protokolldateien zu verwalten (z. B. löschen und archivieren), wenn sie für Ihre Operation nicht mehr von Interesse sind.

Weitere Informationen zur Aussonderung von Protokolldateien finden Sie im Abschnitt „Protokolle verwalten” auf Seite 705.

## Sekundäre Protokolldateien

Obwohl sekundäre Protokolldateien für die lineare Protokollierung definiert sind, werden sie im normalen Betrieb nicht verwendet. Wenn eine Situation eintritt, die wahrscheinlich auf langlebige Transaktionen zurückzuführen ist, ist es nicht möglich, eine Datei aus dem aktiven Pool zu befreien, da sie möglicherweise noch für einen Neustart erforderlich ist. Sekundärdateien werden formatiert und dem Pool für aktive Protokolldateien hinzugefügt.

Wenn die Anzahl der verfügbaren Sekundärdateien verwendet wird, werden Anforderungen für die meisten weiteren Operationen, die eine Protokollaktivität erfordern, zurückgewiesen, wenn ein MQRC_RESOURCE_PROBLEM-Rückkehrcode an die Anwendung zurückgegeben wird, und alle Transaktionen mit langer Laufzeit werden für asynchrone ROLLBACK-Operationen in Betracht gezogen.

**Achtung:** Alle Protokollierungstypen können mit einem unerwarteten Stromausfall umgehen, vorausgesetzt, es liegt kein Hardwarefehler vor.

## Verwenden des Prüfpunktprogramms zur Sicherstellung einer vollständigen Wiederherstellung

Sowohl die Umlaufprotokollierung als auch die Warteschlangenmanager der linearen Protokollierung unterstützen die Neustartwiederherstellung. Unabhängig davon, wie abrupt die vorherige Instanz des Warteschlangenmanagers (z. B. ein Stromausfall) nach einem Neustart beendet wird, stellt der Warteschlangenmanager seinen persistenten Status am Ende der Beendigung wieder in den richtigen Transaktionsstatus zurück.

Die Wiederanlaufwiederherstellung hängt von der Plattenintegrität ab, Ebenso sollte das Betriebssystem die Plattenintegrität sicherstellen, unabhängig davon, wie abrupt eine Beendigung des Betriebssystems auftreten könnte.

Im höchst ungewöhnlichen Fall, dass die Plattenintegrität nicht beibehalten wird, bietet die lineare Protokollierung (und Datenträgerwiederherstellung) einige weitere Redundanz-und Wiederherstellbarkeitsoptionen. Mit immer häufiger verbreiteten Technologien wie RAID ist es immer seltener, Probleme mit der Plattenintegrität zu erleiden, und viele Unternehmen konfigurieren die Umlaufprotokollierung und verwenden nur die Wiederherstellung nach einem Neustart.

IBM MQ ist als Ressourcenmanager nach dem WAL-Prinzip (Write Ahead Logging) konzipiert. Persistente Aktualisierungen von Nachrichtenwarteschlangen werden in zwei Schritten vorgenommen:

1. Protokollsätze, die die Aktualisierung darstellen, werden zuverlässig in das Wiederherstellungsprotokoll geschrieben.
2. Die Warteschlangendatei oder die Puffer werden in einer Art und Weise aktualisiert, die für Ihr System am effizientesten ist, aber nicht unbedingt konsistent.

Die Protokolldateien können somit mehr auf dem neuesten Stand sein als der zugrunde liegende Warteschlangenpuffer und Dateistatus.

Wenn diese Situation unvermindert fortgesetzt werden konnte, ist ein sehr umfangreicher Protokollwiedergaberückstand erforderlich, um den Warteschlangenstatus nach einer Recovery nach einem Systemabsturz konsistent zu machen.

IBM MQ verwendet `checkpoints` , um den Umfang der Protokollwiedergabe zu begrenzen, die nach einer Recovery nach einem Systemabsturz erforderlich ist. Das Schlüsselereignis, das steuert, ob eine Protokolldatei als aktiv bezeichnet wird oder nicht als `checkpoint` bezeichnet wird.

Ein IBM MQ-Prüfpunkt ist ein Punkt:

- Der Konsistenz zwischen den Wiederherstellungsprotokolldateien und Objektdateien.
- Der einen Bereich im Protokoll identifiziert, ab dem durch Wiedergabe der nachfolgenden Protokollsätze garantiert wird, dass die Warteschlange in den richtigen logischen Zustand zurückgespeichert wird, der beim eventuellen Beenden des WS-Managers vorlag.

Im Rahmen eines Prüfpunkts schreibt IBM MQ nach Bedarf ältere Aktualisierungen in die Warteschlangen-Dateien, um den Umfang der Protokollsätze zu begrenzen, die wiedergegeben werden müssen, um die Warteschlangen bei der Wiederherstellung nach einem Systemabsturz in einen konsistenten Status zu bringen.

Der letzte vollständige Prüfpunkt markiert einen Punkt im Protokoll, von dem aus die Wiedergabe während der Recovery nach einem Systemabsturz ausgeführt werden muss. Die Häufigkeit des Prüfpunkts ist somit ein Kompromiss zwischen dem Systemaufwand für die Aufzeichnung von Prüfpunkten und der Verbesserung der potenziellen Wiederherstellungszeit, die von diesen Prüfpunkten impliziert wird.

Ab IBM MQ 9.1.0 plant die Protokollfunktion Prüfpunkte häufiger ein (der nächste Prüfpunkt ist bereits vor Abschluss des vorherigen Prüfpunkts geplant), da sie versucht, das aktive Protokoll in den primären Protokollspeicherbereichen zu halten. Wenn dies nicht möglich ist, wird der Fehler AMQ7466 protokolliert.

Die Position im Protokoll des Starts des letzten vollständigen Prüfpunkts ist einer der Schlüsselfaktoren für die Bestimmung, ob eine Protokolldatei aktiv oder inaktiv ist. Der andere Schlüsselfaktor ist die Position im Protokoll des ersten Protokollsatzes, die sich auf die erste persistente Aktualisierung bezieht, die durch eine aktuelle aktive Transaktion erstellt wurde.

Wenn ein neuer Prüfpunkt in der zweiten oder einer späteren Protokolldatei aufgezeichnet wird und sich keine aktuelle Transaktion auf einen Protokollsatz in der ersten Protokolldatei bezieht, wird die erste Protokolldatei inaktiv. Bei einer Umlaufprotokollierung ist die erste Protokolldatei jetzt bereit, wiederverwendet zu werden. Bei linearer Protokollierung wird die erste Protokolldatei in der Regel weiterhin für die Datenträgerwiederherstellung benötigt.

Wenn Sie die Umlaufprotokollierung oder die automatische Protokollverwaltung konfigurieren, verwaltet der WS-Manager die inaktiven Protokolldateien. Wenn Sie die lineare Protokollierung mit manuellem Protokollmanagement konfigurieren, wird es zu einer Verwaltungstask zum Verwalten der inaktiven Dateien entsprechend den Anforderungen Ihrer Operation.

IBM MQ generiert Prüfpunkte automatisch. Sie werden zu den folgenden Zeitpunkten ausgeführt:

- Beim Start des Warteschlangenmanagers
- Beim Herunterfahren
- Wenn der Protokollspeicherbereich niedrig ist
- Multi Nachdem 50.000 Operationen protokolliert wurden, seit der vorherige Prüfpunkt
- z/OS Nachdem *number_of_operations* seit dem vorherigen Prüfpunkt protokolliert worden ist, wobei *number_of_operations* die Anzahl der in der Eigenschaft **LOGLOAD** festgelegten Operationen ist.

Wenn IBM MQ erneut gestartet wird, findet es den letzten Prüfpunktsatz im Protokoll. Diese Informationen werden in der Prüfpunktdatei gehalten, die am Ende jedes Prüfpunkts aktualisiert wird. Alle Operationen, die seit dem Prüfpunkt ausgeführt wurden, werden erneut wiedergegeben. Dies wird als "Wiedergabephase" bezeichnet.

Die Wiedergabephase bewirkt, dass die Warteschlangen wieder in den logischen Zustand zurückgebracht werden, in dem sie sich vor dem Systemausfall oder dem Systemabschluss befanden. Während der Wiedergabephase wird eine Liste der Transaktionen erstellt, die bei Auftreten des Systemausfalls oder des Systemabschlusses in den Fluten ausgeführt wurden.

Multi Die Nachrichten AMQ7229 und AMQ7230 werden ausgegeben, um den Fortschritt der Wiederholungsphase anzugeben.

Um zu wissen, welche Operationen zurückgesetzt oder festgeschrieben werden sollen, greift IBM MQ auf jeden aktiven Protokollsatz zu, der einer unvollständigen Transaktion zugeordnet ist. Dies wird als Recovery-Phase bezeichnet.

Multi Die Nachrichten AMQ7231, AMQ7232 und AMQ7234 werden ausgegeben, um den Fortschritt der Wiederherstellungsphase anzugeben.

Sobald während der Wiederherstellungsphase auf alle erforderlichen Protokollsätze zugegriffen wurde, wird jede aktive Transaktion wiederum aufgelöst, und jede der Transaktion zugeordnete Operation wird entweder zurückgesetzt oder festgeschrieben. Dies wird als Auflösungsphase bezeichnet.

Multi Die Nachricht AMQ7233 wird ausgegeben, um den Fortschritt der Auflösungsphase anzugeben.

z/OS Unter z/OS besteht der Neustartprozess aus verschiedenen Phasen.

1. Der Wiederherstellungsprotokollbereich wird basierend auf der Datenträgerwiederherstellung erstellt, die für die Seitengruppen und den ältesten Protokollsatz erforderlich ist, der für die Sicherung von Arbeitseinheiten und für unbestätigte Arbeitseinheiten erforderlich ist.
2. Sobald der Protokollbereich ermittelt wurde, wird die Vorwärtsprotokollablesung ausgeführt, um die Seite auf den neuesten Stand zu bringen und alle Nachrichten zu sperren, die sich auf unbestätigte oder in unbestätigte Arbeitseinheiten bezogene Arbeitseinheiten beziehen.
3. Wenn die Vorwärtsprotokollablesung abgeschlossen ist, werden die Protokolle rückwärts gelesen, um alle Arbeitseinheiten auszustellen, die zum Zeitpunkt des Ausfalls in der Vergangenheit oder in einem Backout ausgeführt wurden.

z/OS Ein Beispiel für die Nachrichten, die angezeigt werden können:

```
CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS -  806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range  815
from RBA=00000001E45FF7AD to RBA=00000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE -  816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range  817
from RBA=00000001E48C1882 to RBA=00000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE -  818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED
```

**Anmerkung:** Wenn eine große Menge an Protokoll gelesen werden soll, werden die Nachrichten CSQR031I (Vorwärtswiederherstellung) und CSQR033I (Rückwärtswiederherstellung) in regelmäßigen Abständen ausgegeben, um den Fortschritt zu zeigen.

In Abbildung 84 auf Seite 697 werden alle Datensätze vor dem letzten Prüfpunkt (Prüfpunkt 2) von IBM MQ nicht mehr benötigt. Die Warteschlangen können aus den Prüfpunktinformationen und allen späteren Protokolleinträgen wiederhergestellt werden. Für die Umlaufprotokollierung können alle freigegebenen Dateien erneut verwendet werden, bevor der Prüfpunkt wiederverwendet werden kann. Für ein lineares Protokoll müssen die freigegebenen Protokolldateien nicht mehr für den normalen Betrieb zugänglich gemacht werden und werden inaktiv. Im Beispiel wird der Zeigerkopfzeiger so bewegt, dass er auf den letzten Prüfpunkt (Prüfpunkt 2) zeigt, der dann zum neuen Warteschlangenkopf, Head 2, wird. Protokolldatei 1 kann jetzt wiederverwendet werden.

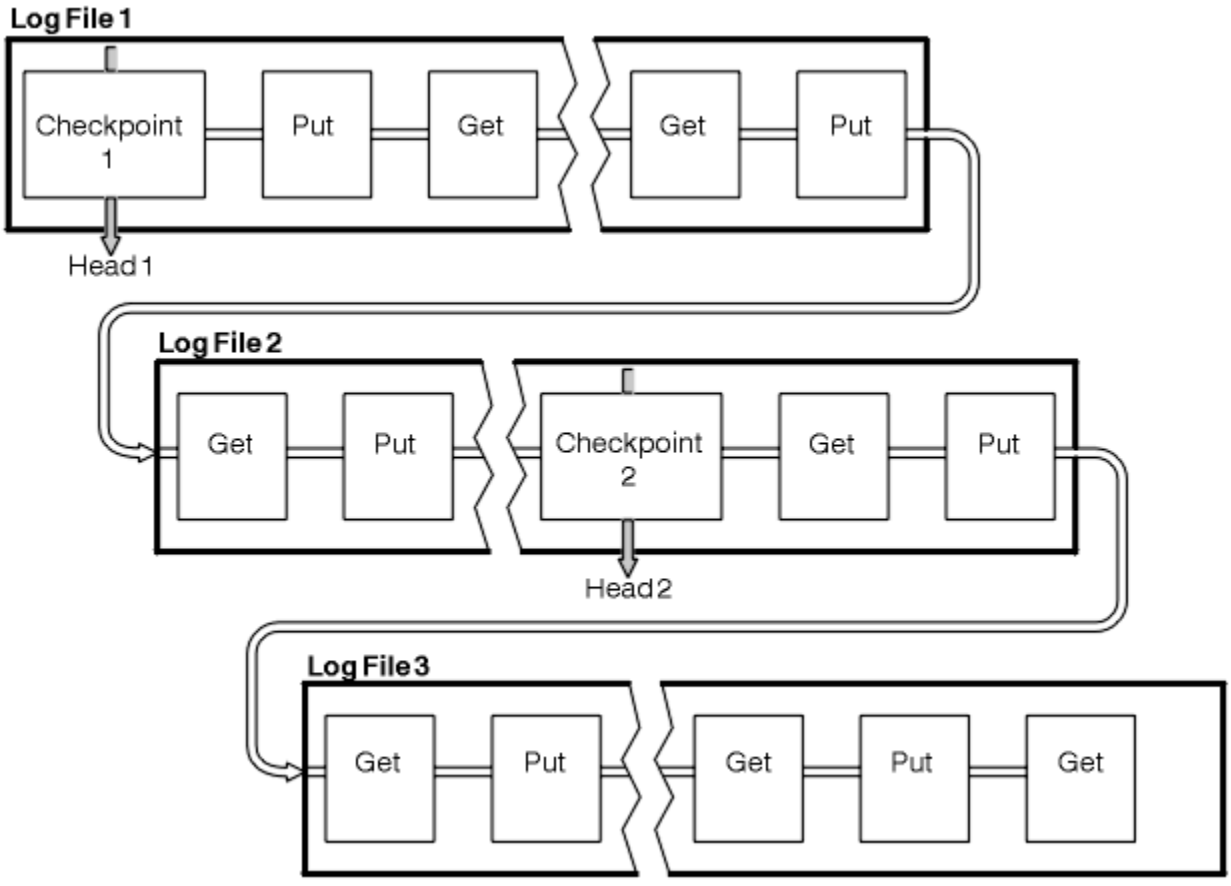



*Abbildung 84. Prüfpunktprogramm*

### *Prüfpunktverlaufes mit Transaktionen mit langer Laufzeit*

Wie eine Transaktion mit langer Laufzeit die Wiederverwendung von Protokolldateien beeinflusst.

Abbildung 85 auf Seite 698 zeigt, wie sich eine Transaktion mit langer Laufzeit auf die Wiederverwendung von Protokolldateien auswirkt. In dem Beispiel hat eine Transaktion mit langer Laufzeit einen Eintrag in das Protokoll gestellt, das als LR 1 dargestellt wird, nachdem der erste Prüfpunkt angezeigt wurde. Die Transaktion wird (bei Punkt LR 2) bis nach dem dritten Prüfpunkt nicht vollständig ausgeführt. Alle Protokollinformationen von LR 1 werden beibehalten, um die Wiederherstellung dieser Transaktion zu ermöglichen, sofern dies erforderlich ist, bis sie abgeschlossen ist.

Nach Abschluss der Transaktion mit langer Laufzeit wird bei LR 2 der Protokollkopf logisch zu Prüfpunkt 3, dem zuletzt protokollierten Prüfpunkt, verschoben. Die Dateien, die Protokollsätze vor Prüfpunkt 3, Head 2, enthalten, werden nicht mehr benötigt. Wenn Sie die Umlaufprotokollierung verwenden, kann der Speicherbereich wiederverwendet werden.

Wenn die primären Protokolldateien vollständig voll sind, bevor die Transaktion mit langer Laufzeit abgeschlossen ist, werden möglicherweise sekundäre Protokolldateien verwendet, um zu vermeiden, dass die Protokolle voll sind.

Aktivitäten, die vollständig unter der Steuerung des Warteschlangenmanagers stehen, z. B. das Prüfpunktprogramm, sind geplant, um die Aktivität im primären Protokoll zu halten.

Wenn jedoch ein sekundärer Protokollspeicherbereich erforderlich ist, um das Verhalten außerhalb der Steuerung des Warteschlangenmanagers (z. B. die Dauer einer Ihrer Transaktionen) zu unterstützen, versucht der Warteschlangenmanager, jeden definierten sekundären Protokollspeicherbereich zu verwenden, damit die Aktivität abgeschlossen werden kann.

Wenn diese Aktivität nicht bis zu 80% des gesamten Protokollspeicherbereichs abgeschlossen ist, leitet der Warteschlangenmanager die Aktion zum Zurückfordern von Protokollspeicherbereich ein, unabhängig von der Tatsache, dass dies Auswirkungen auf die Anwendung hat.

Wenn der Protokollkopf verschoben wird und Sie die Umlaufprotokollierung verwenden, werden die primären Protokolldateien möglicherweise für die Wiederverwendung in Frage gestellt, und die Protokollfunktion, nachdem die aktuelle Datei gefüllt wurde, verwendet die erste verfügbare Primärdatei erneut. Wenn Sie die lineare Protokollierung verwenden, wird der Protokollkopf immer noch in den aktiven Pool verschoben, und die erste Datei wird inaktiv. Eine neue Primärdatei wird formatiert und der Basis des Pools hinzugefügt, wenn sie für zukünftige Protokollierungsaktivitäten bereit ist.

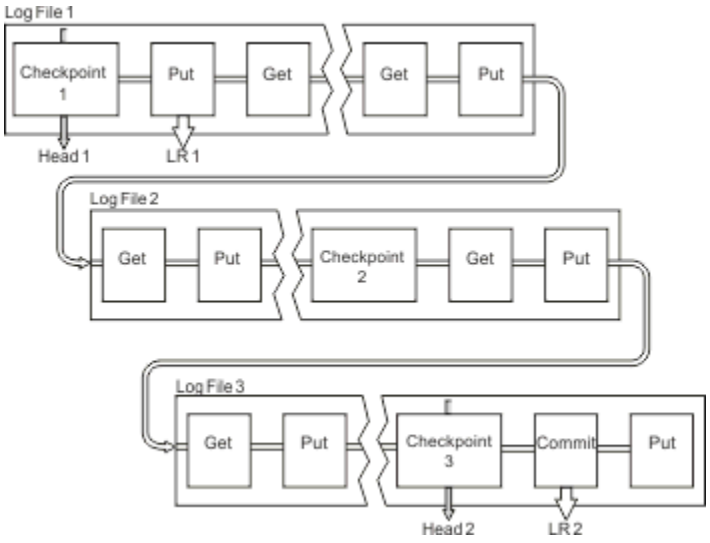



*Abbildung 85. Prüfpunktüberprüfung mit einer Transaktion mit langer Laufzeit*

## Berechnen der Größe des Protokolls

Schätzen der Größe des Protokollwarteschlangenmanagers.

Nach der Entscheidung, ob der WS-Manager eine kreisförmige oder lineare Protokollierung verwendet, müssen Sie die Größe des aktiven Protokolls schätzen, den der Warteschlangenmanager benötigt. Die Größe des aktiven Protokolls wird durch die folgenden Protokollkonfigurationsparameter bestimmt:

**LogFilePages**
Die Größe der einzelnen primären und sekundären Protokolldateien in Einheiten von 4-KB-Seiten.

**LogPrimaryFiles**
Die Anzahl der vorab zugeordneten primären Protokolldateien.

**LogSecondaryFiles**
Die Anzahl der sekundären Protokolldateien, die für die Verwendung erstellt werden können, wenn die primären Protokolldateien voll werden.

**Anmerkungen:**

1. Sie können die Anzahl der primären und sekundären Protokolldateien bei jedem Start des Warteschlangenmanagers ändern, auch wenn Sie die Auswirkungen der Änderungen, die Sie an den Sekundärprotokollen vornehmen, nicht sofort bemerken.
2. Sie können die Größe der Protokolldatei nicht ändern. Sie müssen es **vor** der Erstellung des Warteschlangenmanagers festlegen.
3. Die Anzahl primärer Protokolldateien und die Größe der Protokolldatei bestimmen die Größe des Protokollspeicherbereichs, der beim Erstellen des Warteschlangenmanagers vorab zugeordnet wird.
4. Die Gesamtzahl der primären und sekundären Protokolldateien darf auf AIX and Linux-Systemen nicht größer als 511 sein (bzw. 255 unter Windows), wodurch bei Vorliegen von Transaktionen mit langer Laufzeit die maximale Größe des Protokollspeicherbereichs, der dem WS-Manager für die Wiederherstellung nach einem Neustart zur Verfügung steht, begrenzt wird. Die Größe des Protokollspeicherbereichs, den der WS-Manager möglicherweise für die Datenträgerwiederherstellung benötigt, teilt diese Begrenzung nicht mit.
5. Wenn die *Umlaufprotokollierung* verwendet wird, verwendet der WS-Manager den primären und den sekundären Protokollspeicherbereich. Der WS-Manager ordnet bis zu einem Grenzwert eine sekundäre Protokolldatei zu, wenn eine Protokolldatei voll wird und die nächste primäre Protokolldatei in der Sequenz nicht verfügbar ist.

   Informationen zur Anzahl der Protokolle, die Sie zuordnen müssen, finden Sie unter „Wie groß sollte ich mein aktives Protokoll machen?" auf Seite 699. Die primären Protokollspeicherbereiche werden in der Reihenfolge verwendet, und diese Reihenfolge ändert sich nicht.

   Wenn Sie beispielsweise drei primäre Protokolle `0`, `1` und `2` haben, ist die Reihenfolge der Verwendung `0`, `1`, `2` gefolgt von `1`, `2`, `0`, `2`, `0`, `1`, zurück zu `0`, `1`, `2` usw. Alle sekundären Protokolle, die Sie zugeordnet haben, werden nach Bedarf interspergt.
6. Primäre Protokolldateien werden für die Wiederverwendung während eines Prüfpunkts verfügbar gemacht. Der Warteschlangenmanager berücksichtigt vor dem Ausführen eines Prüfpunkts sowohl den primären als auch den sekundären Protokollspeicherbereich, da die Menge des Protokollspeicherbereichs niedrig ist.

   Der Warteschlangenmanager versucht, Prüfpunkte in einer Weise zu terminieren, die die Protokollbelegung in den primären Speicherbereichen hält.

Weitere Informationen finden Sie unter „Zeilengruppe 'LogDefaults' in der Datei 'mqs.ini'" auf Seite 100.

## *Wie groß sollte ich mein aktives Protokoll machen?*

Die Größe des aktiven Protokollwarteschlangenmanagers wird geschätzt.

Die Größe des aktiven Protokolls ist begrenzt durch:

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

Das Protokoll sollte so groß sein, dass es mit der längsten aktiven Transaktion fertig ist, die ausgeführt wird, wenn der Warteschlangenmanager die maximale Datenmenge pro Sekunde auf Platte schreibt.

Wenn Ihre am längsten laufende Transaktion für N Sekunden ausgeführt wird und die maximale Menge an Daten pro Sekunde, die vom Warteschlangenmanager auf Platte geschrieben werden, B Byte pro Sekunde im Protokoll beträgt, sollte Ihr Protokoll mindestens die folgenden Schritte ausführen:

```
logsize >= 2 * (N+1) * B
```

Der WS-Manager schreibt wahrscheinlich die maximale Datenmenge pro Sekunde auf die Platte, wenn Sie eine Spitzenauslastung ausführen, oder es kann sein, wenn Sie Medienimages aufzeichnen.

Wenn eine Transaktion so lange ausgeführt wird, dass der Protokollspeicherbereich, der seinen ersten Protokollsatz enthält, nicht im aktiven Protokoll enthalten ist, macht der Warteschlangenmanager eine aktive Transaktion zu einem Zeitpunkt rückgängig, beginnend mit der Transaktion mit dem ältesten Protokollsatz.

Der Warteschlangenmanager muss alte Protokollerweiterungen inaktiv machen, bevor die maximale Anzahl primärer und sekundärer Dateien verwendet wird, und der Warteschlangenmanager muss einen anderen Protokollspeicherbereich zuordnen.

Legen Sie fest, wie lange Ihre längste aktive Transaktion ausgeführt werden soll, bevor der WS-Manager die Transaktion rückgängig machen kann. Ihre längste aktive Transaktion wartet möglicherweise auf langsamen Datenaustausch im Netz oder, im Falle einer schlecht konzipierten Transaktion, auf die Benutzereingabe.

Sie können untersuchen, wie lange Ihre Transaktion mit der längsten Laufzeit dauert, indem Sie den folgenden **runmqsc** -Befehl absetzen:

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

Wenn Sie den Befehl `dspmqtrn -a` absetzen, werden alle XA-und Nicht-XA-Befehle in allen Status angezeigt.

Wenn Sie diesen Befehl absetzen, werden Datum und Uhrzeit aufgelistet, an denen der erste Protokollsatz für alle aktuellen Transaktionen geschrieben wurde.

**Achtung:** Für die Berechnung der Protokollgröße ist es die Zeit, seit der erste Protokollsatz geschrieben wurde, und nicht die Zeit seit dem Start der Anwendung oder Transaktion. Runden Sie die Länge der längsten aktiven Transaktion auf die nächste Sekunde auf. Dies liegt an den Optimierungen im Warteschlangenmanager.

Der erste Protokollsatz kann lange nach dem Start der Anwendung geschrieben werden, wenn die Anwendung z. B. mit einem MQGET-Aufruf beginnt, der auf eine Zeitdauer wartet, bevor er tatsächlich eine Nachricht erhält.

Durch Überprüfung der maximal beobachteten Datums-und Uhrzeitausgabe von der

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

Befehl, den Sie ursprünglich eingegeben haben, ab dem aktuellen Datum und der aktuellen Uhrzeit können Sie schätzen, wie lange die längste laufende Transaktion ausgeführt wird.

Stellen Sie sicher, dass Sie diesen **runmqsc** -Befehl wiederholt ausführen, während Ihre Transaktionen mit der längsten Ausführungszeit mit Spitzenauslastung ausgeführt werden, damit Sie die Länge Ihrer Transaktion mit der längsten Laufzeit nicht unterschätzen.

Verwenden Sie in IBM MQ 8.0 die Betriebssystemtools, z. B. **iostat** auf UNIX-Plattformen.

Ab IBM MQ 9.0 können Sie ermitteln, wie viele Bytes der Warteschlangenmanager pro Sekunde in das Protokoll schreibt, indem Sie folgenden Befehl ausgeben:

```
amqsrua -m qmgr -c DISK -t Log
```

Die geschriebenen logischen Byte zeigen die Byte pro Sekunde an, die der Warteschlangenmanager in das Protokoll schreibt. For example:

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

In diesem Beispiel sind die logischen Byte pro Sekunde, die in das Protokoll geschrieben werden, 12760669/sec oder ca. 12 MiB pro Sekunde.

verwenden

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

zeigt, dass die längste aktive Transaktion:

```
CONN(57E14F6820700069)
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r)                     UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```

Da das aktuelle Datum und die aktuelle Uhrzeit 2016-09-20 16.44.19 war, war diese Transaktion für 5 Sekunden ausgeführt worden. Sie müssen jedoch Transaktionen mit 10 Sekunden tolerieren, bevor der WS-Manager sie zurücksetzt. Daher sollte die Protokollgröße wie folgt sein:

```
2 * (10 + 1) * 12 = 264 MiB
```

.

Die Anzahl der Protokolldateien muss die größte erwartete Protokollgröße (im vorherigen Text berechnet) enthalten können. Dies wird wie folgt sein:

Mindestanzahl der Protokolldateien = (Erforderliche Protokollgröße)/(**LogFilePages** * Protokolldateiseitengröße (4096))

Bei Verwendung des Standardwertes **LogFilePages**, der 4096 ist, und der Protokollgrößenschätzung von 264MiB, die im vorhergehenden Text berechnet wird, sollte die Mindestanzahl an Protokolldateien sein:

```
264MiB / (4096 x 4096) = 16.5
```

Das heißt, 17 Protokolldateien.

Wenn Sie Ihr Protokoll so groß sind, dass die erwartete Workload in den Primärdateien ausgeführt wird, gilt Folgendes:

- Die Sekundärdateien stellen einen Teil der Kontingenz bereit, falls zusätzlicher Protokollspeicherbereich benötigt wird.
- Umlaufprotokollierung immer mit vorab zugeordneten Primärdateien, die geringfügig schneller ist als die Zuordnung und das Freigeben von Sekundärdateien.
- Der Warteschlangenmanager verwendet nur den Speicherbereich, der in den Primärdateien verbleibt, um zu berechnen, wann der nächste Prüfpunkt ausgeführt werden soll.

Geben Sie daher im obigen Beispiel die folgenden Werte ein, damit die Workload in den primären Protokolldateien ausgeführt wird:

- **LogFilePages** = 4096
- **LogPrimaryFiles** = 17
- **LogSecondaryFiles** = 5

Beachten Sie Folgendes:

- In diesem Beispiel beträgt die Anzahl der 5 sekundärer Server mehr als 20 Prozent des aktiven Protokollspeicherbereichs.

  Ab IBM MQ 9.1.0 versucht die Protokollfunktion, die Arbeitslast ausschließlich in den Primärdateien zu halten. Daher terminiert die Protokollfunktion Prüfpunkte, wenn ein Bruchteil der Primärdateien allein voll ist.

  Die sekundären Dateien sind unvorhersehbare Ereignisse, wenn es zu unerwarteten Transaktionen mit langer Laufzeit kommt.

  Sie sollten sich bewusst sein, dass der Warteschlangenmanager Maßnahmen ergreift, um die Protokollspeicherbelegung zu reduzieren, wenn mehr als 80 Prozent des gesamten Protokollspeicherbereichs verwendet werden.
- Führen Sie die gleiche Berechnung aus, unabhängig davon, ob Sie die lineare oder die Umlaufprotokollierung verwenden

  Es macht keinen Unterschied, ob Sie die Größe eines linearen oder kreisförmigen aktiven Protokolls berechnen, da das Konzept der aktiven Protokolldatei sowohl in der linearen Protokollierung als auch in der Umlaufprotokollierung dieselbe ist.
- Die Protokollextents, die für die Datenträgerwiederherstellung benötigt werden, befinden sich nicht im aktiven Protokoll und werden daher nicht in der Anzahl der primären und sekundären Dateien gezählt.
- Ab IBM MQ 9.1.0 steht das Feld *LOGUTIL* in DISPLAY QMSTATUS LOG zur Verfügung, mit dessen Hilfe Sie die ungefähr erforderliche Größe des aktiven Protokolls berechnen können.

  Dieses Feld ist so konzipiert, dass Sie eine angemessene Schätzung der erforderlichen Protokollgröße ohne ständige Stichprobenentnahme vornehmen können, um die Dauer der längsten aktiven Transaktionen oder den Spitzendurchsatz des Warteschlangenmanagers zu ermitteln.

## Wie groß sollte ich meine LogFilePages machen?

Stellen Sie sicher, dass Ihre LogFilePages so groß sind, dass Sie die Größe Ihres aktiven Protokolls ohne großen Aufwand erhöhen können, ohne die maximale Anzahl an Primärdateien zu erreichen. Einige wenige große Protokolldateien sind vielen kleinen Protokolldateien vorzuziehen, da einige große Protokolldateien Ihnen mehr Flexibilität bieten, um die Größe Ihres Protokolls zu erhöhen, falls Sie dies benötigen.

Für die lineare Protokollierung können sehr große Protokolldateien die Leistungsvariable erstellen. Bei sehr großen Protokolldateien gibt es einen größeren Schritt, um eine neue Protokolldatei zu erstellen und zu formatieren oder um eine alte Datei zu archivieren. Dies ist eher ein Problem mit der Verwaltung von manuellen und Archivierungsprotokolldateien, da neue Protokolldateien mit automatischer Protokollverwaltung nur selten erstellt werden.

### *Was passiert, wenn ich mein Protokoll zu klein mache?*

Punkte, die Sie berücksichtigen müssen, wenn Sie die Mindestgröße des Protokolls schätzen.

Wenn Sie Ihr Protokoll zu klein machen:

- Transaktionen mit langer Laufzeit werden zurückgesetzt.
- Der nächste Prüfpunkt soll vor dem Ende des vorherigen Prüfpunkts gestartet werden.

**Wichtig:** Egal, wie ungenau Sie die Größe des Protokolls schätzen, die Datenintegrität wird beibehalten.

Im Abschnitt „Verwenden des Prüfpunktprogramms zur Sicherstellung einer vollständigen Wiederherstellung” auf Seite 695 finden Sie eine Erläuterung zu Prüfpunkten. Wenn die Größe des Protokollspeicherbereichs in den aktiven Protokollbereichen knapp wird, terminiert der Warteschlangenmanager die Prüfpunkte häufiger.

Ein Prüfpunkt nimmt eine gewisse Zeit in Anspruch; es wird nicht sofort angezeigt. Je mehr Daten in den Prüfpunkt aufgenommen werden müssen, um so länger dauert der Prüfpunkt. Wenn es sich bei dem Protokoll um kleine Prüfpunkte handelt, können sich die Prüfpunkte überschneiden. Dies bedeutet,

dass der nächste Prüfpunkt angefordert wird, bevor der vorherige Prüfpunkt beendet wurde. Wenn dies geschieht, werden Fehlernachrichten geschrieben.

Wenn Transaktionen mit langer Laufzeit zurückgesetzt werden, oder sich die Prüfpunkte überschneiden, setzt der Warteschlangenmanager die Verarbeitung der Workload fort. Kürzere aktive Transaktionen werden weiterhin normal ausgeführt.

Der Warteschlangenmanager wird jedoch nicht optimal ausgeführt, und die Leistung kann beeinträchtigt werden. Sie sollten den Warteschlangenmanager mit ausreichend Protokollspeicherbereich erneut starten.

### *Was passiert, wenn ich mein Protokoll zu groß mache?*

Punkte, die Sie berücksichtigen müssen, wenn Sie die maximale Größe des Protokolls schätzen.

Wenn Sie Ihr Protokoll zu groß machen:

- Sie können die Zeit für einen Wiederanlauf nach Systemabsturz erhöhen, obwohl dies unwahrscheinlich ist.
- Sie verwenden unnötigen Plattenspeicherplatz.
- Sehr lange laufende Transaktionen werden toleriert.

**Wichtig:** Egal, wie ungenau Sie die Größe des Protokolls schätzen, die Datenintegrität wird beibehalten.

Wenn Sie die maximale Größe des Protokolls schätzen wollen, können Sie die Statistik zur Protokollauslastung verwenden. Weitere Informationen finden Sie unter „Legen Sie fest, wie IMGLOGLN und IMGINTVL festgelegt werden sollen.” auf Seite 709 und ALTER QMGR.

Eine Beschreibung der Art und Weise, wie der Warteschlangenmanager das Protokoll beim Neustart liest, finden Sie in „Verwenden des Prüfpunktprogramms zur Sicherstellung einer vollständigen Wiederherstellung” auf Seite 695. Der Warteschlangenmanager gibt das Protokoll vom letzten Prüfpunkt aus und löst dann alle Transaktionen auf, die aktiv waren, als der WS-Manager beendet wurde.

Um eine Transaktion aufzulösen, liest der Warteschlangenmanager alle Protokollsätze zurück, die dieser Transaktion zugeordnet sind. Diese Protokollsätze können den letzten Prüfpunkt vorgeben.

Wenn Sie dem Warteschlangenmanager ein sehr großes Protokoll zuordnen, erteilen Sie dem Warteschlangenmanager die Berechtigung, jeden Protokollsatz im Protokoll beim Neustart zu lesen, obwohl dies in der Regel nicht der WS-Manager ist. Potenziell kann dieser Prozess im unwahrscheinlichen Fall, dass dies geschieht, längere Zeit in Anspruch nehmen.

Wenn das Prüfpunktprogramm unerwarteterweise gestoppt wurde, bevor der Warteschlangenmanager beendet wurde, erhöht dies die Wiederanlaufzeit für einen Warteschlangenmanager mit einem großen Protokoll drastisch. Wenn Sie die Größe des Protokolls begrenzen, wird die Wiederanlaufzeit für den Notfall begrenzt.

Um diese Probleme zu vermeiden, sollten Sie Folgendes sicherstellen:

- Ihre Workload kann problemlos in ein Protokoll passen, das nicht übermäßig groß ist.
- Sie vermeiden Transaktionen mit langer Laufzeit.

### *Wie groß sollte ich mein Protokolldateisystem machen?*

Schätzen der Größe des Protokolldateisystems, das ein Warteschlangenmanager benötigt.

Es ist wichtig, dass Sie Ihr Protokolldateisystem groß genug machen, damit Ihr Warteschlangenmanager genügend Speicherplatz für das Schreiben des Protokolls hat. Wenn der Warteschlangenmanager das Protokolldateisystem vollständig ausfüllt, schreibt er FFDCs, ROLLBACK-Transaktionen und kann den Warteschlangenmanager abrupt beenden.

Die Größe des Plattenspeicherplatzes, den Sie für Ihr Protokoll reservieren, muss mindestens so groß wie das aktive Protokoll sein. Wie viel größer ist, hängt ab von:

- Ihre Auswahl des Protokolltyps (linear oder kreisförmig)
- Die Größe des aktiven Protokolls (Primärdateien, Sekundärdateien, Protokolldateiseiten)

- Ihre Auswahl der Protokollverwaltung (manuell, automatisch oder archiviert)
- Ihre Notfallpläne im Fall eines beschädigten Objekts.

Wenn Sie ein Umlaufprotokoll auswählen, sollte Ihr Protokolldateisystem

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

Dadurch kann der WS-Manager in alle primären und sekundären Dateien schreiben. Unter außergewöhnlichen Umständen kann der WS-Manager einen zusätzlichen Speicherbereich außerhalb der Anzahl der Absender schreiben. Der vorhergehende Algorithmus berücksichtigt dies.

Wenn Sie ein lineares Protokoll auswählen, sollte das Protokolldateisystem erheblich größer sein als das aktive Protokoll.

Wenn Sie die manuelle Protokollverwaltung auswählen, schreibt der Warteschlangenmanager weiterhin in neue Protokollspeicherbereiche, da er sie benötigt, und es liegt in Ihrer Verantwortung, sie zu löschen (und archivieren), wenn sie nicht mehr benötigt werden.

Wie viel größer das Protokolldateisystem sein muss, hängt im Wesentlichen von Ihrer Strategie ab, um überflüssige oder inaktive Speicherbereiche zu löschen.

Sie können sich entscheiden, Extents zu archivieren und zu löschen, sobald sie inaktiv sind (nicht für die Neustartwiederherstellung erforderlich), oder Sie können sich entscheiden, nur überflüssige Speicherbereiche zu archivieren und zu löschen (die nicht für Datenträger benötigt werden, oder die Wiederherstellung erneut starten).

Wenn Sie nur überflüssige Speicherbereiche archivieren und löschen, und wenn Sie ein beschädigtes Objekt haben, wird **MEDIALOG** nicht vorwärts verschoben, sodass keine weiteren Speicherbereiche überflüssig werden. Sie werden die Archivierung stoppen und Extents löschen, bis Sie das Problem lösen, vielleicht durch die Wiederherstellung des Objekts.

Wenn Sie die Workload nicht stoppen, hängt es von der Größe des Protokolldateisystems ab, wie viel Zeit Sie zur Lösung des Problems haben. Daher ist es am besten, wenn Sie ein großzügiges Protokolldateisystem verwenden, wenn Sie die lineare Protokollierung verwenden.

Wenn Sie ein lineares Protokoll und die automatische Verwaltung oder die Verwaltung von Archivprotokolldateien auswählen, verwendet der Warteschlangenmanager Protokollextents erneut.

Protokollspeicherbereiche, die für die Wiederverwendung verfügbar sind, werden mit dem Buchstaben R vorfixiert. Wenn ein Medienimage aufgezeichnet wird, da überflüssige Speicherbereiche archiviert werden, kann der Warteschlangenmanager diese Speicherbereiche erneut verwenden.

Daher sind die Wiederverwendungsbereiche kleiner als die Datenlänge, die in das Protokoll zwischen Medienimages geschrieben wird:

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

Beim automatischen Aufzeichnen von Medienimages und beim Festlegen von **IMGLOGLN** kann `LogDataLengthBetweenMediaImages` doppelt so groß wie **IMGLOGLN** sein, da **IMGLOGLN** ein Ziel ohne festgelegte maximale Anzahl ist.

Wenn Sie Medienimages manuell aufzeichnen oder automatisch nach Intervall aufzeichnen, hängt `LogDataLengthBetweenMediaImages` von Ihrer Auslastung und dem Intervall zwischen den Images ab.

Neben aktiven Extents und Wiederverwendungsbereichen gibt es inaktive Speicherbereiche (nur für die Datenträgerwiederherstellung erforderlich) und überflüssige Speicherbereiche (die nicht für den Neustart oder die Datenträgerwiederherstellung benötigt werden).

Bei Verwendung der automatischen oder Archivierungsprotokollverwaltung verwendet der Warteschlangenmanager keine Speicherbereiche, die für die Datenträgerwiederherstellung benötigt werden. Die Anzahl inaktiver Speicherbereiche hängt also davon ab, wie oft Sie Medienimages verwenden und ob Sie sie manuell oder automatisch übernehmen.

**IMGINTVL** und **IMGLOGLN** sind Ziele, kein fester Mindestwert oder kein Maximum zwischen Medienimages. Wenn Sie jedoch die maximale Größe des Protokolldateisystems schätzen, die Sie möglicherweise benötigen, ist es unwahrscheinlich, dass automatische Medienimages mehr als zweimal **IMGINTVL** oder **IMGLOGLN** getrennt aufgezeichnet werden.

Wenn Sie Ihr Protokolldateisystem mit Hilfe der automatischen oder der Archivprotokollverwaltung dimensionieren, sollten Sie auch überlegen, was passieren kann, wenn eine Warteschlange oder ein anderes Objekt beschädigt ist. In diesem Fall ist der Warteschlangenmanager nicht in der Lage, ein Datenträgerimage des beschädigten Objekts zu nehmen, und **MEDIALOG** wird nicht vorwärts bewegt.

Wenn Ihre Auslastung fortgesetzt wird, wird Ihr inaktives Protokoll nicht zurückgehalten, da der älteste Speicherbereich, der für die Datenträgerwiederherstellung benötigt wird, noch benötigt wird und nicht wiederverwendet werden kann. Wenn Ihre Workload fortgesetzt wird, müssen Sie bis zum vollständigen Problem des Protokolldateisystems das Problem beheben, bevor der WS-Manager mit dem Rollback von Transaktionen beginnt und möglicherweise sogar abrupt beendet wird.

Daher für die automatische und Archivprotokollverwaltung:

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +
(((TimeBetweenMediaImages *2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))
* LogFilePages
```

**Anmerkung:** Der vorangehende Algorithmus geht davon aus, dass **SET LOG ARCHIVED** für jeden Speicherbereich aufgerufen wird, sobald er nicht mehr für die Datenträgerwiederherstellung benötigt wird, für die Archivprotokollverwaltung.

## Protokolle verwalten

Ab IBM MQ 9.1.0 unterstützt das Produkt die automatische Protokollverwaltung und die automatische Datenträgerwiederherstellung linearer Protokolle. Umlaufprotokolle sind fast selbstverwaltet, erfordern aber manchmal einen Eingriff zum Beheben von Speicherplatzproblemen.

**Anmerkung:** IBM i Die automatische Protokollverwaltung und das Archivprotokollmanagement sind unter IBM i nicht zulässig.

Bei der Umlaufprotokollierung hat der WS-Manager den freigegebenen Speicherbereich in den Protokolldateien freigegeben. Diese Aktivität wird für den Benutzer nicht angezeigt, und Sie sehen in der Regel nicht die Menge des verwendeten Plattenspeicherplatzes, da der zugeordnete Speicherbereich schnell wiederverwendet wird.

Ab IBM MQ 9.1.0 können Sie bei Verwendung der Umlaufprotokollierung sekundäre Dateien löschen. Weitere Informationen finden Sie unter RESET QMGR **TYPE(REDUCELOG)** .

Bei linearer Protokollierung kann sich das Protokoll füllen, wenn ein Prüfpunkt nicht für eine lange Zeit ausgeführt wurde oder wenn eine bereits lange laufende Transaktion einen Protokollsatz vor langer Zeit geschrieben hat. Der WS-Manager versucht oft genug Prüfpunkte zu nehmen, um das erste Problem zu vermeiden.

Multi Wenn das Protokoll voll ist, wird die Nachricht AMQ7463 ausgegeben. Wenn das Protokoll gefüllt wird, weil eine Transaktion mit langer Laufzeit verhindert hat, dass der Speicherbereich freigegeben wurde, wird die Nachricht AMQ7465 ausgegeben.

Von den Protokollsätzen werden nur die Datensätze benötigt, die seit dem Start des letzten vollständigen Prüfpunkts geschrieben wurden, und diejenigen, die von aktiven Transaktionen geschrieben wurden, zum erneuten Starten des Warteschlangenmanagers.

Mit der Zeit werden die ältesten Protokollsätze, die geschrieben wurden, für den Neustart des Warteschlangenmanagers nicht mehr erforderlich.

Wenn eine Transaktion mit langer Laufzeit erkannt wird, ist die asynchrone ROLLBACK-Operation dieser Transaktion geplant. Wenn die asynchrone ROLLBACK-Operation aus einem unerwarteten Grund fehlschlagen sollte, geben einige MQI-Aufrufe 'MQRC_RESOURCE_PROBLEM' in dieser Situation zurück.

Beachten Sie, dass Speicherplatz für das Festschreiben oder Zurücksetzen aller unvollständigen Transaktionen reserviert ist, sodass **MQCMIT** oder **MQBACK** nicht fehlschlagen sollte.

Eine Anwendung, bei der eine Transaktion auf diese Weise rückgängig gemacht wird, kann keine nachfolgenden **MQPUT**- oder **MQGET**-Operationen ausführen, die den Synchronisationspunkt unter derselben Transaktion angeben.

Der Versuch, eine Nachricht unter Synchronisationspunkt in diesem Status zu setzen oder abzurufen, gibt den Wert MQRC_BACKED_OUT zurück. Die Anwendung kann dann **MQCMIT**ausgeben, das MQRC_BACKED_OUT oder **MQBACK** zurückgibt und eine neue Transaktion starten. Wenn die Transaktion, die zu viel Protokollspeicherbereich belegt, rückgängig gemacht wurde, wird der Protokollspeicherbereich freigegeben, und der Warteschlangenmanager wird weiterhin normal ausgeführt.

### *Was passiert, wenn eine Platte voll ist*

Wenn ein Warteschlangenmanager für die Verwendung der linearen Protokollierung konfiguriert ist, reagiert die Protokollierungskomponente des Warteschlangenmanagers folgendermaßen auf eine volle Plattenbelegung.

Wenn die Platte mit den Protokolldateien vollständig belegt ist, gilt Folgendes:

- Der Warteschlangenmanager erkennt diese Bedingung nur beim Erstellen einer neuen Protokolldatei der erforderlichen Größe, was vor dem Zeitpunkt vorgenommen wird, zu der sie benötigt wird.
- Die vollständige Plattenbelegung wird erkannt, wenn das Betriebssystem einen Fehler von der Anforderung zurückgibt, die Datei auf die erforderliche Größe zu erweitern.
- Der Warteschlangenmanager gibt die Nachricht AMQ6708 an das Fehlerprotokoll des Warteschlangenmanagers aus.
- In das Verzeichnis mit den systemweiten Fehlern wird ein Datensatz zu First Failure Support Technology (FFST) geschrieben. Dieser Datensatz enthält Einzelheiten zu der Bedingung der vollständigen Plattenbelegung und sollte aufbewahrt werden, falls Sie sich an den IBM Support wenden müssen.

Die Protokolldateien werden in ihrer festen Größe erstellt und nicht als Protokollsätze, die in sie geschrieben werden, erweitert. Dies bedeutet, dass der verfügbare Plattenspeicherplatz für IBM MQ nur beim Erstellen einer neuen Datei knapp werden kann; beim Schreiben eines Datensatzes ins Protokoll kann dies hingegen nicht geschehen. IBM MQ weiß immer, wie viel Speicherplatz in den vorhandenen Protokolldateien verfügbar ist, und verwaltet den Speicherbereich in den Dateien entsprechend.

Ab IBM MQ 9.1.0 stehen Ihnen bei Verwendung der linearen Protokollierung folgende Optionen zur Verfügung:

- Automatische Verwaltung von Protokollerweiterungen.

  Weitere Informationen zu den neuen Protokollattributen finden Sie in DISPLAY QMSTATUS .

  Siehe auch die folgenden Befehle oder ihre entsprechenden PCF-Äquivalente:

  – RESET QMGR
  – SET LOG für verteilte Plattformen
- Die Optionen, die die Verwendung von Medienimages steuern.

  Weitere Informationen hierzu finden Sie im Befehl ALTER QMGR und ALTER QUEUES :

  – IMGINTVL
  – IMGLOGLN
  – IMGRCOVO
  – IMGRCOVQ
  – IMGSCHED

Die Umlaufprotokollierung gibt ein Ressourcenproblem zurück.

Wenn Sie noch nicht über Speicherplatz hinaus, überprüfen Sie, ob die Konfiguration des Protokolls in der Warteschlangenmanagerkonfigurationsdatei korrekt ist. Möglicherweise können Sie die Anzahl primärer

oder sekundärer Protokolldateien reduzieren, so dass das Protokoll den verfügbaren Speicherplatz nicht übersteigt.

Sie können die Größe der Protokolldateien für einen vorhandenen WS-Manager nicht ändern. Der Warteschlangenmanager erfordert, dass alle Protokollspeicherbereiche die gleiche Größe haben.

### *Protokolldateien verwalten*

Allokalisieren Sie genügend Speicherplatz für Ihre Protokolldateien. Bei linearer Protokollierung können Sie alte Protokolldateien löschen, wenn sie nicht mehr benötigt werden.

## Spezifische Informationen zur Umlaufprotokollierung

Wenn Sie die Umlaufprotokollierung verwenden, stellen Sie beim Konfigurieren Ihres Systems sicher, dass genügend Speicherplatz vorhanden ist, um die Protokolldateien zu speichern (siehe „Zeilengruppe 'LogDefaults' in der Datei 'mqs.ini'" auf Seite 100 und „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140). Die Größe des Plattenspeicherplatzes, der vom Protokoll verwendet wird, erhöht sich nicht über die konfigurierte Größe hinaus, einschließlich des Speicherbereichs für sekundäre Dateien, die bei Bedarf erstellt werden sollen.

## Spezifische Informationen zur linearen Protokollierung

Wenn Sie ein lineares Protokoll verwenden, werden die Protokolldateien kontinuierlich hinzugefügt, wenn Daten protokolliert werden, und die Menge des verwendeten Plattenspeicherplatzes wird mit der Zeit erhöht. Wenn die Rate der Daten, die protokolliert werden, hoch ist, wird der Plattenspeicherplatz schnell von neuen Protokolldateien verwendet.

Im Laufe der Zeit sind die älteren Protokolldateien für ein lineares Protokoll nicht mehr erforderlich, um den WS-Manager erneut zu starten oder die Datenträgerwiederherstellung beschädigter Objekte auszuführen. Mit den folgenden Methoden wird festgelegt, welche Protokolldateien noch benötigt werden:

**Protokollfunktionsereignisnachrichten**
Wenn ein signifikantes Ereignis auftritt, z. B. ein Plattendatenträgerimage, werden Ereignisnachrichten der Protokollfunktion generiert. Der Inhalt der Protokollfunktionsereignisnachrichten gibt die Protokolldateien an, die für den Neustart des Warteschlangenmanagers noch erforderlich sind, und die Datenträgerwiederherstellung. Weitere Informationen zu Ereignisnachrichten der Protokollfunktion finden Sie unter Protokollierungsereignisse.

**Status des Warteschlangenmanagers**
Durch Ausführen des MQSC-Befehls, DISPLAY QMSTATUS oder des PCF-Befehls "Inquire Queue Manager Status" werden die WS-Manager-Informationen, einschließlich Details der erforderlichen Protokolldateien, zurückgegeben. Weitere Informationen zu MQSC-Befehlen finden Sie unter IBM MQ mit MQSC-Befehlen verwalten. Informationen zu PCF-Befehlen finden Sie unter Verwaltungstasks automatisieren.

**WS-Manager-Nachrichten**
Der WS-Manager gibt in regelmäßigen Abständen ein Nachrichtenpaar aus, um anzugeben, welche Protokolldateien erforderlich sind:

- Mit der Nachricht AMQ7467I wird der Name der ältesten Protokolldatei angegeben, die für den Neustart des Warteschlangenmanagers erforderlich ist. Diese Protokolldatei und alle neueren Protokolldateien müssen beim Neustart des Warteschlangenmanagers verfügbar sein.
- Die Nachricht AMQ7468I gibt den Namen der ältesten Protokolldatei an, die für die Datenträgerwiederherstellung erforderlich ist.

Um "ältere" und "neuere" Protokolldateien zu ermitteln, verwenden Sie die Protokolldateinummer und nicht die Änderungszeiten, die vom Dateisystem angewendet werden.

## Informationen, die für beide Protokollierungstypen gelten

Nur Protokolldateien, die für den Neustart des Warteschlangenmanagers erforderlich sind, sind für aktive Protokolldateien erforderlich. Inaktive Protokolldateien können in ein Archivierungsmedium, wie z. B. ein

Band für die Wiederherstellung nach einem Katastrophenfall, kopiert und aus dem Protokollverzeichnis entfernt werden. Inaktive Protokolldateien, die für die Datenträgerwiederherstellung nicht erforderlich sind, können als überflüssige Protokolldateien betrachtet werden. Sie können überflüssige Protokolldateien löschen, wenn sie für Ihre Operation nicht mehr von Interesse sind.

Wenn eine erforderliche Protokolldatei nicht gefunden werden kann, wird die Bedienernachricht AMQ6767E ausgegeben. Stellen Sie die Protokolldatei und alle nachfolgenden Protokolldateien dem Warteschlangenmanager zur Verfügung und wiederholen Sie die Operation.

## Extents des Bereinigungsprotokolls werden automatisch-lineare Protokollierung

Multi

Ab IBM MQ 9.1.0 haben Sie die Möglichkeit, die automatische Verwaltung linearer Protokollspeicherbereiche zu verwenden, die für die Wiederherstellung nicht mehr benötigt werden.

Zum Einrichten der automatischen Verwaltung wird das Attribut **LogManagement** in der Zeilengruppe 'Log' der Datei 'qm.ini' oder der IBM MQ Explorer verwendet. Weitere Informationen finden Sie unter „Zeilengruppe 'Log' in der Datei 'qm.ini'" auf Seite 140.

Weitere Informationen zum Betrieb des Protokolls und zu den folgenden Befehlen für die Verwendung des Protokolls finden Sie unter dem Parameter LOG von **DISPLAY QMSTATUS** :

- RESET QMGR
- SET LOG

## Medienimages automatisch-nur lineare Protokollierung

Ab IBM MQ 9.1.0 gibt es einen allgemeinen Schalter, mit dem gesteuert wird, ob der Warteschlangenmanager automatisch Medienimages schreibt; in der Standardeinstellung ist dieser Schalter nicht eingeschaltet.

Sie können steuern, ob die automatische Datenträgerabbildung und die Häufigkeit des Prozesses durch die Verwendung der folgenden WS-Manager-Attribute gesteuert werden:

**IMGSCHED**
Gibt an, ob der Warteschlangenmanager Medienimages automatisch schreibt.

**IMGINTVL**
Frequenz für das Schreiben von Medienimages in Minuten

**IMGLOGLN**
Megabyte des Protokolls, das seit dem vorherigen Datenträgerimage eines Objekts geschrieben wurde.

Wenn Sie während des Tages eine kritische Zeit haben, wenn die Auslastung sehr schwer ist und Sie sicher sein wollen, dass der Systemdurchsatz nicht durch automatische Medienimages beeinträchtigt wird, können Sie die automatische Datenträgerabbildung vorübergehend ausschalten, indem Sie **IMGSCHED** *(MANUAL)* festlegen.

Sie können **IMGSCHED** zu einem beliebigen Zeitpunkt während der Workload wechseln.

**Achtung: MEDIALOG** wird nicht vorwärts bewegt, wenn Sie keine Datenträgerimages erstellen, d. h., Sie müssen entweder Speicherbereiche archivieren oder sicherstellen, dass ausreichend Plattenspeicherplatz vorhanden ist.

Sie können auch automatische und manuelle Medienimages für andere benutzerdefinierte Objekte steuern, indem Sie das Attribut **IMGRCOVO** verwenden:

- Authentifizierungsdaten
- Kanal
- Clientverbindung
- Empfangsprogramm

- Namensliste
- Prozess
- Aliaswarteschlange
- Lokale Warteschlange
- Service
- Thema

Für interne Systemobjekte, wie z. B. den Objektkatalog und das WS-Manager-Objekt, schreibt der Warteschlangenmanager automatisch Medienimages nach Bedarf.

Weitere Informationen zu den Attributen finden Sie unter ALTER QMGR .

Sie können auch automatische und manuelle Medienimages nur für lokale und permanente dynamische Warteschlangen aktivieren oder inaktivieren. Verwenden Sie dazu das Warteschlangenattribut **IMGRCOVQ** .

Weitere Informationen zum Attribut **IMGRCOVQ** finden Sie im Abschnitt ALTER QUEUES .

**Anmerkungen:**

1. Datenträgerimages werden nur unterstützt, wenn Sie die lineare Protokollierung verwenden. Wenn Sie automatische Medienimages aktiviert haben, aber die Umlaufprotokollierung verwenden, wird eine Fehlernachricht ausgegeben, und das Attribut "Automatische Datenträgerimages" des Warteschlangenmanagers ist inaktiviert.
2. Wenn Sie automatische Medienimages aktiviert haben, aber keine Frequenz, keine Minuten oder Megabyte des Protokolls angegeben haben, wird eine Fehlernachricht ausgegeben, und es werden keine automatischen Medienimages geschrieben.
3. Sie können ein Medienimage mit rcdmqimg manuell aufzeichnen, wenn Sie **IMGSCHED***(AUTO)*festgelegt haben.

   Auf diese Weise können Sie Medienimages zu einem Zeitpunkt aufnehmen, der für Ihr Unternehmen geeignet ist, z. B. wenn Ihr System ruhig ist. Bei der automatischen Datenträgerabbildung werden diese manuellen Medienimages berücksichtigt, da bei einem manuellen Datenträgerimage das Intervall und die Protokolllänge neu festgelegt werden, bevor das nächste automatische Datenträgerimage übernommen wird.
4. Ab IBM MQ 9.1.0 schreibt der Warteschlangenmanager nur noch persistente Nachrichten in Medienimages, keine nicht persistenten Nachrichten. Dadurch kann die Größe von Medienimages bei der Migration auf IBM MQ 9.1.0 oder höher reduziert werden.

## Legen Sie fest, wie IMGLOGLN und IMGINTVL festgelegt werden sollen.

V 9.3.4 Standardmäßig ist **IMGLOGLN** für andere Warteschlangenmanager als native HA-Warteschlangenmanager auf `off` gesetzt. (Native HA-Warteschlangenmanager werden erstellt, wobei **IMGLOGLN** auf den Wert von 25% des verfügbaren Speicherplatzes auf dem Datenträger gesetzt ist, auf den die Wiederherstellungsprotokolle geschrieben werden.)

V 9.3.4 Standardmäßig ist **IMGINTVL** auf 60 Minuten gesetzt. Das durch **IMGINTVL** angegebene Intervall wird berücksichtigt, wenn genügend neue Arbeit auf dem Warteschlangenmanager ausgeführt wurde, damit sich die Aufzeichnung eines neuen Image lohnt. Andernfalls wird die Aufnahme neuer Bilder verzögert.

Sie können die Werte von **IMGLOGLN** und **IMGINTVL** ändern, um die beste Lösung für Ihre Konfiguration zu erreichen. Machen Sie **IMGLOGLN** und **IMGINTVL** groß genug, sodass der Warteschlangenmanager nur einen Bruchteil seiner Zeit für die Aufzeichnung von Medienimages ausgibt, aber klein genug, damit:

- Beschädigte Objekte können in einem angemessenen Zeitraum wiederhergestellt werden und
- Klein genug, damit Ihr Log auf Ihre Festplatte passt, ohne dass der Speicherplatz frei ist.

Wenn Sie **IMGLOGLN** festlegen, besteht die gute Praxis darin, **IMGLOGLN** das Datenvolumen in Ihren Warteschlangen und die Datenrate der Workload zu oft zu hoch zu machen. Je größer Sie **IMGLOGLN**, um so weniger Zeit, als Ihr WS-Manager die Aufzeichnung von Medienimages verbringt.

Wenn Sie **IMGINTVL** festlegen, ist es ebenfalls sinnvoll, **IMGINTVL** die Zeit, die der Warteschlangenmanager benötigt, um ein Medienimage aufzuzeichnen, um ein Vielfaches der Zeit zu machen. Sie können herausfinden, wie lange es dauert, ein Medienimage aufzuzeichnen, indem Sie es manuell aufzeichnen.

Wenn Sie **IMGLOGLN** und **IMGINTVL** zu groß machen, kann die Wiederherstellung eines beschädigten Objekts sehr lange dauern, da alle Speicherbereiche seit dem letzten Datenträgerimage wiedergegeben werden müssen.

**IMGLOGLN** und **IMGINTVL** klein genug machen, damit die maximal zulässige Zeit zum Wiederherstellen eines beschädigten Objekts für Sie akzeptabel ist.

Wenn **IMGLOGLN** und **IMGINTVL** sehr groß werden, bedeutet dies, dass das Protokoll sehr groß wird, da Medienimages so selten aufgezeichnet werden.

**Achtung:** Stellen Sie sicher, dass sich ein Protokoll dieser Größe bequem in Ihrem Protokolldateisystem anpasst, da Ihre Workload zurückgesetzt wird, wenn das Protokolldateisystem vollständig gefüllt ist.

Sie können sowohl **IMGINTVL** als auch **IMGLOGLN** festlegen. Dies kann nützlich sein, um sicherzustellen, dass automatische Medienimages bei hoher Auslastung (gesteuert von **IMGLOGLN**) regelmäßig ausgeführt werden, aber gelegentlich, wenn die Workload sehr hell ist (gesteuert von **IMGINTVL**).

**IMGINTVL** und **IMGLOGLN** sind Ziele für das Intervall und die Protokolldatenlänge, zwischen denen automatische Medienimages ausgeführt werden.

Diese Attribute sollten nicht als festes Maximum oder Minimum angesehen werden. Tatsächlich kann der Warteschlangenmanager beschließen, ein automatisches Datenträgerimage früher zu planen, wenn der Warteschlangenmanager erkennt, dass es sich um eine wirklich gute Zeit handelt:

- Da die Warteschlange leer ist, ist die Datenträgerabbildung am effizientesten in Bezug auf die Leistung, und
- Ein Datenträgerimage wurde für eine Weile nicht aufgezeichnet.

Die Diskrepanz zwischen den automatischen Medienimages kann bei Bedarf etwas länger sein als bei **IMGINTVL** und **IMGLOGLN**.

Die Diskrepanz zwischen Medienimages kann größer sein als **IMGLOGLN** , wenn sich die Datenmenge in Warteschlangen auf **IMGLOGLN** nähert. Die Diskrepanz zwischen Medienimages kann größer sein als **IMGINTVL** , wenn **IMGINTVL** fast so lang ist, wie ein Medienimage aufgezeichnet wird.

Dies ist ein schlechtes Verfahren, da der WS-Manager einen Großteil seiner Zeit in der Aufzeichnung von Medienimages aufwendet.

Bei Verwendung der automatischen Aufzeichnung von Datenträgerimages zeichnet der Warteschlangenmanager ein Datenträgerimage für jedes Objekt und jede Warteschlange einzeln auf, so dass der Warteschlangenmanager die Intervall-und Protokolllänge zwischen den Images für jedes Objekt separat protokolliert.

Nach und nach wird die Aufzeichnung von Medienimages gestaffelt, statt Medienbilder für alle Objekte gleichzeitig aufzuzeichnen. Diese Staffelung verteilt den Leistungseinschlag der Aufzeichnung von Medienimages und ist ein weiterer Vorteil der automatischen Aufzeichnung von Medienimages über die manuelle Aufzeichnung.

## Datenträgerimages manuell-lineare Protokollierung

Wenn Sie ein Datenträgerimage einer Warteschlange aufzeichnen, werden alle persistenten Nachrichten aus dieser Warteschlange in das Protokoll geschrieben. Bei Warteschlangen, die große Mengen an Nachrichtendaten enthalten, ist dies das Schreiben einer großen Menge von Daten in das Protokoll, und dieser Prozess kann sich auf die Leistung des Systems auswirken, während es geschieht.

Das Aufzeichnen von Medienimages von anderen Objekten ist wahrscheinlich relativ schnell, da das Medienimage anderer Objekte keine Benutzerdaten enthält.

Sie müssen sorgfältig überlegen, wann die Medienimages von Warteschlangen aufgezeichnet werden sollen, damit der Prozess nicht in Ihre Spitzenauslastung eingreift.

Sie müssen das Datenträgerimage aller Objekte regelmäßig aufzeichnen, um den ältesten Protokollspeicherbereich zu aktualisieren, der für die Datenträgerwiederherstellung benötigt wird.

Ein guter Zeitpunkt, um das Medienimage einer Warteschlange aufzuzeichnen, ist, wenn sie leer ist, da an diesem Punkt keine Nachrichtendaten in das Protokoll geschrieben werden. Umgekehrt ist eine schlechte Zeit, wenn die Warteschlange sehr tief ist oder sehr große Nachrichten darauf hat.

Ein guter Zeitpunkt, um das Medienimage einer Warteschlange aufzuzeichnen, ist der Moment, in dem Ihr System ruhig ist; in der Erwägung, dass eine schlechte Zeit in der Spitzenauslastung ist. Wenn Ihre Auslastung um Mitternacht immer ruhig ist, können Sie z. B. jede Nacht die Aufzeichnung von Medienimages um Mitternacht festlegen.

Durch die Staffelung der Aufzeichnung jeder Ihrer Warteschlangen kann sich die Leistung ausbreiten und so lessen Auswirkungen auf die Leistung. Je länger es seit dem letzten Aufzeichnen von Medienimages ist, desto wichtiger wird es, sie aufzuzeichnen, da die Anzahl der Protokollspeicherbereiche, die für die Datenträgerwiederherstellung erforderlich sind, zunimmt.

**Anmerkung:** Bei einer Datenträgerwiederherstellung müssen alle erforderlichen Protokolldateien im Protokolldateiverzeichnis gleichzeitig verfügbar sein. Stellen Sie sicher, dass Sie regelmäßig Datenträgerimages von Objekten, die Sie wiederherstellen möchten, verwenden, um zu vermeiden, dass der Plattenspeicherplatz für alle erforderlichen Protokolldateien gehalten wird.

Um beispielsweise ein Medienimage aller Objekte in Ihrem Warteschlangenmanager zu erstellen, führen Sie den Befehl **rcdmqimg** wie in den folgenden Beispielen gezeigt aus:

Windows **unter Windows**

```
rcdmqimg -m QMNAME -t all *
```

Linux AIX **unter AIX and Linux**

```
rcdmqimg -m QMNAME -t all "*"
```

Wenn Sie **rcdmqimg** ausführen, wird die Folgenummer des Datenträgerprotokolls (LSN) weitergeleitet. Weitere Informationen zu Protokollfolgenummern finden Sie in „Speicherauszug für den Inhalt des Protokolls mit dem Befehl 'dmpmqlog' erstellen” auf Seite 720. **rcdmqimg** wird nicht automatisch ausgeführt und muss daher manuell oder über eine von Ihnen erstellte automatische Task ausgeführt werden. Weitere Informationen zu diesem Befehl finden Sie in den Informationen zu rcdmqimg und dmpmqlog .

Die manuelle Aufzeichnung von Medienimages mit **rcdmqimg** zur Verwaltung des Protokollspeicherbereichs ist nicht erforderlich, wenn Sie die lineare Protokollierung mit einer vom Warteschlangenmanager gesteuerten automatischen Mediendarstellung ausgewählt haben.

**Anmerkung:** Die Nachrichten AMQ7467 und AMQ7468 können auch zum Zeitpunkt der Ausführung des Befehls `rcdmqimg` ausgegeben werden.

## Partielle Medienimages

Es ist sinnvoll, IBM MQ-Nachrichten nur für Daten zu verwenden, die in naher Zukunft verarbeitet werden sollen, so dass sich jede Nachricht nur für einen relativ kurzen Zeitraum in einer Warteschlange befindet.

Umgekehrt ist es schlecht, wenn IBM MQ-Nachrichten verwendet werden, um Daten langfristig wie in einer Datenbank zu speichern.

Es ist außerdem eine gute Praxis, sicherzustellen, dass Ihre Warteschlangen relativ flach sind, und schlechte Praxis, tiefe Warteschlangen zu haben, deren Nachrichten lange Zeit in der Warteschlange enthalten sind.

Durch die folgenden Richtlinien können Sie den Warteschlangenmanager optimieren, um die Leistung der automatischen Aufzeichnung von Medienimages zu optimieren.

Das Aufzeichnen des Datenträgerabbilds einer leeren Warteschlange ist sehr effizient (unter einem Leistungspunkt), während das Medienimage einer Warteschlange mit einer großen Datenmenge sehr ineffizient ist, da alle Daten in das Protokoll in das Medienimage geschrieben werden müssen.

Für flache Warteschlangen mit kürzlich ersetzte Nachrichten kann der WS-Manager eine weitere Optimierung vornehmen.

Wenn alle Nachrichten, die sich derzeit in der Warteschlange befinden, in die jüngste Vergangenheit versetzt wurden, kann der Warteschlangenmanager möglicherweise das Datenträgerimage im Namen einer Zeit ( *Wiederherstellungspunkt* ) aufzeichnen, bevor alle Nachrichten gestellt wurden, und so das Image der leeren Warteschlange aufzeichnen können. Dieser Prozess ist sehr kostengünstig in Bezug auf die Leistung.

Wenn alle Nachrichten, die sich am Wiederherstellungspunkt in der Warteschlange befanden, anschließend erhalten wurden, müssen diese Nachrichten nicht im Datenträgerimage aufgezeichnet werden, da sie sich nicht mehr in der Warteschlange befinden.

Dies wird als *Teilmedienbild* bezeichnet. In dem unwahrscheinlichen Fall, dass die Warteschlange wiederhergestellt werden muss, werden alle Protokollsätze, die sich auf diese Warteschlange beziehen, seit dem letzten Datenträgerimage wiedergegeben, so dass alle kürzlich eingestellten Nachrichten zurückgeschrieben werden.

Selbst wenn einige Nachrichten in der Warteschlange am Wiederherstellungspunkt vorhanden sind, die sich derzeit in der Warteschlange befinden (und daher im Teilmedienbild aufgezeichnet werden müssen), ist es immer noch effizienter, dieses kleinere Teilmedienbild aufzuzeichnen, als ein vollständiges Datenträgerimage aller Nachrichten.

Sicherstellen, dass Nachrichten für einen kurzen Zeitraum in Warteschlangen verbleiben, wird die Leistung der automatischen Aufzeichnung von Medienimages wahrscheinlich verbessert.

*Überflüssige Protokolldateien ermitteln-nur lineare Protokollierung*
Für die Umlaufprotokollierung werden keine Daten aus dem Protokollverzeichnis gelöscht. Bei der Verwaltung von linearen Protokolldateien ist es wichtig zu wissen, welche Dateien gelöscht oder archiviert werden können. Diese Informationen helfen Ihnen bei dieser Entscheidung.

Verwenden Sie die Änderungszeiten des Dateisystems nicht, um die "älteren" Protokolldateien zu ermitteln. Verwenden Sie nur die Protokolldateinummer. Die Verwendung von Protokolldateien durch den Warteschlangenmanager folgt komplexen Regeln, einschließlich der Vorzuweisung und Formatierung von Protokolldateien, bevor sie benötigt werden. Es werden möglicherweise Protokolldateien mit Änderungszeiten angezeigt, die irreführend wären, wenn Sie versuchen, diese Zeiten für die Bestimmung des relativen Alters zu verwenden.

Um die älteste benötigte Protokolldatei zu ermitteln, stehen Ihnen drei Stellen zur Verfügung:

- Befehl DISPLAY QMSTATUS
- Ereignisnachrichten der Protokollfunktion und schließlich
- Fehlernachrichten protokollieren

Für den Befehl ANZEIGEN QMSTATUS, um den ältesten Protokollspeicherbereich zu ermitteln, der für folgende Schritte erforderlich ist:

- Starten Sie den Warteschlangenmanager erneut, setzen Sie den Befehl `DISPLAY QMSTATUS RECLOG` ab.
- Datenträgerwiederherstellung ausführen. Geben Sie den Befehl `DISPLAY QMSTATUS MEDIALOG` aus.
- Bestimmen Sie den Namen für die Archivierungsbenachrichtigung, indem Sie den Befehl `DISPLAY QMSTATUS ARCHLOG`ausgeben.

Sie können die Anzahl der sekundären Protokollspeicherbereiche verringern, wenn Sie die Umlaufprotokollierung verwenden, indem Sie den Befehl **RESET QMGR TYPE(REDUCELOG)** ausgeben.

Im Allgemeinen impliziert eine niedrigere Protokolldateinummer ein älteres Protokoll. Wenn Sie nicht über einen sehr hohen Protokolldateiumschlag verfügen, die Reihenfolge von 3000 Protokolldateien pro Tag für 10 Jahre, müssen Sie für den Zahlenumbruch bei 9.999 999 nicht mehr umstellen. In diesem Fall können Sie jede Protokolldatei mit einer Zahl, die kleiner als der RECLOG-Wert ist, archivieren, und Sie können jede Protokolldatei mit einer Zahl löschen, die kleiner ist als die Werte für RECLOG und MEDIALOG.

**Achtung:** Die Protokolldatei wird umgebrochen, so dass die nächste Zahl nach 9 999 999 null ist.

*Position der Protokolldatei*
Denken Sie bei der Auswahl einer Position für Ihre Protokolldateien daran, dass der Betrieb stark beeinträchtigt wird, wenn IBM MQ aufgrund fehlenden Plattenspeicherplatzes ein neues Protokoll nicht formatieren kann.

Wenn Sie ein Umlaufprotokoll verwenden, stellen Sie sicher, dass auf dem Laufwerk genügend Speicherplatz für mindestens die konfigurierten primären Protokolldateien vorhanden ist. Geben Sie außerdem Speicherplatz für mindestens eine sekundäre Protokolldatei an, die benötigt wird, wenn das Protokoll wachsen muss.

Wenn Sie ein lineares Protokoll verwenden, können Sie erheblich mehr Speicherplatz in Anspruch haben. Der Speicherplatz, der vom Protokoll verbraucht wird, nimmt kontinuierlich zu, wenn Daten protokolliert werden.

Sie sollten die Protokolldateien auf einem separaten Plattenlaufwerk aus den WS-Manager-Daten platzieren.

Die Datenintegrität auf diesem Gerät ist von größter Bedeutung-Sie sollten die Redundanz bauen lassen.

Es kann auch möglich sein, die Protokolldateien auf mehreren Plattenlaufwerken in einer spiegelgleichen Anordnung zu platzieren. Dies schützt vor einem Ausfall des Laufwerks, das das Protokoll enthält. Ohne Spiegelung können Sie gezwungen sein, auf die letzte Sicherung Ihres IBM MQ-Systems zurückzugreifen.

### *Kaltstart: Maßnahmen bei fehlenden oder beschädigten Protokollspeicherbereichen*

Wenn in Ihrem Unternehmen einige oder alle Protokollspeicherbereiche, die für die Fehlerbehebung des Neustarts benötigt werden, verloren gehen, kann der Warteschlangenmanager das Wiederherstellungsprotokoll nicht wiedergeben und und daher nicht erneut gestartet werden. Wenn Sie Ihren Warteschlangenmanager erneut starten müssen und das Wiederherstellungsprotokoll in irgendeiner Form beschädigt ist, ist dies unter Gefährdung der Datenintegrität möglich, hiervon wird jedoch strikt abgeraten. Dieser Prozess wird als *Kaltstart* eines Warteschlangenmanagers bezeichnet.

**Wichtig:** Der Kaltstart eines Warteschlangenmanagers sollte nur in Ausnahmesituationen in Betracht gezogen werden, denn er birgt Datenintegritätsrisiken, wie auf dieser Seite beschrieben. IBM empfiehlt bei beschädigten Datendateien statt eines Kaltstarts eine Neuerstellung des Warteschlangenmanagers.

Wenn aus betrieblichen Gründen ein Kaltstart erforderlich ist, bitten Sie den IBM Supportmitarbeiter, die zugrunde liegende Ursache des Problems zu überprüfen. Nach einem Kaltstart sollte der betreffende Warteschlangenmanager so bald wie möglich durch einen neu erstellten Warteschlangenmanager ersetzt werden.

## Die Auswirkungen eines Kaltstarts

Beim Kaltstart erstellt der Warteschlangenmanager ein leeres Wiederherstellungsprotokoll und stützt sich auf die Daten in den Warteschlangendateien und anderen Objektdateien in ihrem vorhandenen Zustand. Da die Daten in den Warteschlangendateien inkonsistent sein können, besteht die Gefahr, dass Nachrichten verloren gehen, dupliziert oder beschädigt werden oder nicht mehr konsistent sind.

Der Warteschlangenmanager speichert die Konfiguration aller anderen dauerhaft festgelegten Objekte im Wiederherstellungsprotokoll sowie in Objektdateien. Auch andere interne Statusdaten werden im

Wiederherstellungsprotokoll aufgezeichnet. Bei einem Kaltstart werden somit die internen Statusdaten zurückgesetzt und alle anderen Konfigurationsdaten sind möglicherweise nicht mehr korrekt.

Die Auswirkungen eines Kaltstarts sind unvorhersehbar und vielfältig, daher sollten Sie einen Kaltstart vermeiden, wenn er nicht unbedingt erforderlich ist. Nach dem Kaltstart können die Inkonsistenzen der Informationen in den Warteschlangen- und Objektdateien so groß sein, dass der Warteschlangenmanager gar nicht erneut gestartet wird.

Falls der Warteschlangenmanager erneut gestartet wird, gibtes keine einfache Möglichkeit, herauszufinden, welche Nachrichtendaten oder Konfigurationen zuverlässig sind und welche nicht. Nach einem Kaltstart kann es auch passieren, dass Warteschlangen beschädigt sind und somit völlig unbrauchbar werden.

Wenn Sie Nachrichten aus einer bestimmten Warteschlange abrufen oder in eine bestimmte Warteschlange stellen können, kann es zudem sein, dass die Nachrichten beschädigt sind, fehlen oder doppelt vorhanden sind. Transaktionen und Kanäle können unbestätigt bleiben. Selbst wenn der Kaltstart Ihres Warteschlangenmanagers erfolgreich ist und die Warteschlangen intakt aussehen, besteht die Gefahr, dass sich die unvorhersehbaren Auswirkungen des Kaltstarts erst viel später zeigen.

## Vorgehensweise, wenn ein Kaltstart notwendig ist

Kaltstarts sollte nicht als standardmäßige betriebliche Praxis betrachtet werden und IBM rät dringend davon ab. Falls Sie jedoch definitiv einen Kaltstart eines Warteschlangenmanagers durchführen müssen, wenden Sie sich an den Unterstützung für IBM MQ .

Der Vorgang für den Kaltstart eines Warteschlangenmanagers war bisher für einen linearen Warteschlangenmanager sehr viel komplizierter als für einen Umlaufwarteschlangenmanager. In IBM MQ 9.1.3 wurde das Kaltstartverfahren stark vereinfacht. Es müssen keine Protokollspeicherbereiche mehr kopiert oder umbenannt werden.

Wenden Sie sich ab IBM MQ 9.1.3 an den IBM Support. Dieser wird Ihnen einen Schlüssel bereitstellen, den Sie für den Kaltstart des Warteschlangenmanagers an den Befehl **strmqm** übergeben.

**Achtung:** Der Befehl IBM MQ 9.1.3 Kaltstart führt immer noch die gleichen Risiken aus, um die Datenintegrität als manueller Kaltstart zu verlieren, und IBM rät dringend davon ab, dies zu tun.

## Vermeidung zukünftiger Kaltstarts: Anfrage

Der strmqm-Befehl benötigt für den Kaltstart einen Schlüssel, da IBM MQ möchte, dass Sie sich an den IBM MQ Support wenden, falls ein Kaltstart nötig ist. IBM MQ möchte verstehen, wie Sie in diese Situation geraten sind.

Ein Kaltstart ist ganz klar unbedingt zu vermeiden. IBM MQ hat erhebliche Anstrengungen unternommen, um einen Kaltstart Ihres Warteschlangenmanagers unnötig zu machen, und IBM möchte gerne wissen, ob noch mehr getan werden kann, damit noch seltener ein Kaltstart notwendig ist.

## Vorkehrungen zur Vermeidung eines Kaltstarts

Die Standardprotokollierungsmethode beim Erstellen eines Warteschlangenmanagers ist die Umlaufprotokollierung. Bei der Umlaufprotokollierung gewähren Sie dem Warteschlangenmanager eine gewisse Anzahl primärer und sekundärer Protokollspeicherbereiche einer bestimmten Größe. Erstellen Sie Ihr Protokolldateisystem groß genug für alle primären und sekundären Protokollspeicherbereiche, dann sollten Sie sie nie verwalten müssen.

Alternativ zur Umlaufprotokollierung können Sie die lineare Protokollierung verwenden. Die lineare Protokollierung bietet Ihnen die zusätzliche Möglichkeit, Warteschlangen und andere Objekte wiederherzustellen, falls der unwahrscheinliche Fall eintritt und sie beschädigt werden. Standardmäßig müssen Sie bei der linearen Protokollierung Protokollspeicherbereiche, die nicht mehr für den Neustart oder die Datenträgerwiederherstellung benötigt werden, allerdings löschen. Dies wird als manuelle Protokollverwaltung bezeichnet.

Bei einer solchen Verwaltung der Protokollspeicherbereiche kann es passieren, dass versehentlich zu viele Protokollspeicherbereiche gelöscht werden, sodass letztendlich ein Kaltstart notwendig ist. Um dieses Risiko zu mildern, sollten Sie die automatische Protokollverwaltung verwenden, sodass der Warteschlangenmanager die Protokollspeicherbereiche für Sie verwaltet.

Ein bewährtes Verfahren besteht darin, das Wiederherstellungsprotokoll in ein separates Protokolldateisystem zu stellen, das nur das Wiederherstellungsprotokoll enthält. Wenn Sie das Wiederherstellungsprotokoll in dasselbe Dateisystem wie die übrigen Dateien des Warteschlangenmanagers stellen, kann es gelegentlich vorkommen, dass sich dieses Dateisystem unbeabsichtigt füllt, vielleicht aufgrund von großen Warteschlangendateien. Machen Sie entweder das Protokollverzeichnis für den Warteschlangenmanager zu einem separaten Dateisystem oder geben Sie mit der Befehlszeilenoption **-ld** im Befehl **crtmqm** ein anderes Protokolldateisystem an.

Wenn das Dateisystem mit den Warteschlangendateien voll wird, können Sie möglicherweise keine Nachrichten in diese Warteschlangen einreihen, der Warteschlangenmanager wird jedoch weiterhin ausgeführt. Wenn das Dateisystem, in dem das Wiederherstellungsprotokoll enthalten ist, voll wird, wird der Warteschlangenmanager abrupt beendet und erst erneut gestartet, wenn Sie Speicherplatz freigeben.

Achten Sie darauf, keine Protokollspeicherbereiche zu löschen, die für die Neustartwiederherstellung benötigt werden, da andernfalls ein Kaltstart notwendig werden könnte. Manchmal kann es sein, dass Sie einen Kaltstart durchführen müssen, da die Platte, auf der das Wiederherstellungsprotokoll enthalten ist, defekt ist. Ein bewährtes Verfahren besteht darin, das Wiederherstellungsprotokoll auf einer replizierten Platte zu speichern und so das Risiko eines Ausfalls der Platte zu mindern.

Durch Verschieben Ihrer Nachrichten und der Konfiguration auf einen neuen Ersatzwarteschlangenmanager können fortlaufende Probleme mit einem Warteschlangenmanager vermieden werden, für den zuvor ein Kaltstart durchgeführt wurde.

Notieren Sie sich, für welche Warteschlangenmanager bereits ein Kaltstart durchgeführt wurde, selbst wenn dies vor langer Zeit geschehen ist und die Warteschlangenmanager in der Zwischenzeit gestoppt, erneut gestartet und migriert wurden. Wenn Sie sich an den IBM Support wenden, geben Sie an, ob für den Warteschlangenmanager bereits ein Kaltstart durchgeführt wurde, und falls ja, erklären Sie möglichst genau, warum der Kaltstart notwendig war.

## Protokoll für Wiederherstellung verwenden

Sie können Informationen aus den Protokollen verwenden, um Sie bei der Wiederherstellung nach Fehlern zu unterstützen.

Es gibt mehrere Möglichkeiten, Ihre Daten zu beschädigen. IBM MQ unterstützt Sie bei der Wiederherstellung in folgenden Fällen:

- Ein beschädigtes Datenobjekt
- Ein Stromausfall im System
- Ein Kommunikationsfehler

In diesem Abschnitt wird erläutert, wie die Protokolle für die Wiederherstellung nach diesen Problemen verwendet werden.

### *Wiederherstellung nach Stromausfall-oder Kommunikationsfehlern*

IBM MQ kann sowohl bei Übertragungsfehlern als auch bei einem Stromausfall wiederhergestellt werden. Es kann sich auch manchmal von anderen Problemtypen wie z. B. unbeabsichtigtes Löschen einer Datei erholen.

Bei einem Kommunikationsfehler bleiben persistente Nachrichten in den Warteschlangen, bis sie von einer empfangenden Anwendung entfernt werden. Wenn die Nachricht übertragen wird, bleibt sie in der Übertragungswarteschlange, bis sie erfolgreich übertragen werden kann. Für die Wiederherstellung nach einem Kommunikationsfehler können Sie die Kanäle in der Regel über den fehlgeschlagenen Link erneut starten.

Wenn die Stromversorgung ausfällt, stellt IBM MQ beim Neustart des Warteschlangenmanagers die Warteschlangen wieder in ihrem festgeschriebenen Status zum Zeitpunkt des Fehlers wieder her. Dadurch

wird sichergestellt, dass keine persistenten Nachrichten verloren gehen. Nicht persistente Nachrichten werden gelöscht; sie überstehen einen abrupten Stopp von IBM MQ nicht.

### *Beschädigte Objekte wiederherstellen*

Es gibt Fälle, in denen ein IBM MQ-Objekt unbrauchbar werden kann, z. B. wegen unbeabsichtigtes Beschädigung. Sie müssen dann entweder Ihr vollständiges System oder einen Teil davon wiederherstellen. Die erforderliche Aktion hängt davon ab, wann der Schaden erkannt wird, ob die ausgewählte Protokollmethode die Datenträgerwiederherstellung unterstützt und welche Objekte beschädigt sind.

## Datenträgerwiederherstellung

Sie können Datenträgerimages für Objekte aufzeichnen, damit sie bei einer Beschädigung wiederhergestellt werden können. Diese Funktion ist nur für Warteschlangenmanager verfügbar, die die lineare Protokollierung oder die replizierte Protokollierung verwenden, und für die lineare Protokollierung nur für Objekte, die als wiederherstellbar definiert sind. Sie definieren, dass Objekttypen wiederherstellbar sind, indem Sie die Warteschlangenmanager-Attribute **IMGRCOVO** und **IMGRCOVQ** verwenden (siehe ALTER QMGR). Wenn ein Objekt, das nicht als wiederherstellbar definiert ist, beschädigt ist, sind die Wiederherstellungsoptionen dieselben wie bei der Umlaufprotokollierung.

Die Datenträgerwiederherstellung erstellt Objekte aus Informationen, die in einem linearen Protokoll oder replizierten Protokollaufgezeichnet wurden, neu. Wenn beispielsweise eine Objektdatei versehentlich gelöscht oder aus einem anderen Grund nicht mehr verwendet werden kann, kann die Datenträgerwiederherstellung erneut erstellt werden. Die Informationen in dem Protokoll, die für die Datenträgerwiederherstellung eines Objekts erforderlich sind, werden als *Datenträgerimage* bezeichnet.

Ein Datenträgerimage ist eine Folge von Protokollsätzen, die ein Bild eines Objekts enthalten, aus dem das Objekt selbst neu erstellt werden kann.

Der erste Protokollsatz, der zum erneuten Erstellen eines Objekts erforderlich ist, wird als *Datenträgerwiederherstellungssatz*bezeichnet; er ist der Anfang des neuesten Medienimages für das Objekt. Der Datenträgerwiederherstellungsdatensatz eines jeden Objekts ist einer der Informationsteile, die während eines Prüfpunkts aufgezeichnet wurden.

Wenn ein Objekt aus seinem Medienimage neu erstellt wird, müssen auch alle Protokollsätze wiedergegeben werden, die die Aktualisierungen beschreiben, die seit dem letzten Abbild für das Objekt ausgeführt wurden.

Betrachten Sie zum Beispiel eine lokale Warteschlange, die ein Image des Warteschlangenobjekts enthält, bevor eine persistente Nachricht in die Warteschlange gestellt wird. Um das aktuellste Image des Objekts erneut zu erstellen, müssen die Protokolleinträge, die das Einreihen der Nachricht in die Warteschlange aufzeichnen, neu wiedergegeben werden, und die Wiedergabe des Images selbst wird nicht mehr angezeigt.

Wenn ein Objekt erstellt wird, enthalten die geschriebenen Protokollsätze genügend Informationen, um das Objekt vollständig neu zu erstellen. Diese Datensätze bilden das erste Medienimage des Objekts. Anschließend zeichnet der Warteschlangenmanager bei jedem Systemabschluss die Datenträgerimages automatisch wie folgt auf:

- Images aller Prozessobjekte und Warteschlangen, die nicht lokal sind
- Images von leeren lokalen Warteschlangen

Medienimages können auch manuell mit dem Befehl **rcdmqimg** aufgezeichnet werden, wie in rcdmqimgbeschrieben. Dieser Befehl schreibt ein Datenträgerimage des IBM MQ-Objekts.

Der Warteschlangenmanager zeichnet Datenträgerimages automatisch auf, wenn **IMGSCHED** *(AUTO)* festgelegt ist. Weitere Informationen zu **IMGINTVL** und **INGLOGLN**finden Sie unter ALTER QMGR .

Wenn ein Datenträgerimage geschrieben wurde, sind nur die Protokolle, die das Datenträgerimage enthalten, und alle nach diesem Zeitpunkt erstellten Protokolle erforderlich, um beschädigte Objekte erneut zu erstellen. Der Vorteil der Erstellung von Datenträgerimages hängt von Faktoren wie der Menge des verfügbaren freien Speichers und der Geschwindigkeit ab, mit der Protokolldateien erstellt werden.

## Wiederherstellung von Medienimages

Ein Warteschlangenmanager stellt einige Objekte während des Starts des Warteschlangenmanagers automatisch von ihrem Medienimage wieder her. Eine Warteschlange wird automatisch wiederhergestellt, wenn sie an einer Transaktion beteiligt war, die beim letzten Abschalten des Warteschlangenmanagers unvollständig war und bei der Neustartverarbeitung als beschädigt oder fehlerhaft erkannt wird.

Sie müssen andere Objekte manuell wiederherstellen, indem Sie den Befehl **rcrmqobj** verwenden, der die Datensätze im Protokoll wiedergibt, um das IBM MQ-Objekt erneut zu erstellen. Das Objekt wird aus dem zuletzt im Protokoll gefundenen Image zusammen mit allen anwendbaren Protokollereignissen zwischen dem Zeitpunkt, zu dem das Image gespeichert wurde, und dem Zeitpunkt, zu dem der Befehl zum Erstellen der Neuerstellungs-Datei ausgegeben wurde, erneut erstellt. Wenn ein IBM MQ-Objekt beschädigt wird, sind die einzigen gültigen Aktionen, die ausgeführt werden können, entweder das Löschen oder die erneute Erstellung durch diese Methode. Nicht persistente Nachrichten können auf diese Weise nicht wiederhergestellt werden.

Weitere Informationen zum Befehl **rcrmqobj** finden Sie unter rcrmqobj .

Die Protokolldatei, die den Datenträgerwiederherstellungssatz enthält, und alle nachfolgenden Protokolldateien müssen im Protokolldateiverzeichnis verfügbar sein, wenn die Datenträgerwiederherstellung eines Objekts versucht wird. Wenn eine erforderliche Datei nicht gefunden werden kann, wird die Bedienernachricht AMQ6767 ausgegeben und die Datenträgerwiederherstellungsoperation schlägt fehl. Wenn Sie keine regulären Medienimages der Objekte verwenden, die Sie erneut erstellen möchten, haben Sie möglicherweise nicht genügend Plattenspeicherplatz, um alle Protokolldateien zu speichern, die für die erneute Erstellung eines Objekts erforderlich sind.

V 9.3.3 Native HA-Warteschlangenmanager verwenden die replizierte Protokollierung. Solche Warteschlangenmanager versuchen, auswählbare Objekte automatisch wiederherzustellen, wenn eine Beschädigung erkannt wird. Nach dem Start versuchen native HA-Warteschlangenmanager standardmäßig automatisch eine asynchrone Wiederherstellung, wenn eine Objektbeschädigung erkannt wird. Eine Wiederherstellung ist möglicherweise nicht sofort möglich, wenn beispielsweise das Objekt von einer Anwendung verwendet wird oder die für die Datenträgerwiederherstellung erforderlichen Protokollspeicherbereiche nicht verfügbar sind. In diesen Situationen wird die asynchrone Wiederherstellung in regelmäßigen Abständen wiederholt. Wenn das Problem, das die Wiederherstellung verhindert hat, behoben ist, wird das Objekt bei der nächsten Wiederholung wiederhergestellt oder das Objekt kann mit dem Befehl **rcrmqobj** manuell wiederhergestellt werden.

## Welche Objektdateien vorhanden sind

Der Warteschlangenmanager speichert die Attribute von Objekten, die in **runmqsc** definiert sind, in Dateien auf Platte. Diese Objektdateien befinden sich in Unterverzeichnissen unter dem Datenverzeichnis des Warteschlangenmanagers.

Linux AIX Auf AIX and Linux-Plattformen werden Kanäle beispielsweise in `/var/mqm/qmgrs/`*`qmgr`*`/channel` gespeichert.

Bei den Daten in diesen Objektdateien handelt es sich um das Datenträgerimage der Objekte. Wenn diese Objektdateien gelöscht oder beschädigt werden, ist das in dieser Datei gespeicherte Objekt beschädigt. Bei Verwendung eines linearen Protokollierungswarteschlangenmanagers können beschädigte Objekte aus dem Protokoll mit dem Befehl rcrmqobj wiederhergestellt werden. Replizierte Protokollierungswarteschlangenmanager (native HA) versuchen automatisch, beschädigte Objekte wiederherzustellen, wenn sie erkannt werden.

Die meisten Objektdateien enthalten nur die Attribute des Objekts, so dass Kanaldateien die Attribute von Kanälen enthalten. Die Ausnahmen sind:

- Katalog

  Der Objektkatalog katalogisiert alle Objekte aller Typen und wird in `qmanager/QMQMOBJCAT` gespeichert.
- Syncfiles

Die Datei syncfile enthält interne Statusdaten, die allen Kanälen zugeordnet sind.

- Warteschlangen

  Warteschlangendateien enthalten sowohl die Nachrichten in dieser Warteschlange als auch die Attribute dieser Warteschlange.

Beachten Sie, dass in **runmqsc** oder im IBM MQ Explorer kein Katalog- oder Syncfile-Objekt verfügbar ist.

Der Katalog und der WS-Manager können aufgezeichnet, aber nicht wiederhergestellt werden. Wenn diese Objekte beschädigt werden, wird der WS-Manager präventiv beendet, und diese Objekte werden beim Neustart automatisch wiederhergestellt.

Subskriptionen werden nicht in Objekten aufgelistet, die aufgezeichnet oder wiederhergestellt werden sollen, da permanente Subskriptionen in einer Systemwarteschlange gespeichert werden. Um permanente Subskriptionen aufzuzeichnen oder wiederherzustellen, müssen Sie stattdessen die Warteschlange SYSTEM.DURABLE.SUBSCRIBER.QUEUE aufzeichnen oder wiederherstellen.

## Wiederanlauf beschädigter Objekte beim Start

Wenn der Warteschlangenmanager während des Starts ein beschädigtes Objekt erkennt, hängt die von ihm abhängige Aktion vom Typ des Objekts und davon ab, ob der Warteschlangenmanager für die Unterstützung der Datenträgerwiederherstellung konfiguriert ist.

Wenn das WS-Manager-Objekt beschädigt ist, kann der Warteschlangenmanager nicht gestartet werden, es sei denn, er kann das Objekt wiederherstellen. Wenn der Warteschlangenmanager mit einem linearen Protokoll konfiguriert ist und somit die Datenträgerwiederherstellung unterstützt, versucht IBM MQ automatisch, das WS-Manager-Objekt aus seinen Medienimages erneut zu erstellen. Wenn die ausgewählte Protokollmethode keine Datenträgerwiederherstellung unterstützt, können Sie entweder eine Sicherung des Warteschlangenmanagers zurückschreiben oder den Warteschlangenmanager löschen.

Wenn Transaktionen aktiv waren, als der Warteschlangenmanager gestoppt wurde, sind auch die lokalen Warteschlangen, die die persistenten, nicht festgeschriebenen Nachrichten enthalten oder in diese Transaktionen eingingen, erforderlich, um den WS-Manager erfolgreich zu starten. Wenn eine dieser lokalen Warteschlangen beschädigt ist und der Warteschlangenmanager die Datenträgerwiederherstellung unterstützt, versucht er automatisch, sie aus ihren Medienimages erneut zu erstellen. Wenn eine der Warteschlangen nicht wiederhergestellt werden kann, kann IBM MQ nicht gestartet werden.

Wenn beschädigte lokale Warteschlangen, die nicht festgeschriebene Nachrichten enthalten, während der Startverarbeitung auf einem Warteschlangenmanager erkannt werden, der keine Datenträgerwiederherstellung unterstützt, werden die Warteschlangen als beschädigte Objekte markiert und die nicht festgeschriebenen Nachrichten in ihnen werden ignoriert. Dies liegt daran, dass es nicht möglich ist, die Datenträgerwiederherstellung beschädigter Objekte in einem solchen Warteschlangenmanager auszuführen, und die einzige Aktion, die noch vorhanden ist, besteht darin, sie zu löschen. Die Nachricht AMQ7472 wird ausgegeben, um alle Schäden zu melden.

## Beschädigte Objekte zu einem anderen Zeitpunkt wiederherstellen

Die Medienwiederherstellung von Objekten erfolgt nur beim Start automatisch (außer für native HA-Warteschlangenmanager, die standardmäßig die automatische Wiederherstellung verwenden). Wenn zu anderen Zeiten eine Objektbeschädigung erkannt wird, wird die Bedienernachricht AMQ7472 ausgegeben und die meisten Operationen, die das Objekt verwenden, schlagen mit dem Rückkehrcode MQRC_OBJECT_DAMAGED fehl. Wenn das Warteschlangenmanagerobjekt zu einem beliebigen Zeitpunkt nach dem Start des Warteschlangenmanagers beschädigt ist, führt der Warteschlangenmanager einen präventiven Systemabschluss durch. Wenn ein Objekt beschädigt wurde, können Sie es löschen oder, wenn der Warteschlangenmanager ein lineares Protokoll verwendet, versuchen, es mit dem Befehl **rcrmqobj** aus seinem Datenträgerimage wiederherzustellen (weitere Informationen finden Sie unter rcrmqobj ).

Wenn eine Warteschlange (oder ein anderes Objekt) beschädigt wird, wird **MEDIALOG** nicht vorwärts verschoben. Dies liegt daran, dass **MEDIALOG** der älteste Speicherbereich ist, der für die Datenträgerwiederherstellung erforderlich ist. Wenn Ihre Workload fortgesetzt wird, wird **CURRLOG** weiterhin vorwärts verschoben, sodass neue Speicherbereiche geschrieben werden. Abhängig von Ihrer Konfiguration (ein-

schließlich Ihrer **LogManagement** -Einstellung) wird möglicherweise das Füllen des Protokolldateisystems gestartet. Wenn das Protokolldateisystem vollständig gefüllt ist, werden die Transaktionen rückgängig gemacht, und der WS-Manager wird möglicherweise abrupt beendet. Wenn eine Warteschlange beschädigt wird, haben Sie möglicherweise nur eine begrenzte Zeit, um zu handeln, bevor Ihr Warteschlangenmanager beendet wird. Wie viel Zeit Sie haben, hängt von der Geschwindigkeit ab, mit der Ihre Auslastung den Warteschlangenmanager dazu veranlasst, neue Speicherbereiche zu schreiben, und die Größe des freien Speicherbereichs, den Sie in Ihrem Protokolldateisystem haben.

Wenn Sie die manuelle Protokollverwaltung verwenden, können Sie möglicherweise Speicherbereiche archivieren, die nicht für die Wiederherstellung nach einem Neustart benötigt werden, und sie dann aus dem Protokolldateisystem löschen, auch wenn sie noch für die Datenträgerwiederherstellung benötigt werden. Dies ist akzeptabel, solange Sie sie bei Bedarf aus dem Archiv wiederherstellen können. Diese Richtlinie führt nicht dazu, dass Ihr Protokolldateisystem gefüllt wird, wenn eine Warteschlange beschädigt wird, und **MEDIALOG** stoppt die Vorwärts-Bewegung. Wenn Sie jedoch nur Speicherbereiche archivieren und löschen, die weder für einen Neustart noch für die Wiederherstellung von Datenträgern erforderlich sind, beginnt Ihr Protokolldateisystem mit dem Ausfüllen, wenn eine Warteschlange beschädigt wird.

Wenn Sie die automatische Verwaltung oder die Verwaltung von Archivprotokolldateien verwenden, verwendet der Warteschlangenmanager keine Speicherbereiche, die noch für die Datenträgerwiederherstellung benötigt werden. Dies gilt auch dann, wenn Sie sie archiviert haben und den Warteschlangenmanager mit SET LOG ARCHIVED benachrichtigt haben. Wenn also eine Warteschlange beschädigt wird, beginnt Ihr Protokolldateisystem mit dem Ausfüllen.

Wenn eine Warteschlange beschädigt wird, werden OBJECT DAMAGED FFDCs geschrieben, und **MEDIALOG** stoppt die Weiterleitung. Das beschädigte Objekt kann anhand der FFDC identifiziert werden oder weil es das Objekt mit dem ältesten **MEDIALOG** ist, wenn Sie seinen Status in **runmqsc**anzeigen.

Wenn Ihr Protokolldateisystem gefüllt wird und Sie besorgt sind, dass Ihre Workload zurückgesetzt wird, weil das Protokolldateisystem voll wird, dann wird das Objekt wiederhergestellt oder die Stilllegung der Workload kann dazu führen, dass diese Vorgänge nicht mehr stattfinden.

V 9.3.3 Bei nativen HA-Warteschlangenmanagern (die die replizierte Protokollierung verwenden) wird die automatische Wiederherstellung beschädigter Objekte versucht. Nach dem Start versuchen native HA-Warteschlangenmanager standardmäßig automatisch eine asynchrone Wiederherstellung, wenn eine Objektbeschädigung erkannt wird. Eine Wiederherstellung ist möglicherweise nicht sofort möglich, wenn beispielsweise das Objekt von einer Anwendung verwendet wird oder die für die Datenträgerwiederherstellung erforderlichen Protokollspeicherbereiche nicht verfügbar sind. In diesen Situationen wird die asynchrone Wiederherstellung in regelmäßigen Abständen wiederholt. Wenn das Problem, das die Wiederherstellung verhindert hat, behoben ist, wird das Objekt bei der nächsten Wiederholung wiederhergestellt oder das Objekt kann mit dem Befehl **rcrmqobj** manuell wiederhergestellt werden.

## IBM MQ-Protokolldateien schützen

Berühren Sie die Protokolldateien nicht, wenn ein Warteschlangenmanager ausgeführt wird. Eine Wiederherstellung ist möglicherweise nicht möglich. Verwenden Sie die Superuser-oder die mqm-Berechtigung zum Schutz von Protokolldateien vor unbeabsichtigtes Ändern.

Entfernen Sie die aktiven Protokolldateien nicht manuell, wenn ein IBM MQ-Warteschlangenmanager ausgeführt wird. Wenn ein Benutzer die Protokolldateien, die ein Warteschlangenmanager erneut starten muss, versehentlich löscht, gibt IBM MQ **keine** Fehler aus und setzt die Verarbeitung der Daten *einschließlich persistenter Nachrichten* fort. Der WS-Manager wird normal beendet, kann aber nicht erneut gestartet werden. Die Wiederherstellung von Nachrichten wird dann nicht mehr möglich.

Benutzer mit der Berechtigung zum Entfernen von Protokollen, die von einem aktiven Warteschlangenmanager verwendet werden, haben auch die Berechtigung zum Löschen anderer wichtiger WS-Manager-Ressourcen (z. B. Warteschlangendateien, Objektkatalog und ausführbare IBM MQ-Dateien). Sie können daher beispielsweise aus Unerfahrenheit einen aktiven oder ruhenden WS-Manager in einer Weise beschädigen, die IBM MQ nicht selbst schützen kann.

Gehen Sie bei der Übertragung von Superuser-oder mqm-Berechtigungen vorsichtig vor.

## Speicherauszug für den Inhalt des Protokolls mit dem Befehl 'dmpmqlog' erstellen

Verwendung des Befehls dmpmqlog zum Erstellen eines Speicherauszugs für den Inhalt des Warteschlangenmanagerprotokolls.

Verwenden Sie den Befehl dmpmqlog , um den Inhalt des Warteschlangenmanagerprotokolls zu erstellen. Standardmäßig wird ein Speicherauszug aller aktiven Protokollsätze erstellt, d. a. der Befehl startet das Erstellen eines Speicherauszugs aus dem Protokollkopf (in der Regel der Anfang des letzten abgeschlossenen Prüfpunkts).

Für das Protokoll kann in der Regel nur ein Speicherauszug erstellt werden, wenn der Warteschlangenmanager nicht aktiv ist. Da der Warteschlangenmanager während der Beendigung einen Prüfpunkt nimmt, enthält der aktive Teil des Protokolls normalerweise eine kleine Anzahl Protokollsätze. Sie können jedoch den Befehl dmpmqlog verwenden, um weitere Protokollsätze mit einer der folgenden Optionen zu erstellen, um die Startposition des Speicherauszugs zu ändern:

- Starten Sie das Dumping von der *Basis* des Protokolls. Die Basis des Protokolls ist der erste Protokollsatz in der Protokolldatei, der den Kopf des Protokolls enthält. Die Menge der zusätzlichen Daten, die in diesem Fall erstellt werden, hängt davon ab, wo sich der Protokollkopf in der Protokolldatei befindet. Wenn die Protokolldatei am Anfang der Protokolldatei steht, wird nur ein kleiner Teil der zusätzlichen Daten erstellt. Befindet sich der Kopf am Ende der Protokolldatei, wird ein Speicherauszug für deutlich mehr Daten erstellt.
- Geben Sie die Startposition des Speicherauszugs als einzelnen Protokollsatz an. Jeder Protokollsatz wird durch eine eindeutige *Protokollfolgenummer (Log Sequence Number, LSN)* identifiziert. Bei der Umlaufprotokollierung kann dieser Startprotokollsatz nicht vor der Basis des Protokolls liegen. Diese Einschränkung gilt nicht für lineare Protokolle. Sie müssen möglicherweise inaktive Protokolldateien erneut instanziieren, bevor Sie den Befehl ausführen. Sie müssen eine gültige Protokollfolgenummer (LSN) angeben, die aus der vorherigen Ausgabe des Befehls dmpmqlog als Startposition übernommen wurde.

  Mit der linearen Protokollierung können Sie beispielsweise den nextlsn aus der letzten Ausgabe des Befehls dmpmqlog angeben. Der nextlsn wird in Log File Header angezeigt und gibt die Protokollfolgenummer des nächsten Protokollsatzes an, der geschrieben werden soll. Verwenden Sie diese Funktion als Startposition, um alle Protokollsätze zu formatieren, die seit der letzten Erstellung des Protokollspeicherauszugs geschrieben wurden.
- **Nur für lineare Protokolle** : Sie können dmpmqlog anweisen, mit dem Formatieren von Protokollsätzen aus allen angegebenen Protokolldateiausdehnung zu beginnen. In diesem Fall erwartet dmpmqlog , dass diese Protokolldatei und jede aufeinanderfolgende Protokolldatei im selben Verzeichnis wie die aktiven Protokolldateien gefunden werden. Diese Option gilt nicht für Umlaufprotokolle, wobei dmpmqlog nicht auf Protokollsätze vor der Basis des Protokolls zugreifen kann.

Die Ausgabe des Befehls dmpmqlog ist die Log File Header und eine Reihe formatierter Protokollsätze. Der WS-Manager verwendet mehrere Protokollsätze, um Änderungen an seinen Daten zu erfassen.

Ein Teil der Informationen, die formatiert werden, ist nur intern verwendet. Die folgende Liste enthält die nützlichsten Protokollsätze:

**Protokolldateiheader**
Jedes Protokoll verfügt über einen einzigen Protokolldateiheader, der immer die erste ist, die mit dem Befehl dmpmqlog formatiert wird. Sie enthält die folgenden Felder:

| | |
|---|---|
| *logactive* | Die Anzahl primärer Protokollspeicherbereiche. |
| *loginactive* | Die Anzahl sekundärer Protokollerweiterungen. |
| *logsize* | Die Anzahl der 4-KB-Seiten pro Speicherbereich. |
| *baselsn* | Die erste Protokollfolgenummer (LSN) in der Protokollspeicherbereichsdatei, die den Protokollkopf enthält. |

*nextlsn* Die Protokollfolgenummer (LSN) des nächsten Protokollsatzes, der geschrieben werden soll.

*headlsn* Die Protokollfolgenummer (LSN) des Protokollsatzes am Kopf des Protokolls.

*tailsn* Die Protokollfolgenummer (LSN), die die Endposition des Protokolls angibt.

*hflag1* Gibt an, ob das Protokoll CIRCULAR oder LOG RETAIN (linear) ist.

*HeadExtentID* Die Protokollspeicherbereichsdatei, die den Protokollkopf enthält.

**Protokollsatz-Header**
Jeder Protokollsatz innerhalb des Protokolls hat einen festen Header mit den folgenden Informationen:

*LSN* Die Protokollfolgenummer.

*LogRecdType* Der Typ des Protokollsatzes.

*XTranid* Die Transaktions-ID, die diesem Protokollsatz zugeordnet ist (falls vorhanden).

Ein *TranType*-Wert von 'MQI' zeigt eine auf IBM MQ beschränkte Transaktion an. Ein *TranType* von XA ist an anderen Ressourcenmanagern beteiligt. Aktualisierungen, die in derselben Arbeitseinheit enthalten sind, haben denselben *XTranid*.

*QueueName* Die Warteschlange, die diesem Protokollsatz zugeordnet ist (falls vorhanden).

*Qid* Die eindeutige interne Kennung für die Warteschlange.

*PrevLSN* Die Protokollfolgenummer (LSN) des vorherigen Protokollsatzes innerhalb derselben Transaktion (falls vorhanden).

**WS-Manager starten**
Diese Protokolle, die der Warteschlangenmanager gestartet hat.

*StartDate* Das Datum, an dem der WS-Manager gestartet wurde.

*StartTime* Die Zeit, zu der der WS-Manager gestartet wurde.

**Warteschlangenmanager stoppen**
Diese Protokolle, die der Warteschlangenmanager gestoppt hat, wurden gestoppt.

*StopDate* Das Datum, an dem der WS-Manager gestoppt wurde.

*StopTime* Die Zeit, zu der der WS-Manager gestoppt wurde.

*ForceFlag* Der Typ des verwendeten Systemabschlusses.

**Prüfpunkt starten**
Gibt den Start eines Warteschlangenmanagerprüfpunkts an.

**Prüfpunkt beenden**
Gibt das Ende eines Warteschlangenmanagerprüfpunkts an.

*ChkPtLSN* Die Protokollfolgenummer (LSN) des Protokollsatzes, der diesen Prüfpunkt gestartet hat.

**Nachricht einreihen**
Diese Nachricht protokolliert eine persistente Nachricht, die in eine Warteschlange gestellt wird. Wenn die Nachricht unter Synchronisationspunkt gesetzt wurde, enthält der Protokollsatz-Header einen Nicht-Nullwert *XTranid*. Der Rest des Datensatzes enthält:

| | |
|---|---|
| *MapIndex* | Eine Kennung für die Nachricht in der Warteschlange. Sie kann dazu verwendet werden, die entsprechende MQGET -Nachricht abzugleichen, die zum Abrufen dieser Nachricht aus der Warteschlange verwendet wurde. In diesem Fall kann ein nachfolgender *Get Message* -Protokollsatz gefunden werden, der die gleichen QueueName und MapIndex enthält. Zu diesem Zeitpunkt kann die MapIndex -ID für eine nachfolgende Nachricht in diese Warteschlange wiederverwendet werden. |
| *Daten* | Im Hex-Speicherauszug für diesen Protokollsatz sind verschiedene interne Daten enthalten, gefolgt von einer Darstellung des Nachrichtendeskriptors (eyecatcher MD) und dann der Nachrichtendaten selbst. |

**Teil einreihen**
Persistente Nachrichten, die zu groß für einen einzelnen Protokollsatz sind, werden als mehrere *Put Part* -Protokollsätze gefolgt von einem einzelnen *Put Message* -Datensatz protokolliert. Wenn *Put Part* Datensätze vorhanden sind, wird das Feld *PrevLSN* die *Put Part* -Datensätze und den endgültigen *Put Message* -Datensatz miteinander verketten.

| | |
|---|---|
| *Daten* | Setzt die Nachrichtendaten da fort, wo der vorherige Protokollsatz aufgehört hatte. |

**Nachricht abrufen**
Es werden nur persistente Nachrichten protokolliert. Wenn die Nachricht unter Synchronisationspunkt wurde, enthält der Protokollsatz-Header einen Nicht-Nullwert *XTranid*. Der Rest des Datensatzes enthält:

| | |
|---|---|
| *MapIndex* | Bezeichnet die Nachricht, die aus der Warteschlange abgerufen wurde. Der aktuellste *Put Message* -Protokollsatz, der denselben *QueueName* und *MapIndex* enthält, gibt die Nachricht an, die abgerufen wurde. |
| *QPriority* | Die Priorität der Nachricht, die aus der Warteschlange abgerufen wurde. |

**Transaktion starten**
Gibt den Start einer neuen Transaktion an. Ein TranType-Wert von 'MQI' zeigt eine auf IBM MQ beschränkte Transaktion an. Ein TranType von XA gibt an, dass andere Ressourcenmanager involviert sind. Alle Aktualisierungen, die von dieser Transaktion vorgenommen werden, haben denselben *XTranid*.

**Transaktion vorbereiten**
Gibt an, dass der WS-Manager bereit ist, die Aktualisierungen festzuschreiben, die dem angegebenen *XTranid* zugeordnet sind. Dieser Protokollsatz wird als Teil einer zweiphasigeCommit mit anderen Ressourcenmanagern geschrieben.

**Commit-Transaktion**
Gibt an, dass der Warteschlangenmanager alle Aktualisierungen festgeschrieben hat, die von einer Transaktion vorgenommen wurden.

**Rollback-Transaktion**
Gibt die Absicht des WS-Managers an, eine Transaktion rückgängig zu machen.

**Transaktion beenden**
Gibt das Ende einer rückgängig gewickelten Transaktion an.

**Transaktionstabelle**
Dieser Datensatz wird während des Synchronisationspunkts geschrieben. Er zeichnet den Status jeder Transaktion auf, die persistente Aktualisierungen vorgenommen hat. Für jede Transaktion werden die folgenden Informationen aufgezeichnet:

| | |
|---|---|
| *XTranid* | Die Transaktions-ID. |
| *FirstLSN* | Die Protokollfolgenummer (LSN) des ersten Protokollsatzes, der der Transaktion zugeordnet ist. |
| *LastLSN* | Die Protokollfolgenummer (LSN) des letzten Protokollsatzes, der der Transaktion zugeordnet ist. |

**Transaktionsteilnehmer**
Dieser Protokollsatz wird von der Komponente "XA Transaction Manager" des Warteschlangenmanagers geschrieben. Es zeichnet die externen Ressourcenmanager auf, die an Transaktionen beteiligt sind. Für jeden Teilnehmer werden die folgenden Informationen aufgezeichnet:

| | |
|---|---|
| *RMName* | Der Name des Ressourcenmanagers. |
| *RMID* | Die Kennung des Ressourcenmanagers. Dies wird auch in nachfolgenden *Transaction Prepared* -Protokollsätzen protokolliert, in denen globale Transaktionen aufgezeichnet werden, an denen der Ressourcenmanager beteiligt ist. |
| *SwitchFile* | Die Switchloaddatei für diesen Ressourcenmanager. |
| *XAOpenString* | Die offene XA-Zeichenfolge für diesen Ressourcenmanager. |
| *XACloseString* | Die XA-Zeichenfolge für diesen Ressourcenmanager. |

**Transaktion vorbereitet**
Dieser Protokollsatz wird von der Komponente "XA Transaction Manager" des Warteschlangenmanagers geschrieben. Sie zeigt an, dass die angegebene globale Transaktion erfolgreich vorbereitet wurde. Jeder der beteiligten Ressourcenmanager wird angewiesen, sich festzuschreiben. Der *RMID* jedes vorbereiteten Ressourcenmanagers wird im Protokollsatz aufgezeichnet. Wenn der WS-Manager selbst an der Transaktion beteiligt ist, wird ein *Participant Entry* mit einem *RMID* von null vorhanden sein.

**Transaktionsforget**
Dieser Protokollsatz wird von der Komponente "XA Transaction Manager" des Warteschlangenmanagers geschrieben. Sie folgt dem *Transaction Prepared* -Protokollsatz, wenn die Festschreibungsentscheidung an jeden Teilnehmer zugestellt wurde.

**Bereinigungs-warteschlange**
In diesem Protokoll wird die Tatsache protokolliert, dass alle Nachrichten in einer Warteschlange gelöscht wurden, z. B. mit dem MQSC-Befehl CLEAR QUEUE.

**Warteschlangenattribute**
Dadurch wird die Initialisierung oder Änderung der Attribute einer Warteschlange protokolliert.

**Objekt erstellen**
Hiermit wird das Erstellen eines IBM MQ-Objekts protokolliert.

| | |
|---|---|
| *ObjName* | Der Name des Objekts, das erstellt wurde. |
| *UserId* | Die Benutzer-ID, die die Erstellung ausführt. |

**Objekt löschen**
Hiermit wird das Löschen eines IBM MQ-Objekts protokolliert.

| | |
|---|---|
| *ObjName* | Der Name des Objekts, das gelöscht wurde. |

## IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen

Sie können Warteschlangenmanager vor möglichen Beschädigungen durch Hardwarefehler schützen, indem Sie Warteschlangenmanager und WS-Manager-Daten sichern, nur die Konfiguration des Warteschlangenmanagers sichern und einen Sicherungswarteschlangenmanager verwenden.

## Informationen zu diesem Vorgang

**Vorsicht:** Sie müssen sehr vorsichtig sein, wenn Sie einen WS-Manager in ein anderes Betriebssystem verschieben. Weitere Informationen finden Sie im Abschnitt Warteschlangenmanager in ein anderes Betriebssystem verschieben .

In regelmäßigen Abständen können Sie Maßnahmen ergreifen, um WS-Manager vor möglichen Beschädigungen durch Hardwarefehler zu schützen. Es gibt drei Möglichkeiten zum Schutz eines Warteschlangenmanagers:

**Sichern Sie die WS-Manager-Daten.**
Wenn die Hardware fehlschlägt, wird möglicherweise ein WS-Manager gestoppt. Wenn die Protokolldaten eines Warteschlangenmanagers aufgrund des Hardwarefehls verloren gehen, kann der WS-Manager möglicherweise nicht erneut gestartet werden. Wenn Sie WS-Manager-Daten sichern, können Sie möglicherweise einige oder alle verloren gegangenen WS-Manager-Daten wiederherstellen.

Im Allgemeinen sind die weniger Daten, die Sie im Falle eines Hardwarefehls verloren haben, bei einem Hardwarefehler, der zu einem Verlust der Integrität des Wiederherstellungsprotokolls führt, im Allgemeinen die weniger Daten, die Sie sichern.

Damit WS-Manager-Daten gesichert werden können, muss der Warteschlangenmanager nicht aktiv sein.

**Nur die Konfiguration des WS-Managers sichern**
Wenn die Hardware fehlschlägt, wird möglicherweise ein WS-Manager gestoppt. Wenn sowohl die Konfiguration des WS-Managers als auch die Protokolldaten aufgrund des Hardwarefehls verloren gehen, kann der Warteschlangenmanager nicht erneut gestartet oder aus dem Protokoll wiederhergestellt werden. Wenn Sie die WS-Manager-Konfiguration sichern, können Sie den Warteschlangenmanager und alle seine Objekte aus gespeicherten Definitionen erneut erstellen.

Damit die Warteschlangenmanagerkonfiguration gesichert werden kann, muss der Warteschlangenmanager aktiv sein.

**Sicherungswarteschlangenmanager verwenden**
Wenn der Hardwarefehler schwer wiegend ist, kann ein Warteschlangenmanager möglicherweise nicht wiederhergestellt werden. Wenn der nicht wiederherstellbare Warteschlangenmanager in dieser Situation über einen dedizierten Sicherungswarteschlangenmanager verfügt, kann der Sicherungswarteschlangenmanager an Stelle des nicht wiederherstellbaren Warteschlangenmanagers aktiviert werden. Wenn die Datei regelmäßig aktualisiert wird, kann das Sicherungs-WS-Managerprotokoll Protokolldaten enthalten, die das letzte vollständige Protokoll des nicht wiederherstellbaren Warteschlangenmanagers enthalten.

Ein Sicherungswarteschlangenmanager kann aktualisiert werden, während der vorhandene WS-Manager noch aktiv ist.

## Prozedur

- Informationen zum Sichern und Wiederherstellen von WS-Manager-Daten finden Sie unter:
  - „WS-Manager-Daten sichern” auf Seite 724.
  - „WS-Manager-Daten zurückschreiben” auf Seite 726.
- Informationen zum Sichern und Wiederherstellen der WS-Manager-Konfiguration finden Sie unter:
  - „WS-Manager-Konfiguration sichern” auf Seite 726
  - „Warteschlangenmanagerkonfiguration wird zurückgespeichert” auf Seite 727
- Informationen zum Erstellen, Aktualisieren und Starten eines Backup-Warteschlangenmanagers finden Sie in „Sicherungswarteschlangenmanager verwenden” auf Seite 728.

# WS-Manager-Daten sichern

Das Sichern von WS-Manager-Daten kann Ihnen dabei helfen, den möglichen Datenverlust durch Hardwarefehler zu schützen.

## Vorbereitende Schritte

Bevor Sie mit der Sicherung des Warteschlangenmanagers beginnen, müssen Sie sicherstellen, dass der Warteschlangenmanager nicht aktiv ist. Wenn Sie versuchen, eine Sicherung eines aktiven Warteschlangenmanagers zu erstellen, ist die Sicherung möglicherweise nicht konsistent, da die Aktualisierungen in Bearbeitung sind, wenn die Dateien kopiert werden. Wenn möglich, stoppen Sie Ihren Warteschlangenmanager, indem Sie den Befehl **endmqm -w** (wait shutdown) ausführen. Verwenden Sie nur dann, wenn dies fehlschlägt, den Befehl **endmqm -i** (immediate shutdown).

## Informationen zu diesem Vorgang

Führen Sie die folgenden Tasks aus, um eine Sicherungskopie der Daten eines Warteschlangenmanagers zu erstellen:

## Vorgehensweise

1. Suchen Sie nach den Verzeichnissen, unter denen der WS-Manager seine Daten und seine Protokolldateien mit den Informationen in den Konfigurationsdateien platziert.

   Weitere Informationen finden Sie unter „IBM MQ -Konfigurationsdaten in INI-Dateien auf Multiplatforms ändern” auf Seite 90.

   **Anmerkung:** Die Namen, die in dem Verzeichnis angezeigt werden, werden umgesetzt, um sicherzustellen, dass sie mit der Plattform kompatibel sind, auf der Sie IBM MQ verwenden. Weitere Informationen zu Namensumsetzungen finden Sie im Artikel IBM MQ-Dateinamen verstehen.
2. Übernehmen Sie Kopien aller Daten-und Protokolldateiverzeichnisse des Warteschlangenmanagers, einschließlich aller Unterverzeichnisse.

   Stellen Sie sicher, dass Sie keine Dateien, insbesondere die Protokollsteuerdatei, wie in „Wie Logs aussehen” auf Seite 691 beschrieben, und die Konfigurationsdateien, wie im Abschnitt „Initialisierungs- und Konfigurationsdateien” auf Seite 260 beschrieben, vergessen. Einige der Verzeichnisse sind möglicherweise leer, aber Sie benötigen alle Verzeichnisse, um die Sicherung zu einem späteren Zeitpunkt zurückschreiben zu können.

   Sichern Sie bei der Umlaufprotokollierung die WS-Manager-Daten und die Protokolldateiverzeichnisse gleichzeitig, so dass Sie eine konsistente Gruppe von WS-Manager-Daten und -Protokollen wiederherstellen können.

   Zur linearen Protokollierung sichern Sie die WS-Manager-Daten und die Protokolldateiverzeichnisse zur gleichen Zeit. Es ist möglich, nur die WS-Manager-Datendateien zurückzuspeichern, wenn eine entsprechende vollständige Folge von Protokolldateien verfügbar ist.
3. Die Eignerschaften der Dateien beibehalten.

   Linux AIX Bei IBM MQ for UNIX- und Linux-Systemen können Sie dies mit dem Befehl **tar** tun. (Wenn Sie Warteschlangen mit mehr als 2 GB haben, können Sie den Befehl **tar** nicht verwenden. Weitere Informationen hierzu finden Sie im Abschnitt Große Warteschlangen aktivieren .

   **Anmerkung:** Wenn Sie ein Upgrade auf IBM WebSphere MQ 7.5 und höher durchführen, müssen Sie sicherstellen, dass eine Sicherung der `qm.ini`-Datei und der Registrierungseinträge erstellt wird. Die Warteschlangenmanager-Informationen werden in der `qm.ini`-Datei gespeichert und können für die Zurücksetzung auf eine frühere Version von IBM MQ verwendet werden.

### Zugehörige Tasks

Stoppen eines Warteschlangenmanagers

„Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern” auf Seite 15
Die Konfigurationsinformationen für IBM MQ werden unter AIX, Linux, and Windows in Konfigurationsdateien gespeichert. Sichern Sie nach der Erstellung eines Warteschlangenmanagers Ihre Konfigurationsdateien. Wenn Sie dann einen anderen WS-Manager erstellen, der Probleme verursacht, können Sie die Sicherungen erneut erstellen, wenn Sie die Ursache des Problems entfernt haben.

## WS-Manager-Daten zurückschreiben

Führen Sie die folgenden Schritte aus, um eine Sicherung der Daten eines Warteschlangenmanagers wiederherzustellen.

### Vorbereitende Schritte

Bevor Sie die Sicherung starten, stellen Sie sicher, dass der Warteschlangenmanager nicht aktiv ist.

Wenn Sie eine Sicherung eines Warteschlangenmanagers in einem Cluster wiederherstellen, finden Sie weitere Informationen unter „Cluster-WS-Manager wiederherstellen" auf Seite 397 und Clustering: Verfügbarkeit, mehrere Instanzen und Disaster-Recovery .

**Anmerkung:** Wenn Sie ein Upgrade auf eine höhere Version von IBM MQdurchführen, müssen Sie eine Sicherung der Datei **.ini** und der Registry-Einträge erstellen. Die Warteschlangenmanager-Informationen werden in der **.ini**-Datei gespeichert und können für die Zurücksetzung auf eine frühere Version von IBM MQ verwendet werden.

### Vorgehensweise

1. Suchen Sie die Verzeichnisse, unter denen der WS-Manager seine Daten und seine Protokolldateien platziert, indem Sie die Informationen in den Konfigurationsdateien verwenden.
2. Leeren Sie die Verzeichnisse, in die Sie die Sicherungsdaten stellen wollen.
3. Kopieren Sie die Daten des Sicherungswarteschlangenmanagers und die Protokolldateien an die richtigen Stellen.

   Stellen Sie sicher, dass Sie über eine Protokollsteuerdatei sowie über die Protokolldateien verfügen.

   Sichern Sie bei der Umlaufprotokollierung die WS-Manager-Daten und die Protokolldateiverzeichnisse gleichzeitig, so dass Sie eine konsistente Gruppe von WS-Manager-Daten und -Protokollen wiederherstellen können.

   Zur linearen Protokollierung sichern Sie die WS-Manager-Daten und die Protokolldateiverzeichnisse zur gleichen Zeit. Es ist möglich, nur die WS-Manager-Datendateien zurückzuspeichern, wenn eine entsprechende vollständige Folge von Protokolldateien verfügbar ist.
4. Aktualisieren Sie die Konfigurationsinformationsdateien.

   Überprüfen Sie, ob die Konfigurationsdateien von IBM MQ und des Warteschlangenmanagers konsistent sind, damit IBM MQ die wiederhergestellten Daten an den richtigen Stellen suchen kann.
5. Überprüfen Sie die sich ergebende Verzeichnisstruktur, um sicherzustellen, dass alle erforderlichen Verzeichnisse vorhanden sind.

   Weitere Informationen zu IBM MQ-Verzeichnissen und Unterverzeichnissen finden Sie im Abschnitt Verzeichnisstruktur auf Windows-Systemen und Verzeichnisinhalt auf AIX and Linux-Systemen.

### Ergebnisse

Wenn die Daten ordnungsgemäß gesichert und wiederhergestellt wurden, wird der WS-Manager jetzt gestartet.

## Multi WS-Manager-Konfiguration sichern

Wenn Sie die Konfiguration des Warteschlangenmanagers sichern, können Sie einen Warteschlangenmanager aus seinen Definitionen erneut erstellen, wenn sowohl die Warteschlangenmanager-Konfigurations- als auch die Protokolldaten aufgrund des Hardwarefehlerfehlers verloren gehen und der Warteschlangenmanager nicht erneut gestartet werden kann oder aus dem Protokoll wiederhergestellt werden kann.

### Informationen zu diesem Vorgang

ALW Unter AIX, Linux, and Windows können Sie den Befehl **dmpmqcfg** verwenden, um einen Speicherauszug für die Konfiguration eines IBM MQ-Warteschlangenmanagers zu erstellen.

IBM i Unter IBM i können Sie den Befehl **DMPMQMCFG** (Speicherauszug für MQ-Konfiguration ausgeben) verwenden, um die Konfigurationsobjekte und die Berechtigungen für einen Warteschlangenmanager zu erstellen.

## Vorgehensweise

1. Stellen Sie sicher, dass der WS-Manager aktiv ist.
2. Verwenden Sie abhängig von Ihrer Plattform einen der folgenden Befehle, um die Konfiguration des Warteschlangenmanagers zu sichern:
   - ALW Unter AIX, Linux, and Windows: Führen Sie den Befehl 'Dump MQ Configuration' (**dmpmqcfg**) unter Verwendung der Standardformatierungsoption (-f mqsc) MQSC und aller Attribute (-a) aus, verwenden Sie die Standardausgabeumleitung, um die Definition in einer Datei zu speichern. For example:

     ```
     dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
     ```

   - IBM i Unter IBM i: Führen Sie den Befehl **DMPMQMCFG** (Speicherauszug für MQ-Konfiguration) mit der Standardformatierungsoption OUTPUT (*MQSC) und EXPATTR (*ALL) aus, und verwenden Sie TOFILE und TOMBR, um die Definitionen in einer physischen Teildatei zu speichern. For example:

     ```
     DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC) TOMBR(MYQMGR
     DEF)
     ```

**Zugehörige Tasks**

„Warteschlangenmanagerkonfiguration wird zurückgespeichert” auf Seite 727
Sie können die Konfiguration für einen Warteschlangenmanager aus einer Sicherung wiederherstellen, indem Sie zuerst sicherstellen, dass der Warteschlangenmanager ausgeführt wird, und anschließend den entsprechenden Befehl für Ihre Plattform ausführen.

**Zugehörige Verweise**

dmpmqcfg (Konfiguration des Speicherauszugs-WS-Managers)
Speicherauszugs-MQ-Konfiguration (DMPMQMCFG)

# Multi Warteschlangenmanagerkonfiguration wird zurückgespeichert

Sie können die Konfiguration für einen Warteschlangenmanager aus einer Sicherung wiederherstellen, indem Sie zuerst sicherstellen, dass der Warteschlangenmanager ausgeführt wird, und anschließend den entsprechenden Befehl für Ihre Plattform ausführen.

## Informationen zu diesem Vorgang

ALW Unter AIX, Linux, and Windows können Sie den Befehl **runmqsc** verwenden, um die Konfiguration eines IBM MQ-Warteschlangenmanagers wiederherzustellen.

IBM i Unter IBM i können Sie den Befehl **STRMQMMQSC** verwenden, um die Konfigurationsobjekte und die Berechtigungen für einen Warteschlangenmanager wiederherzustellen.

## Vorgehensweise

1. Stellen Sie sicher, dass der WS-Manager aktiv ist.

   Beachten Sie, dass der Warteschlangenmanager möglicherweise erneut erstellt wurde, wenn die Beschädigung der Daten und Protokolle durch andere Mittel nicht behoben werden kann.
2. Verwenden Sie abhängig von Ihrer Plattform einen der folgenden Befehle, um die WS-Manager-Konfiguration wiederherzustellen:

- ALW Führen Sie unter AIX, Linux, and Windows den Befehl **runmqsc** für den Warteschlangenmanager aus und verwenden Sie die Standardeingabeumleitung, um die Definitionen aus einer Scriptdatei wiederherzustellen, die vom Befehl **dmpmqcfg** (Speicherauszug MQ-Konfiguration) generiert wird (siehe „WS-Manager-Konfiguration sichern” auf Seite 726). For example:

  ```
  runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
  ```

- IBM i Unter IBM i: Führen Sie **STRMQMMQSC** für den Warteschlangenmanager aus, und verwenden Sie die Parameter **SRCMBR** und **SRCFILE**, um die Definitionen aus der physischen Teildatei zurückzuschreiben, die vom Befehl **DMPMQMCFG** (Speicherauszug für MQ-Konfiguration ausgeben) generiert wird (siehe „WS-Manager-Konfiguration sichern” auf Seite 726). For example:

  ```
  STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
  ```

**Zugehörige Tasks**

„WS-Manager-Konfiguration sichern” auf Seite 726
Wenn Sie die Konfiguration des Warteschlangenmanagers sichern, können Sie einen Warteschlangenmanager aus seinen Definitionen erneut erstellen, wenn sowohl die Warteschlangenmanager-Konfigurations- als auch die Protokolldaten aufgrund des Hardwarefehlerfehlers verloren gehen und der Warteschlangenmanager nicht erneut gestartet werden kann oder aus dem Protokoll wiederhergestellt werden kann.

**Zugehörige Verweise**

dmpmqcfg (Konfiguration des Speicherauszugs-WS-Managers)
runmqsc (MQSC-Befehle ausführen)
Speicherauszugs-MQ-Konfiguration (DMPMQMCFG)
IBM MQ-Befehle starten (STRMQMMQSC)

## Sicherungswarteschlangenmanager verwenden

Ein vorhandener Warteschlangenmanager kann einen dedizierten Sicherungswarteschlangenmanager für Notfallwiederherstellungszwecke verwenden.

### Informationen zu diesem Vorgang

Ein Sicherungswarteschlangenmanager ist eine inaktive Kopie des vorhandenen Warteschlangenmanagers. Wenn der vorhandene Warteschlangenmanager aufgrund eines schwerwiegenden Hardwarefehers nicht wiederhergestellt werden kann, kann der Sicherungswarteschlangenmanager online gebracht werden, um den nicht wiederherstellbaren WS-Manager zu ersetzen.

Die vorhandenen Warteschlangenmanagerprotokolldateien müssen regelmäßig in den Sicherungswarteschlangenmanager kopiert werden, um sicherzustellen, dass der Sicherungswarteschlangenmanager eine effektive Methode für die Wiederherstellung nach einem Katastrophenfall bleibt. Der vorhandene Warteschlangenmanager muss nicht gestoppt werden, damit Protokolldateien kopiert werden können. Sie sollten eine Protokolldatei jedoch nur kopieren, wenn der Warteschlangenmanager den Schreibvorgang beendet hat. Informationen dazu, wie sichergestellt wird, dass eine bestimmte Protokolldatei nicht mehr geschrieben wird, damit sie sicher kopiert werden kann, finden Sie im Abschnitt „Sicherungswarteschlangenmanager aktualisieren” auf Seite 730 .

**Anmerkung:** Da das vorhandene Warteschlangenmanagerprotokoll ständig aktualisiert wird, gibt es immer eine geringfügige Diskrepanz zwischen dem vorhandenen WS-Managerprotokoll und den Protokolldaten, die in das Sicherungs-WS-Managerprotokoll kopiert werden. Durch regelmäßige Aktualisierungen des Sicherungswarteschlangenmanagers wird die Diskrepanz zwischen den beiden Protokollen minimiert.

Wenn ein Sicherungswarteschlangenmanager benötigt wird, um ihn online zu stellen, muss er aktiviert und dann gestartet werden. Die Voraussetzung, einen Sicherungswarteschlangenmanager vor dem Start zu aktivieren, ist eine vorbeugende Maßnahme zum Schutz vor einem versehentlichen Start eines Sicherungswarteschlangenmanagers. Nachdem ein Sicherungswarteschlangenmanager aktiviert wurde, kann er nicht mehr aktualisiert werden.

**Wichtig:** Sobald der alte Sicherungswarteschlangenmanager zum neuen aktiven Warteschlangenmanager geworden ist, gibt es aus welchem Grund auch immer keinen Sicherungswarteschlangenmanager mehr. Dies ist eine effektive Form der asynchronen Replikation. Daher wird erwartet, dass der neue aktive Warteschlangenmanager eine logische Zeit hinter dem alten aktiven WS-Manager hat. Der alte aktive WS-Manager fungiert daher nicht mehr als Sicherung für den neuen aktiven Warteschlangenmanager.

## Prozedur

- Informationen zur Verwendung eines Sicherungswarteschlangenmanagers finden Sie in den folgenden Abschnitten:
  - „Sicherungswarteschlangenmanager erstellen” auf Seite 729
  - „Sicherungswarteschlangenmanager aktualisieren” auf Seite 730
  - „Sicherungswarteschlangenmanager starten” auf Seite 730

**Zugehörige Konzepte**

„Protokollierung: Stellen Sie sicher, dass die Nachrichten nicht verloren gehen.” auf Seite 691
IBM MQ zeichnet in einem Wiederherstellungsprotokoll alle signifikanten Änderungen an den persistenten Daten auf, die vom WS-Manager gesteuert werden.

## *Sicherungswarteschlangenmanager erstellen*

Sie erstellen einen Sicherungswarteschlangenmanager als inaktive Kopie des vorhandenen Warteschlangenmanagers.

## Informationen zu diesem Vorgang

**Wichtig:** Sie können einen Sicherungswarteschlangenmanager nur verwenden, wenn Sie die lineare Protokollierung verwenden.

Für einen Sicherungswarteschlangenmanager ist Folgendes erforderlich:

- Damit werden dieselben Attribute wie der vorhandene Warteschlangenmanager verwendet, z. B. der Name des Warteschlangenmanagers, der Protokollierungstyp und die Größe der Protokolldatei.
- Befinden Sie sich auf derselben Plattform wie der vorhandene Warteschlangenmanager.
- Die Codeversion ist gleich oder höher als die Codeversion des vorhandenen Warteschlangenmanagers.

## Vorgehensweise

1. Erstellen Sie mit dem Steuerbefehl **crtmqm**einen Sicherungswarteschlangenmanager für den vorhandenen Warteschlangenmanager.
2. Übernehmen Sie Kopien aller vorhandenen Daten- und Protokolldateiverzeichnisse des vorhandenen Warteschlangenmanagers, einschließlich aller Unterverzeichnisse, wie in „WS-Manager-Daten sichern” auf Seite 724 beschrieben.
3. Überschreiben Sie die Daten-und Protokolldateiverzeichnisse des Sicherungswarteschlangenmanagers, einschließlich aller Unterverzeichnisse, mit den Kopien, die aus dem vorhandenen Warteschlangenmanager übernommen wurden.
4. Führen Sie den Steuerbefehl **strmqm** auf dem Sicherungswarteschlangenmanager wie im folgenden Beispiel gezeigt aus:

   ```
   strmqm -r BackupQMName
   ```

   Dieser Befehl markiert den Warteschlangenmanager als Sicherungswarteschlangenmanager in IBM MQ und gibt alle kopierten Protokollextents zurück, um den Sicherungswarteschlangenmanager in Schritt mit dem vorhandenen Warteschlangenmanager zu bringen.

**Zugehörige Verweise**

crtmqm (WS-Manager erstellen)
strmqm (Warteschlangenmanager starten)

## *Sicherungswarteschlangenmanager aktualisieren*

Um sicherzustellen, dass ein Sicherungswarteschlangenmanager eine effektive Methode für die Wiederherstellung nach einem Katastrophenfall bleibt, muss er regelmäßig aktualisiert werden.

### Informationen zu diesem Vorgang

Durch die regelmäßige Aktualisierung wird die Diskrepanz zwischen dem Backup-WS-Manager-Protokoll und dem aktuellen WS-Manager-Protokoll verringert. Es ist nicht erforderlich, den Warteschlangenmanager zu stoppen, bevor Sie ihn sichern.

**Warnung:** Wenn Sie eine nicht zusammenhängende Gruppe von Protokollen in das Protokollverzeichnis des Sicherungswarteschlangenmanagers kopieren, werden nur die Protokolle bis zu dem Punkt, an dem das erste fehlende Protokoll gefunden wird, wiedergegeben.

### Vorgehensweise

1. Geben Sie den folgenden Scriptbefehl (MQSC) für den Warteschlangenmanager aus, der gesichert werden soll:

   ```
   RESET QMGR TYPE(ADVANCELOG)
   ```

   Dadurch wird das Schreiben in das aktuelle Protokoll gestoppt und die Protokollierung des Warteschlangenmanagers in den nächsten Protokollspeicherbereich fortgeschrieben. Dadurch wird sichergestellt, dass alle Informationen, die bis zur aktuellen Zeit protokolliert wurden, gesichert werden.
2. Rufen Sie die (neue) aktuelle Nummer des aktiven Protokollspeicherbereichs ab, indem Sie den folgenden Scriptbefehl (MQSC) auf dem zu sichernden WS-Manager absetzen:

   ```
   DIS QMSTATUS CURRLOG
   ```
3. Kopieren Sie die aktualisierten Protokollspeicherbereichsdateien aus dem aktuellen Protokollverzeichnis des Warteschlangenmanagers in das Protokollverzeichnis des Sicherungswarteschlangenmanagers.

   Kopieren Sie alle Protokollspeicherbereiche seit der letzten Aktualisierung und bis (aber nicht einschließlich) den aktuellen Speicherbereich, der in „2” auf Seite 730 angegeben ist. Kopieren Sie nur Protokollspeicherbereichsdateien, die mit "S. .." beginnen.
4. Führen Sie den Steuerbefehl **strmqm** auf dem Sicherungswarteschlangenmanager wie im folgenden Beispiel gezeigt aus:

   ```
   strmqm -r BackupQMName
   ```

   Dadurch werden alle kopierten Protokollextents replays und der Sicherungswarteschlangenmanager wird mit dem Warteschlangenmanager in Schritt gebracht. Wenn die Wiedergabe abgeschlossen ist, empfangen Sie eine Nachricht, die alle für die Neustartwiederherstellung erforderlichen Protokollerweiterungen und alle für die Datenträgerwiederherstellung erforderlichen Protokollspeicherbereiche enthält.

**Zugehörige Verweise**

RESET QMGR

ANZEIGEN QMSTATUS

strmqm (Warteschlangenmanager starten)

## *Sicherungswarteschlangenmanager starten*

Sie können einen Sicherungswarteschlangenmanager für einen nicht wiederherstellbaren Warteschlangenmanager ersetzen.

## Informationen zu diesem Vorgang

Wenn Sie eine Sicherung eines Warteschlangenmanagers in einem Cluster wiederherstellen, finden Sie weitere Informationen unter „Cluster-WS-Manager wiederherstellen” auf Seite 397 und Clustering: Verfügbarkeit, mehrere Instanzen und Disaster-Recovery .

Wenn ein nicht wiederherstellbarer Warteschlangenmanager über einen dedizierten Sicherungswarteschlangenmanager verfügt, können Sie den Sicherungswarteschlangenmanager an Stelle des nicht wiederherstellbaren Warteschlangenmanagers aktivieren.

Wenn ein nicht wiederherstellbarer Warteschlangenmanager durch einen Sicherungswarteschlangenmanager ersetzt wird, können einige der WS-Manager-Daten aus dem nicht wiederherstellbaren Warteschlangenmanager verloren gehen. Die Menge der verloren gegangenen Daten hängt davon ab, wie kürzlich der Sicherungswarteschlangenmanager zuletzt aktualisiert wurde. Je mehr zuletzt die letzte Aktualisierung durchgeführt wurde, der Verlust des Datenverlusts in weniger Warteschlangenmanagern.

**Anmerkung:** Obwohl die WS-Manager-Daten und -Protokolldateien in unterschiedlichen Verzeichnissen gespeichert sind, stellen Sie sicher, dass Sie die Verzeichnisse gleichzeitig sichern und zurückschreiben. Wenn die WS-Manager-Daten und -Protokolldateien unterschiedlich sind, befindet sich der WS-Manager nicht in einem gültigen Status und wird wahrscheinlich nicht gestartet. Selbst wenn es gestartet wird, sind Ihre Daten wahrscheinlich beschädigt.

## Vorgehensweise

1. Führen Sie den Steuerbefehl **strmqm** aus, um den Sicherungswarteschlangenmanager wie im folgenden Beispiel zu aktivieren:

   ```
   strmqm -a BackupQMName
   ```

   Der Sicherungswarteschlangenmanager ist aktiviert. Jetzt, da sie aktiv ist, kann der Sicherungswarteschlangenmanager nicht mehr aktualisiert werden.
2. Führen Sie den Steuerbefehl **strmqm** aus, um den Sicherungswarteschlangenmanager wie im folgenden Beispiel gezeigt zu starten:

   ```
   strmqm BackupQMName
   ```

   IBM MQ betrachtet dies als Neustartwiederherstellung und verwendet das Protokoll aus dem Sicherungswarteschlangenmanager. Während der letzten Aktualisierung für den Sicherungswarteschlangenmanager ist eine Wiedergabe aufgetreten, daher werden nur die aktiven Transaktionen vom zuletzt aufgezeichneten Prüfpunkt rückgängig gemacht.
3. Starten Sie alle Kanäle erneut.
4. Überprüfen Sie die sich ergebende Verzeichnisstruktur, um sicherzustellen, dass alle erforderlichen Verzeichnisse vorhanden sind.

   Weitere Informationen zu IBM MQ-Verzeichnissen und -Unterverzeichnissen finden Sie im Abschnitt Unterstützung für die Dateisystemplanung.
5. Stellen Sie sicher, dass Sie über eine Protokollsteuerdatei sowie über die Protokolldateien verfügen. Überprüfen Sie außerdem, ob die Konfigurationsdateien von IBM MQ und des Warteschlangenmanagers konsistent sind, damit IBM MQ an den richtigen Stellen nach den wiederhergestellten Daten suchen kann.

## Ergebnisse

Wenn die Daten ordnungsgemäß gesichert und wiederhergestellt wurden, wird der Warteschlangenmanager jetzt gestartet.

**Zugehörige Tasks**

„Gestoppte Kanäle erneut starten” auf Seite 251
Wenn ein Kanal in den Status STOPPED wechselt, müssen Sie den Kanal manuell erneut starten.

**Zugehörige Verweise**

strmqm (Warteschlangenmanager starten)

# Änderungen an der Cluster-Fehlerbehebung (auf anderen Servern als z/OS)

Der Warteschlangenmanager führt Operationen, die Probleme verursacht haben, erneut aus, bis die Probleme behoben sind. Wenn die Probleme nach fünf Tagen nicht gelöst sind, schaltet sich der Warteschlangenmanager ab, um zu verhindern, dass der Cache noch mehr veraltet.

Der Warteschlangenmanager führt Operationen, die Probleme verursacht haben, erneut aus, bis die Probleme behoben sind. Wenn die Probleme nach fünf Tagen nicht gelöst sind, schaltet sich der Warteschlangenmanager ab, um zu verhindern, dass der Cache noch mehr veraltet. Je mehr der Cache veraltet, desto mehr Probleme treten auf. Dieses Verhalten in Bezug auf Clusterfehler gilt nicht für z/OS.

Jeder Aspekt der Clusterverwaltung wird für einen Warteschlangenmanager durch den lokalen Repository-Manager-Prozess `amqrrmfa` behandelt. Der Prozess wird auf allen Warteschlangenmanagern ausgeführt, selbst wenn keine Clusterdefinitionen vorhanden sind.

IBM MQ, anstatt den Repository-Manager zu stoppen und ohne ihn weiter zu fahren, führt der Repository-Manager fehlgeschlagene Operationen erneut aus. Wenn der Warteschlangenmanager ein Problem mit dem Repository-Manager feststellt, kann er auf zwei Arten vorgehen.

1. Wenn der Fehler die Operation des Warteschlangenmanagers nicht beeinträchtigt, schreibt dieser eine Nachricht in das Fehlerprotokoll. Die fehlgeschlagene Operation wird alle 10 Minuten ausgeführt, bis sie erfolgreich ist. Standardmäßig muss der Fehler innerhalb von fünf Tagen behoben sein; gelingt dies nicht, schreibt der Warteschlangenmanager eine Nachricht in das Fehlerprotokoll und wird beendet. Die Beendigung nach fünf Tagen kann zurückgestellt werden.
2. Wenn die Operation des Warteschlangenmanagers durch den Fehler beeinträchtigt wird, schreibt dieser eine Nachricht in das Fehlerprotokoll und wird sofort beendet.

Ein Fehler, der die Operation des Warteschlangenmanagers beeinträchtigt, kann vom Warteschlangenmanager nicht diagnostiziert werden oder hat möglicherweise nicht vorhersehbare Folgen. Dieser Fehlertyp führt oft dazu, dass der Warteschlangenmanager eine FFST-Datei schreibt. Fehler, die die Operation des Warteschlangenmanagers beeinträchtigen, können durch einen Fehler in IBM MQ oder durch einen Administrator oder ein Programm verursacht werden, bei dem etwas Unerwartetes ausgeführt wird, wie z. B. das Beenden eines IBM MQ-Prozesses.

Durch die Änderung des Verhaltens bei der Fehlerbehebung soll die Zeit beschränkt werden, während der der Warteschlangenmanager weiterhin mit einer zunehmenden Anzahl inkonsistenter Clusterdefinitionen ausgeführt wird. Da die Anzahl der Inkonsistenzen in Clusterdefinitionen zunimmt, nimmt auch die Wahrscheinlichkeit eines unnormalen Anwendungsverhaltens zu.

Die Standardauswahl, den Warteschlangenmanager nach fünf Tagen zu beenden, ist ein Kompromiss zwischen dem Begrenzen der Anzahl an Inkonsistenzen und dem Verfügbarmachen des Warteschlangenmanagers, bis die Probleme ermittelt und behoben sind.

Sie können die Dauer, bis der Warteschlangenmanager beendet wird, unbegrenzt erweitern und zwischenzeitlich das Problem beheben oder auf das geplante Beenden des Warteschlangenmanagers warten. Die Wartedauer von fünf Tagen ermöglicht es, dass der Warteschlangenmanager über ein langes Wochenende ausgeführt wird und Sie die Möglichkeit haben, auf Probleme zu reagieren oder die Zeitspanne zu verlängern, bevor der Warteschlangenmanager erneut gestartet wird.

## Korrekturmaßnahmen

Sie haben verschiedene Möglichkeiten, um Probleme in Clustern zu beheben. Die erste Möglichkeit besteht darin, das Problem zu überwachen und zu beheben und die zweite Möglichkeit, das Problem zu überwachen und zu verschieben.

1. Überwachen Sie das Fehlerprotokoll des Warteschlangenmanagers auf die Fehlermeldungen AMQ9448 und AMQ5008, und beheben Sie das Problem.

   AMQ9448 zeigt an, dass der Repository-Manager nach der Ausführung eines Befehls einen Fehler zurückgegeben hat. Dieser Fehler markiert den Anfang des Versuchs, den Befehl alle 10 Minuten erneut auszuführen und den Warteschlangenmanager schließlich nach fünf Tagen zu beenden, es sei denn, Sie stellen das Herunterfahren zurück.

AMQ5008 zeigt an, dass der Warteschlangenmanager gestoppt wurde, weil ein IBM MQ-Prozess fehlt. AMQ5008 führt dazu, dass der Repository Manager nach fünf Tagen stoppt. Wenn der Repository-Manager gestoppt wird, wird auch der Warteschlangenmanager gestoppt.

2. Überwachen Sie das Fehlerprotokoll des Warteschlangenmanagers auf die Fehlermeldung AMQ9448, und stellen Sie die Behebung des Problems zurück.

   Wenn Sie den Abruf von Nachrichten aus `SYSTEM.CLUSTER.COMMAND.QUEUE` inaktivieren, stoppt der Repository-Manager den Versuch, Befehle auszuführen, und setzt die Ausführung im Leerlauf, also ohne Arbeiten zu verrichten, unbegrenzt fort. Dabei werden jedoch alle vom Repository-Manager belegten Warteschlangenkennungen freigegeben. Da der Repository-Manager nicht gestoppt wird, wird auch der Warteschlangenmanager nach fünf Tagen nicht gestoppt.

   Führen Sie einen MQSC-Befehl aus, um den Abruf von Nachrichten aus `SYSTEM.CLUSTER.COMMAND.QUEUE` zu inaktivieren:

   ```
   ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
   ```

   Zur Wiederaufnahme des Nachrichtenempfangs aus `SYSTEM.CLUSTER.COMMAND.QUEUE` führen Sie den entsprechenden WebSphere MQ-Scriptbefehl aus:

   ```
   ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
   ```

### Besondere Hinweise

Durch das Stoppen von `amqrrmfa` in IBM MQ wird der Warteschlangenmanager gestoppt, da dies als Warteschlangenmanagerfehler betrachtet wird. Sie dürfen den Prozess `amqrrmfa` nur stoppen, wenn Sie den Optimierungsparameter `TolerateRepositoryFailure`für den Warteschlangenmanager festgelegt haben.

#### Beispiel

```
TuningParameters:
        TolerateRepositoryFailure=TRUE
```

*Abbildung 86. Setzen Sie die Option `TolerateRepositoryFailure` in der Datei 'qm.ini' auf TRUE*

**Zugehörige Konzepte**

„Warteschlangenmanagerkonfigurationsdateien, qm.ini” auf Seite 104
Eine WS-Manager-Konfigurationsdatei, `qm.ini`, enthält Informationen, die für einen bestimmten Warteschlangenmanager relevant sind. Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

## JMS -und Jakarta Messaging -Ressourcen konfigurieren

Eine der Methoden, mit denen eine JMS -oder Jakarta Messaging -Anwendung die Ressourcen erstellen und konfigurieren kann, die sie benötigt, um eine Verbindung zu IBM MQ herzustellen und auf Ziele zum Senden oder Empfangen von Nachrichten zuzugreifen, ist die Verwendung von JNDI ( Java Naming and Directory Interface), um verwaltete Objekte von einer Position innerhalb des Namens-und Verzeichnisservice abzurufen, der als JNDI-Namensbereich bezeichnet wird. Bevor eine JMS-Anwendung verwaltete Objekte aus einem JNDI-Namespace abrufen kann, müssen Sie die verwalteten Objekte zunächst erstellen und konfigurieren.

### Informationen zu diesem Vorgang

JM 3.0 V 9.3.0 V 9.3.0 Ab IBM MQ 9.3.0wird Jakarta Messaging 3.0 für die Entwicklung neuer Anwendungen unterstützt. IBM MQ 9.3.0 unterstützt weiterhin JMS 2.0 für vorhandene Anwendungen. Die Verwendung der Jakarta Messaging 3.0 -API und der JMS 2.0 -API in derselben Anwendung wird nicht unterstützt. Weitere Informationen finden Sie unter Using IBM MQ classes for JMS/Jakarta Messaging.

Sie können verwaltete Objekte mit einem der folgenden Tools in IBM MQ erstellen und konfigurieren:

**IBM MQ JMS -und Jakarta Messaging -Verwaltungstools**
Das Verwaltungstool von IBM MQ JMS , **JMSAdmin**, und das Jakarta Messaging -Verwaltungstool, **JMS30Admin**, sind Befehlszeilentools, mit denen Sie IBM MQ JMS -und Jakarta Messaging -Objekte erstellen und konfigurieren können, die in LDAP, in einem lokalen Dateisystem oder an anderen Positionen gespeichert sind. Die JMS -und Jakarta Messaging -Verwaltungstools verwenden eine Syntax, die **runmqsc**ähnelt, und unterstützen auch das Scripting.

Die Verwaltungstools verwenden eine Konfigurationsdatei, um die Werte bestimmter Eigenschaften festzulegen. Es wird eine Beispielkonfigurationsdatei bereitgestellt, die Sie bearbeiten können, um sie an Ihr System anzupassen, bevor Sie beginnen, mit dem Tool JMS-Ressourcen zu konfigurieren. Weitere Informationen zur Konfigurationsdatei finden Sie in „JMSAdmin-und JMS30Admin -Tools konfigurieren” auf Seite 741.

JMS 2.0 **IBM MQ Explorer**
Für JMS 2.0können Sie IBM MQ Explorer verwenden, um JMS 2.0 -Objektdefinitionen zu erstellen und zu verwalten, die in LDAP, in einem lokalen Dateisystem oder an anderen Positionen gespeichert sind.

JM 3.0 V 9.3.0 V 9.3.0 Für Jakarta Messaging 3.0können Sie JNDI nicht mit IBM MQ Explorerverwalten. Die JNDI-Verwaltung wird von der Variante Jakarta Messaging 3.0 von **JMSAdmin**( **JMS30Admin**) unterstützt.

IBM MQ-JMS-Anwendungen, die in WebSphere Application Server bereitgestellt werden, müssen auf JMS-Objekte aus dem JNDI-Repository des Anwendungsservers zugreifen. Wenn Sie also JMS-Messaging zwischen WebSphere Application Server und IBM MQ verwenden, müssen Sie Objekte in WebSphere Application Server erstellen, die den Objekten entsprechen, die Sie in IBM MQ erstellen.

JM 3.0 V 9.3.0 V 9.3.0 Obwohl IBM MQ 9.3 Jakarta Messaging 3.0unterstützt, hat WebSphere Application Server derzeit keine funktional entsprechende Unterstützung. Daher konfigurieren Sie in WebSphere Application Server Java Message Service 2.0 -Ressourcen.

IBM MQ Explorer und das IBM MQ-JMS-Verwaltungstool können nicht verwendet werden, um IBM MQ-JMS-Objekte zu verwalten, die in WebSphere Application Server gespeichert sind. Stattdessen können Sie mithilfe der folgenden Tools verwaltete Objekte in WebSphere Application Server erstellen und konfigurieren:

**Administrationskonsole von WebSphere Application Server**
Die Administrationskonsole von WebSphere Application Server ist ein webbasiertes Tool, mit dessen Hilfe Sie IBM MQ-JMS-Objekte in WebSphere Application Server verwalten können.

**wsadmin-Scripting-Client von WebSphere Application Server**
Der Scripting-Client 'wsadmin' von WebSphere Application Server enthält spezielle Befehle zum Verwalten von IBM MQ-JMS-Objekten in WebSphere Application Server.

Wenn Sie eine JMS-Anwendung für den Zugriff auf die Ressourcen eines IBM MQ-Warteschlangenmanagers über WebSphere Application Server verwenden möchten, verwenden Sie den IBM MQ-Messaging-Provider in WebSphere Application Server, der eine Version der IBM MQ classes for JMS enthält. Der IBM MQ-Ressourcenadapter, der mit WebSphere Application Server bereitgestellt wird, wird von allen Anwendungen verwendet, die JMS-Messaging mit dem IBM MQ-Messaging-Provider ausführen. Der IBM MQ-Ressourcenadapter wird in der Regel automatisch aktualisiert, wenn Sie WebSphere Application Server-Fixpacks anwenden, wenn Sie den Ressourcenadapter jedoch zuvor manuell aktualisiert haben, müssen Sie Ihre Konfiguration manuell aktualisieren, damit die Wartung ordnungsgemäß angewendet wird.

**Zugehörige Konzepte**
Verbindungsfactorys und Ziele in einer IBM MQ Classes for JMS-Anwendung erstellen und konfigurieren
**Zugehörige Verweise**
runmqsc (MQSC-Befehle ausführen)

# Verbindungsfactorys und Ziele in einem JNDI-Namensbereich konfigurieren

JMS -und Jakarta Messaging -Anwendungen greifen über JNDI ( Java Naming and Directory Interface) auf verwaltete Objekte im Namens-und Verzeichnisservice zu. Die verwalteten Objekte JMS oder Jakarta Messaging werden an einer Position innerhalb des Namens-und Verzeichnisservice gespeichert, der als JNDI-Namensbereich bezeichnet wird. Eine JMS -oder Jakarta Messaging -Anwendung kann die verwalteten Objekte suchen, um eine Verbindung zu IBM MQ herzustellen und auf Ziele zum Senden oder Empfangen von Nachrichten zuzugreifen.

## Informationen zu diesem Vorgang

JMS -oder Jakarta Messaging -Anwendungen suchen die Namen der JMS -oder Jakarta Messaging -Objekte im Namens-und Verzeichnisservice unter Verwendung von Kontexten:

**Ausgangskontext**
Der Ausgangskontext definiert das Stammverzeichnis des JNDI-Namensbereichs. Für jede Position im Namens-und Verzeichnisservice müssen Sie einen Ausgangskontext angeben, um einen Ausgangspunkt anzugeben, von dem aus eine JMS -oder Jakarta Messaging -Anwendung die Namen der verwalteten Objekte an dieser Position des Namens-und Verzeichnisservice auflösen kann.

**Subkontexte**
Ein Kontext kann einen oder mehrere Subkontexte haben. Ein Subkontext ist eine Untergliederung eines JNDI-Namensbereichs und kann verwaltete Objekte wie Verbindungsfactorys und Ziele sowie andere Subkontexte enthalten. Ein Subkontext ist kein eigenes Objekt; er ist lediglich eine Erweiterung der Namenskonvention für die Objekte im Subkontext.

Bevor eine IBM MQ classes for JMS -oder IBM MQ classes for Jakarta Messaging -Anwendung verwaltete Objekte aus einem JNDI-Namensbereich abrufen kann, müssen Sie zuerst die verwalteten Objekte erstellen. Sie können die folgenden Typen von JMS -oder Jakarta Messaging -Objekten erstellen:

**Verbindungsfactory**
Ein Verbindungsfactoryobjekt JMS oder Jakarta Messaging definiert eine Gruppe von Standardkonfigurationseigenschaften für Verbindungen. Eine JMS -oder Jakarta Messaging -Anwendung verwendet eine Verbindungsfactory, um eine Verbindung zu IBM MQherzustellen. Sie können eine Verbindungs-Factory erstellen, die für eine der beiden Messaging-Domänen, die Punkt-zu-Punkt-Messaging-Domäne und die Publish/Subscribe-Messaging-Domäne spezifisch ist.

Alternativ können Sie ab JMS 1.1 domänenunabhängige Verbindungsfactorys erstellen, die sowohl für Punkt-zu-Punkt- als auch für Publish/Subscribe-Messaging verwendet werden können. Weitere Informationen finden Sie unter JMS-und Jakarta-Messaging-Modell.

**Destination**
Ein JMS -oder Jakarta Messaging -Ziel ist ein Objekt, das das Ziel von Nachrichten darstellt, die der Client erzeugt, und die Quelle von Nachrichten, die eine JMS -Anwendung konsumiert. Die Anwendung JMSoder Jakarta Messaging kann entweder ein einzelnes Zielobjekt verwenden, um Nachrichten in einzureihen und daraus abzurufen, oder die Anwendung kann separate Zielobjekte verwenden. Es gibt zwei Typen von Zielobjekten:

- JMS -oder Jakarta Messaging -Warteschlangenziel für Punkt-zu-Punkt-Messaging
- Beim Publish/Subscribe-Messaging verwendetes JMS -oder Jakarta Messaging -Topicziel

JMS 2.0 Für JMS 2.0können Sie Kontexte und verwaltete Objekte mit IBM MQ Explorer oder mit dem IBM MQ JMS -Verwaltungstool **JMSAdmin**erstellen.

**Anmerkung:** JM 3.0 V 9.3.0 V 9.3.0 Für Jakarta Messaging 3.0können Sie JNDI nicht mit IBM MQ Explorerverwalten. Die JNDI-Verwaltung wird von der Variante Jakarta Messaging 3.0 von **JMSAdmin**( **JMS30Admin**) unterstützt.

Das folgende Diagramm zeigt ein Beispiel für JMS -oder Jakarta Messaging -Objekte, die in einem IBM MQ -JNDI-Namensbereich erstellt wurden.

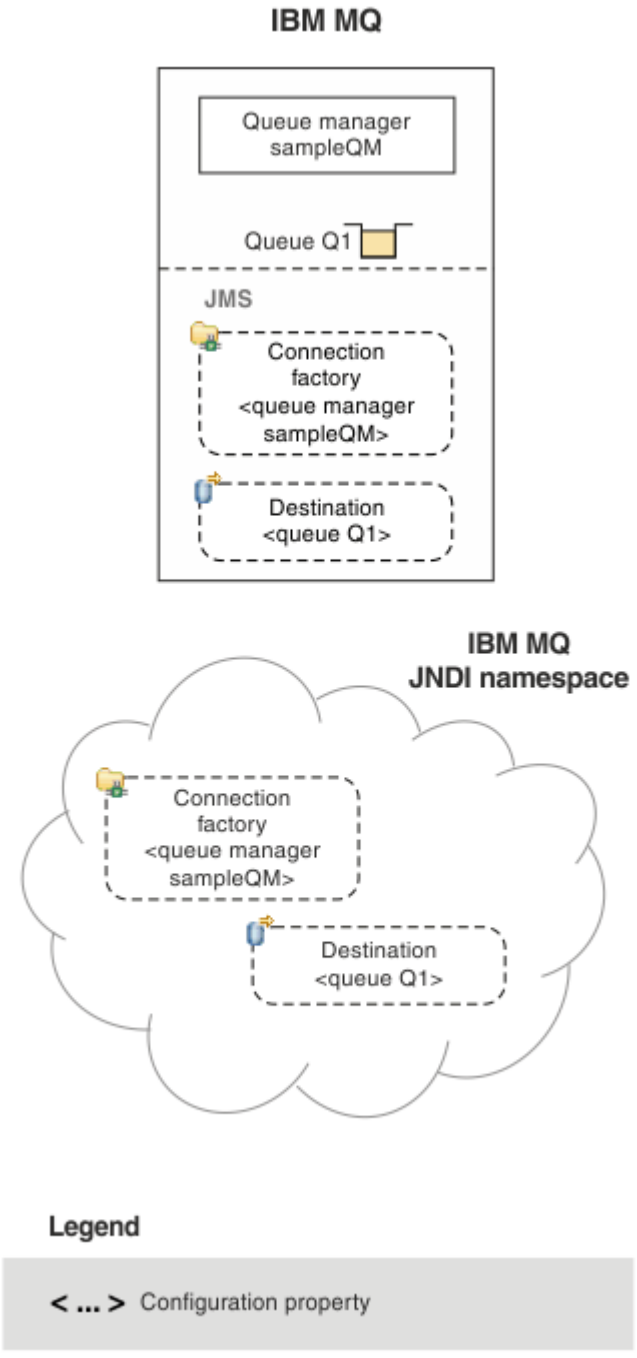



*Abbildung 87. JMS -oder Jakarta Messaging -Objekte, die in IBM MQ erstellt wurden*

Wenn Sie das JMS-Messaging zwischen WebSphere Application Server und IBM MQ verwenden, müssen Sie entsprechende Objekte in WebSphere Application Server erstellen, die für die Kommunikation mit IBM MQ verwendet werden sollen. Wenn Sie eines dieser Objekte in WebSphere Application Server erstellen, wird es im JNDI-Namensbereich von WebSphere Application Server gespeichert, wie im folgenden Diagramm dargestellt.

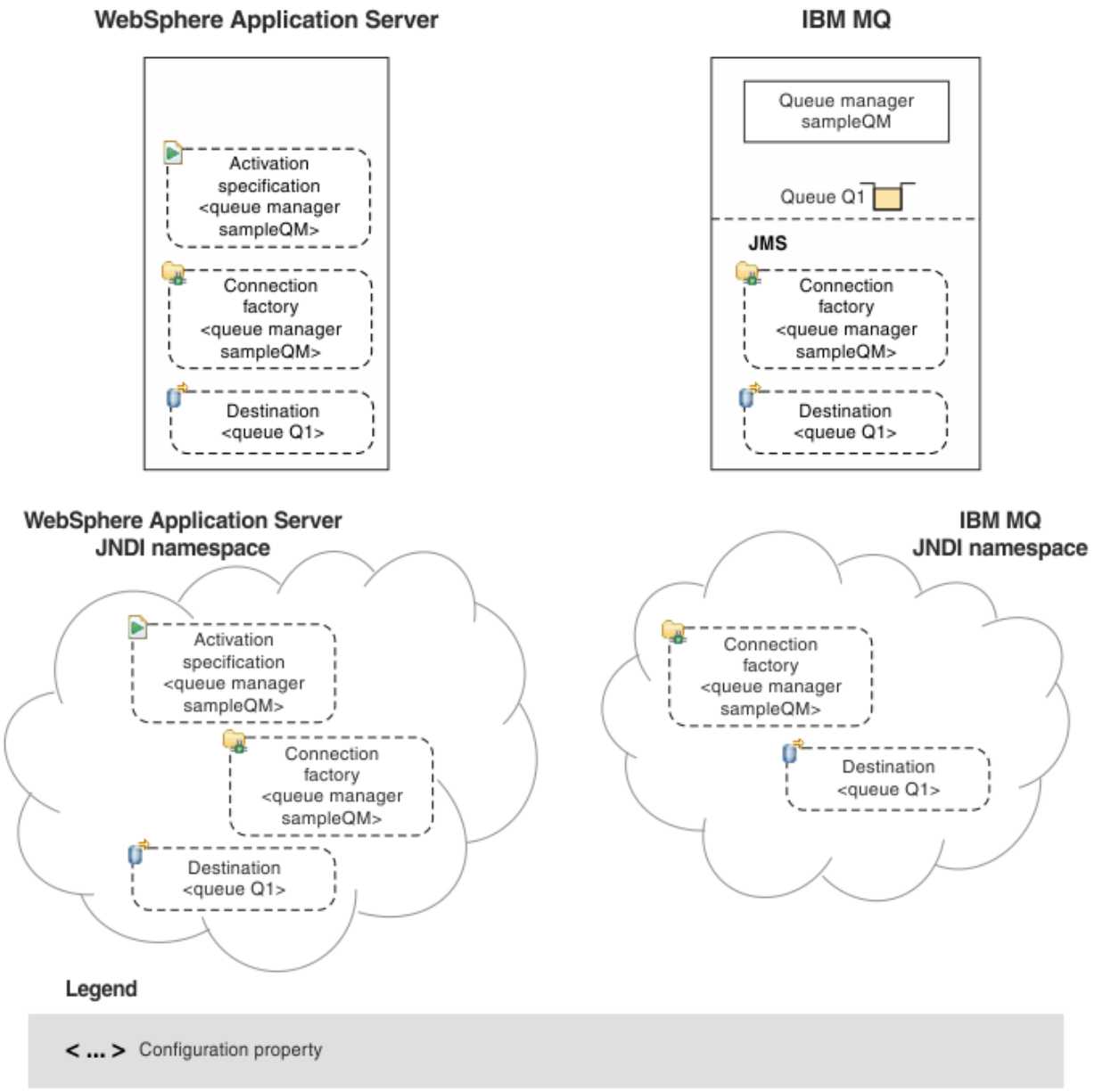



*Abbildung 88. In WebSphere Application Server erstellte Objekte und die entsprechenden Objekte in IBM MQ*

JM 3.0 V 9.3.0 V 9.3.0 Obwohl IBM MQ 9.3 Jakarta Messaging 3.0unterstützt, hat WebSphere Application Server derzeit keine funktional entsprechende Unterstützung. Daher konfigurieren Sie in WebSphere Application Server Java Message Service 2.0 -Ressourcen.

Wenn Ihre Anwendung eine nachrichtengesteuerte Bean (MDB) verwendet, wird die Verbindungs-Factory nur für abgehende Nachrichten verwendet, und eingehende Nachrichten werden von einer Aktivierungsspezifikation empfangen. Aktivierungsspezifikationen sind Teil des Standards Java EE Connector Architecture 1.5 (JCA 1.5). JCA 1.5 bietet eine Standardmöglichkeit für die Integration von JMS-Providern, wie z. B. IBM MQ, mit Java EE-Anwendungsservern, wie z. B. WebSphere Application Server. Eine JMS-Aktivierungsspezifikation kann einer oder mehreren Message-driven Beans (MDBs) zugeordnet sein und stellt die erforderliche Konfiguration bereit, damit diese MDBs an einem Ziel ankommende Nachrichten überwachen.

Sie können die WebSphere Application Server -Administrationskonsole oder wsadmin-Scripting-Befehle verwenden, um die benötigten JMS -Ressourcen zu erstellen und zu konfigurieren.

### Prozedur

- JMS 2.0
  Informationen zur Konfiguration von JMS -Objekten für IBM MQ mithilfe von IBM MQ Explorerfinden Sie in „JMS 2.0-Objekte mit IBM MQ Explorer konfigurieren” auf Seite 738.
- JMS 2.0
  Informationen zum Konfigurieren von JMS -Objekte für IBM MQ mit dem IBM MQ JMS -Verwaltungstool **JMSAdmin**finden Sie unter „JMS -und Jakarta Messaging -Objekte mit den Verwaltungstools konfigurieren” auf Seite 739.
- JM 3.0 V 9.3.0 V 9.3.0
  Informationen zum Konfigurieren von Jakarta Messaging -Objekte für IBM MQ mit dem IBM MQ Jakarta Messaging -Verwaltungstool **JMS30Admin**finden Sie unter „JMS -und Jakarta Messaging -Objekte mit den Verwaltungstools konfigurieren” auf Seite 739.
- JMS 2.0
  Informationen zum Konfigurieren von JMS-Objekten für WebSphere Application Server finden Sie unter „JMS 2.0-Ressourcen in WebSphere Application Server konfigurieren” auf Seite 750.

### Ergebnisse

Eine IBM MQ classes for JMS -oder IBM MQ classes for Jakarta Messaging -Anwendung kann die verwalteten Objekte aus dem JNDI-Namensbereich abrufen und bei Bedarf eine oder mehrere ihrer Eigenschaften über die IBM JMS -Erweiterungen oder die IBM MQ JMS -Erweiterungen festlegen oder ändern.

**Zugehörige Tasks**

JNDI für das Abrufen von verwalteten Objekten in einer JMS-Anwendung verwenden

Verbindungsfactorys und Ziele in einer Anwendung der IBM MQ classes for JMS erstellen und konfigurieren

## JMS 2.0 JMS 2.0-Objekte mit IBM MQ Explorer konfigurieren

Über die grafische Benutzerschnittstelle von IBM MQ Explorer können Sie JMS-Objekte aus IBM MQ-Objekten sowie IBM MQ-Objekte aus JMS-Objekten erstellen. Darüber hinaus können Sie sonstige IBM MQ-Objekte verwalten und überwachen.

### Informationen zu diesem Vorgang

JMS 2.0 IBM MQ Explorer ist die grafische Benutzerschnittstelle, in der Sie IBM MQ-Objekte verwalten und überwachen können, unabhängig davon, ob sie auf Ihrem lokalen oder auf einem fernen System gespeichert sind. IBM MQ Explorer wird unter Windows und Linux for x86-64ausgeführt. Er kann über Fernzugriff eine Verbindung zu Warteschlangenmanagern herstellen, die auf einer unterstützten Plattform einschließlich z/OS ausgeführt werden, wodurch Ihr gesamter Messaging-Backbone über die Konsole angezeigt, durchsucht und geändert werden kann.

**Anmerkung:** JM 3.0 V 9.3.0 V 9.3.0 Für Jakarta Messaging 3.0können Sie JNDI nicht mit IBM MQ Explorerverwalten. Die JNDI-Verwaltung wird von der Variante Jakarta Messaging 3.0 von **JMSAdmin**( **JMS30Admin**) unterstützt.

In IBM MQ Explorer werden alle Verbindungsfactorys in Ordnern für Verbindungsfactorys in dem entsprechenden Kontext und in den Subkontexten gespeichert.

Sie können die folgenden Arten von Tasks mit IBM MQ Explorer ausführen, im Kontext eines vorhandenen Objekts im IBM MQ Explorer oder in einem Assistenten zum Erstellen neuer Objekte:

- Erstellen einer JMS-Verbindungsfactory aus einem der folgenden IBM MQ-Objekte:
  - IBM MQ-Warteschlangenmanager auf Ihrem lokalen Computer oder auf einem fernen System
  - IBM MQ-Kanal

  - IBM MQ-Listener
- Hinzufügen eines IBM MQ-Warteschlangenmanagers zu IBM MQ Explorer unter Verwendung einer JMS-Verbindungsfactory
- Erstellen einer JMS-Warteschlange aus einer IBM MQ-Warteschlange
- Erstellen einer IBM MQ-Warteschlange aus einer JMS-Warteschlange
- Erstellen eines JMS-Topics aus einem IBM MQ-Topic, bei dem es sich um ein IBM MQ-Objekt oder ein dynamisches Topic handeln kann
- Erstellen eines IBM MQ-Topics aus einem JMS-Topic

### Prozedur

- Starten Sie IBM MQ Explorer, wenn er nicht bereits aktiv ist.

  Wenn IBM MQ Explorer ausgeführt wird und die Begrüßungsseite angezeigt wird, schließen Sie die Begrüßungsseite, um mit der Verwaltung der IBM MQ-Objekte zu beginnen.
- Wenn Sie dies noch nicht getan haben, erstellen Sie einen Ausgangskontext, der das Stammverzeichnis des JNDI-Namensbereichs definiert, in dem die JMS-Objekte im Namens- und Verzeichnisservice gespeichert werden.

  Wenn Sie den Ausgangskontext in IBM MQ Explorer hinzugefügt haben, können Sie Verbindungsfactory- und Zielobjekte sowie Subkontexte im JNDI-Namensbereich erstellen.

  Der Ausgangskontext wird in der Navigatoransicht im JMS-Ordner "Administered Objects" angezeigt. Beachten Sie, dass zwar der vollständige Inhalt des JNDI-Namensbereichs angezeigt wird, in IBM MQ Explorer jedoch nur die IBM MQ classes for JMS-Objekte bearbeitet werden können, die dort gespeichert sind. Weitere Informationen hierzu finden Sie im Abschnitt Ausgangskontext hinzufügen .
- Erstellen und konfigurieren Sie die Subkontexte und von JMS verwalteten Objekte, die Sie benötigen.

  Weitere Informationen finden Sie im Abschnitt Von JMS verwaltete Objekte erstellen und konfigurieren.
- Konfigurieren Sie IBM MQ.

  Weitere Informationen finden Sie unter IBM MQ mit IBM MQ Explorer konfigurieren.

**Zugehörige Konzepte**

Einführung in IBM MQ Explorer

Verbindungsfactorys und Ziele in einer Anwendung der IBM MQ classes for JMS erstellen und konfigurieren

## JMS -und Jakarta Messaging -Objekte mit den Verwaltungstools konfigurieren

IBM MQ stellt Verwaltungstools bereit, mit denen Sie die Eigenschaften von acht Typen von IBM MQ classes for JMS -oder IBM MQ classes for Jakarta Messaging -Objekten definieren und in einem JNDI-Namensbereich speichern können. Anwendungen können dann JNDI verwenden, um diese verwalteten Objekte aus dem Namespace abzurufen.

### Informationen zu diesem Vorgang

JMS 2.0 Für JMS 2.0wird die JNDI-Verwaltung vom Tool **JMSAdmin** unterstützt.

JM 3.0 V 9.3.0 V 9.3.0 Für Jakarta Messaging 3.0wird die JNDI-Verwaltung von der Variante Jakarta Messaging 3.0 von **JMSAdmin**( **JMS30Admin**) unterstützt.

In der folgenden Tabelle werden die acht Typen von verwalteten Objekten angezeigt, die Sie mit Verben erstellen, konfigurieren und bearbeiten können. Die Schlüsselwortspalte enthält die Zeichenfolgen, die Sie für *TYPE* in den in Tabelle 36 auf Seite 740 gezeigten Befehlen ersetzen können.

| *Tabelle 36. Die Objekttypen JMS und Jakarta Messaging , die vom Verwaltungstool verarbeitet werden* | | |
|---|---|---|
| **Objekttyp** | **Schlüsselwort** | **Beschreibung** |
| MQConnectionFactory | CF | Die IBM MQ-Implementierung der ConnectionFactory-Schnittstelle von JMS. Dies stellt ein Factory-Objekt zum Erstellen von Verbindungen in den Punkt-zu-Punkt-und Publish/Subscribe-Domänen dar. |
| MQQueueConnectionFactory | QCF | Die IBM MQ-Implementierung der Schnittstelle 'QueueConnectionFactory' von JMS QueueConnectionFactory. Dies stellt ein Factory-Objekt zum Erstellen von Verbindungen in der Punkt-zu-Punkt-Domäne dar. |
| MQTopicConnectionFactory | TCF | Die IBM MQ-Implementierung der Schnittstelle 'JMS TopicConnectionFactory'. Dies stellt ein Factory-Objekt zum Erstellen von Verbindungen in der Publish/Subscribe-Domäne dar. |
| MQQueue | Q | Die IBM MQ-Implementierung der Warteschlangenschnittstelle von JMS. Dies ist ein Ziel für Nachrichten in der Punkt-zu-Punkt-Domäne. |
| MQTopic | T | Die IBM MQ-Implementierung der JMS-Themenschnittstelle. Dies stellt ein Ziel für Nachrichten in der Publish/Subscribe-Domäne dar. |
| MQXAConnectionFactory „1" auf Seite 740 | XACF | Die IBM MQ-Implementierung der Schnittstelle 'JMS XAConnectionFactory'. Diese stellt ein Factory-Objekt zum Erstellen von Verbindungen in den Punkt-zu-Punkt- und Publish/Subscribe-Domänen dar und gibt an, wo die Verbindungen die XA-Versionen von JMS-Klassen verwenden. |
| MQXAQueueConnectionFactory „1" auf Seite 740 | XAQCF | Die IBM MQ-Implementierung der Schnittstelle 'JMS XAQueueConnectionFactory'. Diese stellt ein Factory-Objekt zum Erstellen von Verbindungen in der Punkt-zu-Punkt-Domäne dar, die die XA-Versionen von JMS-Klassen verwenden. |
| MQXATopicConnectionFactory „1" auf Seite 740 | XATCF | Die IBM MQ-Implementierung der Schnittstelle 'JMS XATopicConnectionFactory'. Diese stellt ein Factory-Objekt zum Erstellen von Verbindungen in der Publish/Subscribe-Domäne dar, die die XA-Versionen von JMS-Klassen verwenden. |

**Anmerkung:**

1. Diese Klassen werden für die Verwendung durch Anbieter von Anwendungsservern bereitgestellt. Es ist unwahrscheinlich, dass sie für Anwendungsprogrammierer direkt von Nutzen sein können.

Weitere Informationen zum Konfigurieren dieser Objekte finden Sie in „JMS-Objekte konfigurieren" auf Seite 749.

Die Eigenschaftstypen und Werte, die Sie für die Verwendung dieses Tools benötigen, sind im Abschnitt Eigenschaften von Objekten der IBM MQ classes for JMS aufgeführt.

Sie können das Tool auch verwenden, um Verzeichnisnamensbereichssubkontexte innerhalb der JNDI zu bearbeiten, wie in „Subkontexte konfigurieren” auf Seite 746 beschrieben.

JMS 2.0 Für JMS 2.0 und frühere Versionen können Sie auch verwaltete IBM MQ classes for JMS -Objekte mit IBM MQ Explorererstellen und konfigurieren.

JM 3.0 V 9.3.0 V 9.3.0 Für Jakarta Messaging 3.0können Sie JNDI nicht mit IBM MQ Explorerverwalten. Die JNDI-Verwaltung wird von der Variante Jakarta Messaging 3.0 von **JMSAdmin**( **JMS30Admin**) unterstützt.

**Zugehörige Konzepte**

Verbindungsfactorys und Ziele in einer Anwendung der IBM MQ classes for JMS erstellen und konfigurieren

JNDI für das Abrufen von verwalteten Objekten in einer JMS-Anwendung verwenden

## JMSAdmin-und JMS30Admin -Tools konfigurieren

Die IBM MQ JMS -und Jakarta Messaging -Verwaltungstools verwenden eine Konfigurationsdatei, um die Werte bestimmter Eigenschaften festzulegen. In jedem Fall wird eine Beispielkonfigurationsdatei bereitgestellt, die Sie an Ihr System anpassen können.

### Informationen zu diesem Vorgang

JM 3.0 V 9.3.0 V 9.3.0 IBM MQ 9.3.0 bietet jetzt Unterstützung für Jakarta Messaging 3.0. JMS 2.0 wird weiterhin vollständig unterstützt.

Die Konfigurationsdatei ist eine Textdatei, die aus einer Gruppe von Schlüssel/Wert-Paaren besteht, die durch das Gleichheitszeichen (=) getrennt sind. Sie konfigurieren das Verwaltungstool, indem Sie Werte für die drei Eigenschaften festlegen, die in der Konfigurationsdatei definiert sind. Das folgende Beispiel zeigt diese drei Eigenschaften:

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

In diesem Beispiel gibt ein Hashzeichen (#) in der ersten Spalte der Zeile einen Kommentar oder eine Zeile an, die nicht verwendet wird.

Eine Beispielkonfigurationsdatei, die als Standardkonfigurationsdatei dient, wird mit IBM MQ bereitgestellt. Die Beispieldatei heißt `JMSAdmin.config` (für JMS 2.0) oder `JMS30Admin.config` (für Jakarta Messaging 3.0). Diese Datei befindet sich im Verzeichnis *MQ_JAVA_INSTALL_PATH*`/bin` . Sie können entweder die Beispieldatei bearbeiten, um die für Ihr System erforderlichen Einstellungen zu definieren, oder eine eigene Konfigurationsdatei erstellen.

Wenn Sie das Verwaltungstool starten, können Sie die Konfigurationsdatei angeben, die Sie verwenden möchten, indem Sie den Befehlszeilenparameter `-cfg` verwenden, wie in „Die Tools JMSAdmin und JMS30Admin starten” auf Seite 743beschrieben. Wenn Sie beim Aufrufen des Tools keinen Konfigurationsdateinamen angeben, versucht das Tool, die Standardkonfigurationsdatei ( `JMSAdmin.config` oder `JMS30Admin.config`) zu laden. Diese Datei wird zuerst im aktuellen Verzeichnis und dann im Verzeichnis *MQ_JAVA_INSTALL_PATH*`/bin` gesucht, wobei *MQ_JAVA_INSTALL_PATH* der Pfad zu Ihrer IBM MQ classes for JMS -bzw. IBM MQ classes for Jakarta Messaging -Installation ist.

Die Namen von JMS -oder Jakarta Messaging -Objekten, die in einer LDAP-Umgebung gespeichert sind, müssen den LDAP-Namenskonventionen entsprechen. Eine dieser Konventionen besteht darin, dass Objekt-und Kontextnamen ein Präfix enthalten müssen, wie z. B. `cn=` (allgemeiner Name) oder `ou=` (Organi-

sationseinheit). Das Verwaltungstool vereinfacht die Verwendung von LDAP-Service-Providern, indem es Ihnen erlaubt, auf Objekt-und Kontextnamen ohne Präfix zu verweisen. Wenn Sie kein Präfix angeben, fügt das Tool dem Namen, den Sie angeben, automatisch ein Standardpräfix hinzu. Für LDAP ist dies cn=. Falls erforderlich, können Sie das Standardpräfix ändern, indem Sie die Eigenschaft **NAME_PREFIX** in der Konfigurationsdatei festlegen.

**Anmerkung:** Möglicherweise müssen Sie Ihren LDAP-Server für die Speicherung von Java-Objekten konfigurieren. Weitere Informationen finden Sie in der Dokumentation zu Ihrem LDAP-Server.

## Vorgehensweise

1. Definieren Sie den Serviceprovider, den das Tool verwendet, indem Sie die Eigenschaft **INITIAL_CONTEXT_FACTORY** konfigurieren.

   Die unterstützten Werte für diese Eigenschaft lauten wie folgt:

   - com.sun.jndi.ldap.LdapCtxFactory (für LDAP)
   - com.sun.jndi.fscontext.RefFSContextFactory (für Dateisystemkontext)
   - z/OS com.ibm.jndi.LDAPCtxFactory wird nur unter z/OS unterstützt und bietet Zugriff auf einen LDAP-Server. Diese Klasse ist jedoch nicht kompatibel mit com.sun.jndi.ldap.LdapCtxFactory, indem Objekte, die mit einer InitialContextFactory erstellt wurden, nicht mit der anderen gelesen oder geändert werden können.

   Sie können das Verwaltungstool auch verwenden, um eine Verbindung zu anderen JNDI-Kontexten herzustellen, indem Sie drei Parameter verwenden, die in der JMSAdmin-oder JMS30Admin -Konfigurationsdatei definiert sind. Gehen Sie wie folgt vor, um eine andere InitialContextFactory zu verwenden

   a) Setzen Sie die Eigenschaft **INITIAL_CONTEXT_FACTORY** auf den erforderlichen Klassennamen.

   b) Definieren Sie das Verhalten von InitialContextFactory mit den Eigenschaften **USE_INITIAL_DIR_CONTEXT**, **NAME_PREFIX** und **NAME_READABILITY_MARKER** .

      Die Einstellungen für diese Eigenschaften werden in den Kommentaren der Beispielkonfigurationsdatei beschrieben.

   Sie müssen die Eigenschaften **USE_INITIAL_DIR_CONTEXT**, **NAME_PREFIX** und **NAME_READABILITY_MARKER** nicht definieren, wenn Sie einen der unterstützten **INITIAL_CONTEXT_FACTORY** -Werte verwenden. Sie können jedoch Werte für diese Eigenschaften angeben, wenn Sie die Systemstandardwerte außer Kraft setzen wollen. Wenn Ihre Objekte z. B. in einer LDAP-Umgebung gespeichert sind, können Sie das Standardpräfix ändern, das das Tool zu Objekt-und Kontextnamen hinzufügt, indem Sie die Eigenschaft **NAME_PREFIX** auf das erforderliche Präfix setzen.

   Wenn Sie eine oder mehrere der drei Eigenschaften von InitialContextFactory weglassen, stellt das Verwaltungstool geeignete Standardwerte bereit, die auf den Werten der anderen Eigenschaften basieren.

2. Definieren Sie die URL des Ausgangskontexts der Sitzung, indem Sie die Eigenschaft **PROVIDER_URL** konfigurieren.

   Diese URL ist das Stammverzeichnis aller JNDI-Operationen, die vom Tool ausgeführt werden. Es werden zwei Formen dieser Eigenschaft unterstützt:

   - ldap://hostname/contextname
   - file: [ Laufwerk:] /Pfadname

   Das Format der LDAP-URL kann abhängig von Ihrem LDAP-Provider variieren. Weitere Informationen finden Sie in der LDAP-Dokumentation.

3. Definieren Sie, ob JNDI Sicherheitsberechtigungsnachweise an Ihren Service-Provider übergibt, indem Sie die Eigenschaft **SECURITY_AUTHENTICATION** konfigurieren.

   Diese Eigenschaft wird nur verwendet, wenn ein LDAP-Service-Provider verwendet wird, und kann einen der folgenden drei Werte annehmen:

**none (anonyme Authentifizierung)**
Wenn Sie diesen Parameter auf `none` setzen, gibt JNDI keine Sicherheitsberechtigungsnachweise an den Service-Provider aus, und die *anonyme Authentifizierung* wird ausgeführt.

**`simple` (einfache Authentifizierung)**
Wenn Sie den Parameter auf `simple` setzen, werden die Sicherheitsberechtigungsnachweise über JNDI an den zugrunde liegenden Service-Provider übergeben. Diese Sicherheitsberechtigungsnachweise sind in Form eines definierten Benutzernamens (Benutzer-DN) und eines Kennworts vorhanden.

**`CRAM-MD5` (Authentifizierungsverfahren CRAM-MD5)**
Wenn Sie den Parameter auf `CRAM-MD5` setzen, werden Sicherheitsberechtigungsnachweise über JNDI an den zugrunde liegenden Service-Provider übergeben. Diese Sicherheitsberechtigungsnachweise sind in Form eines definierten Benutzernamens (Benutzer-DN) und eines Kennworts vorhanden.

Wenn Sie keinen gültigen Wert für die Eigenschaft **SECURITY_AUTHENTICATION** angeben, wird die Eigenschaft standardmäßig auf `none` gesetzt.

Wenn Sicherheitsberechtigungsnachweise erforderlich sind, werden Sie beim Initialisieren des Tools zur Eingabe aufgefordert. Sie können dies vermeiden, indem Sie die Eigenschaften **PROVIDER_USERDN** und **PROVIDER_PASSWORD** in der JMSAdmin-Konfigurationsdatei festlegen.

**Anmerkung:** Wenn Sie diese Eigenschaften nicht verwenden, wird der eingegebene Text *einschließlich des Kennworts* an die Anzeige zurückgemeldet. Dies kann Sicherheitsauswirkungen haben.

Das Tool authentifiziert sich nicht selbst. Die Authentifizierungstask wird an den LDAP-Server delegiert. Der LDAP-Serveradministrator muss Zugriffsberechtigungen für verschiedene Teile des Verzeichnisses einrichten und verwalten. Weitere Informationen finden Sie in der LDAP-Dokumentation. Wenn die Authentifizierung fehlschlägt, zeigt das Tool eine entsprechende Fehlernachricht an und wird beendet.

Ausführlichere Informationen zu Sicherheit und JNDI finden Sie in der Dokumentation auf der Oracle-Website Java ( Oracle Technology Network for Java Developers ).

## Die Tools JMSAdmin und JMS30Admin starten

Die IBM MQ JMS -und Jakarta Messaging -Verwaltungstools verfügen über eine Befehlszeilenschnittstelle, die Sie entweder interaktiv oder zum Starten eines Batchprozesses verwenden können.

### Informationen zu diesem Vorgang

Der interaktive Modus stellt eine Eingabeaufforderung zur Verfügung, in der Sie Verwaltungsbefehle eingeben können. Im Stapelbetrieb enthält der Befehl zum Starten des Tools den Namen einer Datei, die ein Verwaltungsbefehlsscript enthält.

### Prozedur

Interaktiver Modus

- Geben Sie den folgenden Befehl ein, um das Tool im interaktiven Modus zu starten:

  JMS 2.0

  ```
  JMSAdmin [-t] [-v] [-cfg config_filename]
  ```

  JM 3.0

  ```
  JMS30Admin [-t] [-v] [-cfg config_filename]
  ```

  Dabei gilt:

**-t**
Aktiviert den Trace (Standardeinstellung ist trace off).

Die Tracedatei wird in `"%MQ_JAVA_DATA_PATH%"\errors` ( Windows ) oder `/var/mqm/trace` ( AIX and Linux ) generiert. Der Name der Tracedatei hat das folgende Format:

`mqjms_`*`PID`*`.trc`

Dabei steht *`PID`* für die Prozess-ID der JVM.

**-v**
Erzeugt eine ausführliche Ausgabe (Standardeinstellung ist eine kurze Ausgabe).

**-cfg Konfigurationsdateiname**
Benennt eine alternative Konfigurationsdatei. Wenn dieser Parameter nicht angegeben wird, wird die Standardkonfigurationsdatei `JMSAdmin.config` (für JMS 2.0) oder `JMS30Admin.config` (für Jakarta Messaging 3.0) verwendet. Weitere Informationen zur Konfigurationsdatei finden Sie in „JMSAdmin-und JMS30Admin -Tools konfigurieren” auf Seite 741.

Es wird eine Eingabeaufforderung angezeigt, die angibt, dass das Tool bereit ist, Verwaltungsbefehle zu akzeptieren. Diese Eingabeaufforderung wird zunächst wie folgt angezeigt:

```
InitCtx>
```

Dies weist darauf hin, dass der aktuelle Kontext (d. a. der JNDI-Kontext, auf den alle Benennungs-und Verzeichnisoperationen derzeit verweisen) der Ausgangskontext ist, der im Konfigurationsparameter **PROVIDER_URL** definiert ist. Weitere Informationen zu diesem Parameter finden Sie unter „JMSAdmin-und JMS30Admin -Tools konfigurieren” auf Seite 741.

Wenn Sie den Verzeichnisnamensbereich durchlaufen, ändert sich die Eingabeaufforderung, sodass die Eingabeaufforderung immer den aktuellen Kontext anzeigt.

Stapelbetrieb

- Geben Sie den folgenden Befehl ein, um das Tool im Stapelbetrieb zu starten:

  JMS 2.0

  ```
  JMSAdmin test.scp
  ```

  JM 3.0

  ```
  JMS30Admin test.scp
  ```

  Dabei steht *`test.scp`* für eine Scriptdatei, die Verwaltungsbefehle enthält. Weitere Informationen finden Sie unter „Verwaltungsbefehle mit JMSAdmin und JMS30Admin verwenden” auf Seite 744. Der letzte Befehl in der Datei muss der Befehl `END` sein.

## Verwaltungsbefehle mit JMSAdmin und JMS30Admin verwenden

Die IBM MQ JMS -und Jakarta Messaging -Verwaltungstools akzeptieren Befehle, die aus einem Verwaltungsverb und den entsprechenden Parametern bestehen.

### Informationen zu diesem Vorgang

In der folgenden Tabelle sind die Verwaltungsverben aufgelistet, die Sie beim Eingeben von Befehlen mit den Verwaltungstools verwenden können.

| *Tabelle 37. Verwaltungsverben* | | |
|---|---|---|
| **Verb** | **Kurzform** | **Beschreibung** |
| ALTER | ALT | Ändern Sie mindestens eine der Eigenschaften eines verwalteten Objekts. |
| DEFINIER | DEF | Ein verwalttes Objekt erstellen und speichern oder einen Subkontext erstellen |
| ANZEIGEN | DIS | Zeigt die Eigenschaften eines oder mehrerer gespeicherter verwalt-ter Objekte oder den Inhalt des aktuellen Kontexts an. |
| LÖSCHEN | DEL | Entfernt ein oder mehrere verwaltete Objekte aus dem Namensbe-reich oder entfernt einen leeren Subkontext. |
| ÄNDERUNG | CHG | Ändern Sie den aktuellen Kontext, sodass der Benutzer den Ver-zeichnisnamensbereich an einer beliebigen Stelle unterhalb des Ausgangskontextes durchqueren kann (anstehende Sicherheitsfrei-gabe) |
| KOPIEREN | CP | Erstellen Sie eine Kopie eines gespeicherten verwalteten Objekts und speichern Sie es unter einem alternativen Namen. |
| MOVE | MV | Ändern des Namens, unter dem ein verwalttes Objekt gespeichert wird |
| ENDE | | Verwaltungstool schließen |

## Prozedur

- Wenn das Verwaltungstool noch nicht gestartet wurde, starten Sie es wie im Abschnitt „Die Tools JMSAdmin und JMS30Admin starten” auf Seite 743 beschrieben.

  Die Eingabeaufforderung wird angezeigt und zeigt an, dass das Tool bereit ist, Verwaltungsbefehle zu akzeptieren. Diese Eingabeaufforderung wird zunächst wie folgt angezeigt:

  ```
  InitCtx>
  ```

  Um den aktuellen Kontext zu ändern, verwenden Sie das Verb CHANGE wie in „Subkontexte konfigurieren” auf Seite 746 beschrieben.
- Geben Sie Befehle in das folgende Format ein:

  ```
  verb [param]*
  ```

  Dabei steht **verb** für eines der in Tabelle 37 auf Seite 745aufgelisteten Verwaltungsverben. Alle gültigen Befehle enthalten ein Verb, das am Anfang des Befehls in der Standardform oder in der Kurzform angezeigt wird. Bei Verb-Namen wird die Groß-/Kleinschreibung nicht beachtet.
- Um einen Befehl zu beenden, drücken Sie die Eingabetaste, es sei denn, Sie möchten mehrere Befehle zusammen eingeben. In diesem Fall geben Sie das Pluszeichen (+) direkt ein, bevor Sie die Eingabe-taste drücken.

  In der Regel drücken Sie die Eingabetaste, um die Befehle zu beenden. Sie können dies jedoch überschreiben, indem Sie das Pluszeichen (+) direkt eingeben, bevor Sie die Eingabetaste drücken. Auf diese Weise können Sie mehrzeilweise Befehle eingeben, wie im folgenden Beispiel gezeigt:

  ```
  DEFINE Q(BookingsInputQueue) +
  QMGR(QM.POLARIS.TEST) +
  QUEUE(BOOKINGS.INPUT.QUEUE) +
  PORT(1415) +
  CCSID(437)
  ```

- Verwenden Sie zum Schließen des Verwaltungstools das Verb **END** .
  Dieses Verb kann keine Parameter verwenden.

## Subkontexte konfigurieren

Sie können die Verben **CHANGE**, **DEFINE**, **DISPLAY** und **DELETE** verwenden, um Subkontexte für Verzeichnisnamensbereiche zu konfigurieren.

### Informationen zu diesem Vorgang

Die Verwendung dieser Verben ist in der folgenden Tabelle beschrieben.

*Tabelle 38. Syntax und Beschreibung der Befehle, die zum Bearbeiten von Subkontexten verwendet werden*

| **Befehlssyntax** | **Beschreibung** |
|---|---|
| DEFINE CTX (ctxName) | Es wird versucht, einen untergeordneten Subkontext des aktuellen Kontexts zu erstellen, der den Namen ctxName hat. Gibt an, ob ein Sicherheitsverstoß vorliegt, wenn der Subkontext bereits vorhanden ist oder wenn der angegebene Name nicht gültig ist. |
| ANZEIGEN CTX | Zeigt den Inhalt des aktuellen Kontexts an. Administrierte Objekte werden mit a, Subkontexten mit `[D]` annotiert. Der Java-Typ jedes Objekts wird ebenfalls angezeigt. |
| DELETE CTX (ctxName) | Es wird versucht, den untergeordneten Kontext des aktuellen Kontextes mit dem Namen 'ctxName' zu löschen. Falls der Kontext nicht gefunden wird, ist er nicht leer, oder wenn ein Sicherheitsverstoß vorliegt. |
| CTX ÄNDERN (ctxName) | Ändert den aktuellen Kontext, so dass er sich jetzt auf den untergeordneten Kontext mit dem Namen ctxName bezieht. Es kann einer von zwei speziellen Werten von ctxName angegeben werden:<br>**= AKTIV**<br>wird in den übergeordneten Kontext des aktuellen Kontexts verschoben<br>**= INIT**<br>wird direkt in den Ausgangskontext verschoben<br>Es ist fehlgeschlagen, wenn der angegebene Kontext nicht vorhanden ist oder wenn ein Sicherheitsverstoß vorliegt. |

Die Namen von JMS -oder Jakarta Messaging -Objekten, die in einer LDAP-Umgebung gespeichert sind, müssen den LDAP-Namenskonventionen entsprechen. Eine dieser Konventionen besteht darin, dass Objekt-und Kontextnamen ein Präfix enthalten müssen, wie z. B. cn= (allgemeiner Name) oder ou= (Organisationseinheit). Das Verwaltungstool vereinfacht die Verwendung von LDAP-Service-Providern, indem es Ihnen erlaubt, auf Objekt-und Kontextnamen ohne Präfix zu verweisen. Wenn Sie kein Präfix angeben, fügt das Tool dem Namen, den Sie angeben, automatisch ein Standardpräfix hinzu. Für LDAP ist dies cn=. Falls erforderlich, können Sie das Standardpräfix ändern, indem Sie die Eigenschaft **NAME_PREFIX** in der Konfigurationsdatei festlegen. Weitere Informationen finden Sie unter „JMSAdmin-und JMS30Admin -Tools konfigurieren” auf Seite 741.

**Anmerkung:** Möglicherweise müssen Sie Ihren LDAP-Server für die Speicherung von Java-Objekten konfigurieren. Weitere Informationen finden Sie in der Dokumentation zu Ihrem LDAP-Server.

## JMS-Objekte erstellen

Verwenden Sie das Verb DEFINE , um Verbindungsfactory-und Zielobjekte für JMS oder Jakarta Messaging zu erstellen und in einem JNDI-Namensbereich zu speichern. Um Ihre Objekte in einer LDAP-Um-

gebung zu speichern, müssen Sie diese Namen angeben, die bestimmten Konventionen entsprechen. Das Verwaltungstool kann Ihnen bei der Einhaltung der LDAP-Namenskonventionen helfen, indem Sie Objektnamen ein Standardpräfix hinzufügen.

## Informationen zu diesem Vorgang

Das Verb DEFINE erstellt ein verwalttes Objekt mit dem Typ, dem Namen und den Eigenschaften, die Sie angeben. Das neue Objekt wird im aktuellen Kontext gespeichert.

Die Namen von JMS -oder Jakarta Messaging -Objekten, die in einer LDAP-Umgebung gespeichert sind, müssen den LDAP-Namenskonventionen entsprechen. Eine dieser Konventionen besteht darin, dass Objekt-und Kontextnamen ein Präfix enthalten müssen, wie z. B. cn= (allgemeiner Name) oder ou= (Organisationseinheit). Das Verwaltungstool vereinfacht die Verwendung von LDAP-Service-Providern, indem es Ihnen erlaubt, auf Objekt-und Kontextnamen ohne Präfix zu verweisen. Wenn Sie kein Präfix angeben, fügt das Tool dem Namen, den Sie angeben, automatisch ein Standardpräfix hinzu. Für LDAP ist dies cn=. Falls erforderlich, können Sie das Standardpräfix ändern, indem Sie die Eigenschaft **NAME_PREFIX** in der Konfigurationsdatei festlegen. Weitere Informationen finden Sie unter „JMSAdmin-und JMS30Admin -Tools konfigurieren” auf Seite 741.

**Anmerkung:** Möglicherweise müssen Sie Ihren LDAP-Server für die Speicherung von Java-Objekten konfigurieren. Weitere Informationen finden Sie in der Dokumentation zu Ihrem LDAP-Server.

## Vorgehensweise

1. Wenn das Verwaltungstool noch nicht gestartet wurde, starten Sie es wie im Abschnitt „Die Tools JMSAdmin und JMS30Admin starten” auf Seite 743 beschrieben.

   Die Eingabeaufforderung wird angezeigt und zeigt an, dass das Tool bereit ist, Verwaltungsbefehle zu akzeptieren.
2. Stellen Sie sicher, dass in der Eingabeaufforderung der Kontext angezeigt wird, in dem das neue Objekt erstellt werden soll.

   Wenn Sie das Verwaltungstool starten, wird die Eingabeaufforderung zunächst wie folgt angezeigt:

   ```
   InitCtx>
   ```

   Um den aktuellen Kontext zu ändern, verwenden Sie das Verb CHANGE wie in „Subkontexte konfigurieren” auf Seite 746 beschrieben.
3. Verwenden Sie die folgende Befehlssyntax, um eine Verbindungsfactory, eine Warteschlangen-Destination oder ein Topic-Ziel zu erstellen:

   ```
   DEFINE TYPE (name) [property]*
   ```

   Geben Sie also das Verb DEFINE ein, gefolgt vom Verweis auf ein verwaltetes Objekt *TYPE* (name), gefolgt von null oder mehr *Eigenschaften* (siehe Eigenschaften von IBM MQ classes for JMS-Objekten).
4. Verwenden Sie die folgende Befehlssyntax, um eine Verbindungsfactory, eine Warteschlangen-Destination oder ein Topic-Ziel zu erstellen:

   ```
   DEFINE TYPE (name) [property]*
   ```
5. Um das neu erstellte Objekt anzuzeigen, verwenden Sie das Verb DISPLAY mit der folgenden Befehlssyntax:

   ```
   DISPLAY TYPE (name)
   ```

**Beispiel**

Das folgende Beispiel zeigt eine Warteschlange mit dem Namen testQueue, die im Ausgangskontext mit dem Verb DEFINE erstellt wurde. Da dieses Objekt in einer LDAP-Umgebung gespeichert wird, obwohl der Objektname `testQueue` nicht mit einem Präfix eingegeben wird, fügt das Tool automatisch eine hinzu, um die Übereinstimmung mit der LDAP-Namenskonvention sicherzustellen. Wenn der Befehl DISPLAY `Q(testQueue)` übergeben wird, wird dieses Präfix ebenfalls hinzugefügt.

JM 3.0 V 9.3.0 V 9.3.0

```
InitCtx> DEFINE Q(testQueue)

InitCtx> DISPLAY CTX

Contents of InitCtx

a cn=testQueue       com.ibm.mq.jakarta.jms.MQQueue

1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

JMS 2.0

```
InitCtx> DEFINE Q(testQueue)

InitCtx> DISPLAY CTX

Contents of InitCtx

a cn=testQueue       com.ibm.mq.jms.MQQueue

1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

### *Beispielfehlerbedingungen bei der Erstellung eines JMS-Objekts*

Wenn Sie ein Objekt erstellen, kann es zu einer Reihe allgemeiner Fehlerbedingungen kommen.

Es folgt ein Beispiel für diese Fehlerbedingungen:

**CipherSpec zugeordnet zu CipherSuite**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

**Ungültige Eigenschaft für Objekt**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

**Ungültiger Typ für Eigenschaftswert**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

**Eigenschaftsüberschneidungen-Client/Bindungen**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

**Eigenschaftsüberschneidung-Initialisierung verlassen**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

**Eigenschaftswert außerhalb des gültigen Bereichs**

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

**Unbekannte Eigenschaft**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

Im Folgenden finden Sie Beispiele für Fehlerbedingungen, die in Windows auftreten können, wenn verwaltete JNDI-Objekte in einer JMS -Anwendung gesucht werden.

1. Wenn Sie den WebSphere-JNDI-Provider com.ibm.websphere.naming.WsnInitialContextFactory verwenden, müssen Sie einen Schrägstrich (/) verwenden, um auf verwaltete Objekte zuzugreifen, die in Subkontexten definiert sind; z. B. jms/MyQueueName. Wenn Sie einen Backslash (\) verwenden, wird eine Ausnahme vom Typ "InvalidNameException" ausgelöst.
2. Wenn Sie den Oracle-JNDI-Provider com.sun.jndi.fscontext.RefFSContextFactory verwenden, müssen Sie einen Backslash (\) verwenden, um auf verwaltete Objekte zuzugreifen, die in Subkontexten definiert sind; z. B. ctx1 \ \fred. Wenn Sie einen Schrägstrich (/) verwenden, wird die Ausnahmebedingung NameNotFoundException ausgelöst.

## JMS-Objekte konfigurieren

Sie können die Verben ALTER, DEFINE, DISPLAY, DELETE, COPY und MOVE verwenden, um verwaltete Objekte im Verzeichnisnamensbereich zu bearbeiten.

### Informationen zu diesem Vorgang

Tabelle 39 auf Seite 749 fasst die Verwendung dieser Verben zusammen. Ersetzen Sie *TYPE* durch das Schlüsselwort, das für das erforderliche verwaltete Objekt steht, wie in „JMS -und Jakarta Messaging -Objekte mit den Verwaltungstools konfigurieren” auf Seite 739 beschrieben.

*Tabelle 39. Syntax und Beschreibung der Befehle, die zum Bearbeiten von verwalteten Objekten verwendet werden*

| **Befehlssyntax** | **Beschreibung** |
|---|---|
| ALTER *TYPE* (Name) [ Merkmal] * | Es wird versucht, die Eigenschaften des verwalteten Objekts mit den angegebenen Objekten zu aktualisieren. Wenn es einen Sicherheitsverstoß gibt, wenn das angegebene Objekt nicht gefunden wird oder wenn die neuen Eigenschaften nicht gültig sind, schlägt fehl. |
| DEFINE *TYPE* (Name) [ Merkmal] * | Es wird versucht, ein verwaltettes Objekt des Typs *TYPE* mit den angegebenen Eigenschaften zu erstellen und unter dem Namen name im aktuellen Kontext zu speichern. Wenn der angegebene Name nicht gültig ist oder ein Objekt mit diesem Namen vorhanden ist oder wenn die angegebenen Eigenschaften nicht gültig sind, schlägt die Sicherheitsverletzung fehl. |

*Tabelle 39. Syntax und Beschreibung der Befehle, die zum Bearbeiten von verwalteten Objekten verwendet werden (Forts.)*

| **Befehlssyntax** | **Beschreibung** |
|---|---|
| DISPLAY *TYPE* (Name) | Zeigt die Eigenschaften des verwalteten Objekts des Typs *TYPE* an, das unter dem Namen name im aktuellen Kontext gebunden ist. Gibt an, ob das Objekt nicht vorhanden ist, oder wenn ein Sicherheitsverstoß vorliegt. |
| DELETE *TYPE* (Name) | Es wird versucht, das verwaltete Objekt des Typs *TYPE* mit dem Namen name aus dem aktuellen Kontext zu entfernen. Gibt an, ob das Objekt nicht vorhanden ist, oder wenn ein Sicherheitsverstoß vorliegt. |
| COPY *TYPE* (NameA)<br>*TYPE* (NameB) | Erstellt eine Kopie des verwalteten Objekts vom Typ *TYPE* mit dem Namen nameA und benennt die Kopie nameB. Dies geschieht im Rahmen des aktuellen Kontexts. Fehlerhafte Objekte, wenn das zu kopierende Objekt nicht vorhanden ist, wenn ein Objekt mit dem Namen nameB vorhanden ist oder wenn ein Sicherheitsverstoß vorliegt. |
| MOVE *TYPE* (NameA) *TYPE* (NameB) | Verschiebt das verwaltete Objekt des Typs *TYPE* mit dem Namen nameA in nameB (umbenennen). Dies geschieht im Rahmen des aktuellen Kontexts. Gibt an, ob das Objekt, das versetzt werden soll, nicht vorhanden ist, wenn ein Objekt mit dem Namen nameB vorhanden ist oder wenn ein Sicherheitsverstoß vorliegt. |

# JMS 2.0 JMS 2.0-Ressourcen in WebSphere Application Server konfigurieren

Um JMS 2.0-Ressourcen in WebSphere Application Server zu konfigurieren, können Sie entweder die Administrationskonsole oder wsadmin-Befehle verwenden.

## Vorbereitende Schritte

JMS 3.0 V 9.3.0 V 9.3.0 Obwohl IBM MQ 9.3 Jakarta Messaging 3.0unterstützt, hat WebSphere Application Server derzeit keine funktional entsprechende Unterstützung. Daher konfigurieren Sie in WebSphere Application Server Java Message Service 2.0 -Ressourcen.

## Informationen zu diesem Vorgang

Java Message Service 2.0 -Anwendungen basieren in der Regel auf extern konfigurierten Objekten, die beschreiben, wie die Anwendung eine Verbindung zu ihrem JMS -Provider und den Zielen, auf die sie zugreift, herstellt. JMS-Anwendungen verwenden das Java Naming Directory Interface (JNDI), um zur Laufzeit auf die folgenden Arten von Objekten zuzugreifen:

- Aktivierungsspezifikationen (von Java EE-Anwendungsservern verwendet)
- Einheitliche Verbindungsfactorys (bei JMS 1.1 und höher werden domänenunabhängige (einheitliche) Verbindungsfactorys domänenspezifischen Warteschlangenverbindungsfactorys und Topicverbindungsfactorys vorgezogen)
- Topic-Verbindungsfactorys (von JMS 1.0-Anwendungen verwendet)
- Warteschlangenverbindungsfactorys (von JMS 1.0-Anwendungen verwendet)
- Warteschlangen
- Themen

Über den IBM MQ -Messaging-Provider in WebSphere Application Serverkönnen Java Message Service -Messaging-Anwendungen (JMS) Ihr IBM MQ -System als externen Provider von JMS -Messaging-Ressourcen verwenden. Um diesen Ansatz zu aktivieren, konfigurieren Sie den IBM MQ-Messaging-Provider in WebSphere Application Server, um JMS-Ressourcen für die Verbindung zu einem beliebigen Warteschlangenmanager im IBM MQ-Netz zu definieren.

Mithilfe von WebSphere Application Server können Sie IBM MQ-Ressourcen für Anwendungen (z. B. Warteschlangenverbindungsfactorys) konfigurieren sowie Nachrichten und Subskriptionen verwalten, die JMS-Zielen zugeordnet sind. Die Verwaltung der Sicherheit erfolgt über IBM MQ.

**Zugehörige Tasks**

IBM MQ und WebSphere Application Server gemeinsam verwenden

**WebSphere Application Server-Themen**

Interoperation mit dem IBM MQ-Messaging-Provider

Messaging mit dem IBM MQ-Messaging-Provider verwalten

Zuordnung von Namen von Administrationskonsolenanzeigen zu Befehlsnamen und IBM MQ-Namen

## JMS 2.0 JMS 2.0-Ressourcen über die Administrationskonsole konfigurieren

Sie können die Administrationskonsole von WebSphere Application Server verwenden, um Aktivierungsspezifikationen, Verbindungsfactorys und Ziele für den IBM MQ JMS-Provider zu konfigurieren.

### Informationen zu diesem Vorgang

Sie können die Administrationskonsole von WebSphere Application Server verwenden, um eine der folgenden Ressourcen zu erstellen, anzuzeigen oder zu ändern:

- Aktivierungsspezifikationen
- Domänenunabhängige Verbindungsfactorys (JMS 1.1 oder höher)
- Warteschlangenverbindungsfactorys
- Topic-Verbindungsfactorys
- Warteschlangen
- Themen

Die folgenden Schritte bieten eine Übersicht über die Verwendungsmöglichkeiten der Administrationskonsole zur Konfiguration von JMS-Ressourcen für die Verwendung mit dem IBM MQ-Messaging-Provider. In jedem Schritt ist der Name des Abschnitts in der Produktdokumentation zu WebSphere Application Server angegeben, in dem Sie weitere Informationen finden. Unter *Verwandte Links* finden Sie Links zu diesen Topics in IBM Documentation.

In einer WebSphere Application Server-Zelle mit mehreren Versionen können Sie IBM MQ-Ressourcen auf Knoten aller Versionen verwalten. Einige Eigenschaften sind jedoch nicht für alle Versionen verfügbar. In dieser Situation werden nur die Eigenschaften dieses bestimmten Knotens in der Administrationskonsole angezeigt.

### Prozedur

Gehen Sie wie folgt vor, um eine Aktivierungsspezifikation zur Verwendung mit dem IBM MQ-Messaging-Provider zu erstellen oder zu konfigurieren:

- Um eine Aktivierungsspezifikation zu erstellen, verwenden Sie den Ressourcenassistenten IBM MQ JMS.

  Sie können entweder alle Details für die Aktivierungsspezifikation über den Assistenten angeben oder Sie können mithilfe einer Clientkanaldefinitionstabelle (CCDT, Client Channel Definition Table) die Verbindungsdetails für IBM MQ angeben. Wenn Sie die Verbindungsdetails mit Hilfe des Assistenten angeben, können Sie entweder Host-und Portinformationen separat eingeben oder, wenn Sie einen Multi-Instanz-Warteschlangenmanager verwenden, Host-und Portinformationen in Form einer Verbin-

dungsnamensliste eingeben. Weitere Informationen finden Sie im Abschnitt *Aktivierungsspezifikation für den IBM MQ-Messaging-Provider erstellen*.

- Wenn Sie die Konfigurationseigenschaften einer Aktivierungsspezifikation anzeigen oder ändern möchten, verwenden Sie hierfür die Anzeige mit den Einstellungen für die Verbindungsfactory des IBM MQ-Messaging-Providers in der Administrationskonsole.

  Diese Konfigurationseigenschaften steuern, wie Verbindungen zu zugeordneten Warteschlangen und Topics erstellt werden. Weitere Informationen hierzu finden Sie im Abschnitt *Aktivierungsspezifikation für den IBM MQ-Messaging-Provider konfigurieren*.

Gehen Sie wie folgt vor, um eine einheitliche Verbindungsfactory, eine Warteschlangenverbindungsfactory oder eine Topic-Verbindungsfactory zur Verwendung mit dem IBM MQ-Messaging-Provider zu erstellen oder zu konfigurieren:

- Wenn Sie eine Verbindungsfactory erstellen möchten, wählen Sie zunächst den Typ der zu erstellenden Verbindungsfactory aus und geben anschließend im Assistenten zum Erstellen von IBM MQ JMS-Ressourcen die Einzelheiten an.

  – Wenn Ihre JMS-Anwendung nur Punkt-zu-Punkt-Messaging verwenden soll, erstellen Sie eine domänenspezifische Verbindungsfactory für die Punkt-zu-Punkt-Messaging-Domäne, die zum Erstellen von Verbindungen speziell für Punkt-zu-Punkt-Messaging verwendet werden kann.

  – Wenn Ihre JMS-Anwendung nur Publish/Subscribe-Messaging verwenden soll, erstellen Sie eine domänenspezifische Verbindungsfactory für die Publish/Subscribe-Messaging-Domäne, die zum Erstellen von Verbindungen speziell für Publish/Subscribe-Messaging verwendet werden kann.

  – Erstellen Sie für JMS 1.1 oder höher eine domänenunabhängige Verbindungsfactory, die sowohl für Punkt-zu-Punkt-Messaging als auch für Publish/Subscribe-Messaging verwendet werden kann, sodass Ihre Anwendung sowohl Punkt-zu-Punkt- als auch Publish/Subscribe-Arbeiten im Rahmen derselben Transaktion ausführen kann.

  Sie können auswählen, ob Sie über den Assistenten alle Details für die Verbindungsfactory angeben möchten oder ob Sie die Verbindungsdetails für IBM MQ mithilfe einer Clientkanaldefinitionstabelle (CCDT) angeben möchten. Wenn Sie die Verbindungsdetails mit Hilfe des Assistenten angeben, können Sie entweder Host-und Portinformationen separat eingeben oder, wenn Sie einen Multi-Instanz-Warteschlangenmanager verwenden, Host-und Portinformationen in Form einer Verbindungsnamensliste eingeben. Weitere Informationen hierzu finden Sie im Abschnitt *Verbindungsfactory für den IBM MQ-Messaging-Provider erstellen*.

Gehen Sie wie folgt vor, um die Konfigurationseigenschaften einer Verbindungsfactory anzuzeigen oder zu ändern:

- Verwenden Sie die Anzeige mit den Einstellungen der Verbindungs-Factory für die Administrationskonsole für den Typ der Verbindungs-Factory, die Sie konfigurieren möchten.

  Mit den Konfigurationseigenschaften wird gesteuert, wie Verbindungen zu zugeordneten Warteschlangen und Topics erstellt werden. Weitere Informationen finden Sie im Abschnitt *Verbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren*, *Warteschlangenverbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren* oder *Topic-Verbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren*.

Gehen Sie wie folgt vor, um ein JMS-Warteschlangenziel für Punkt-zu-Punkt-Messaging mit dem IBM MQ-Messaging-Provider zu konfigurieren:

- Verwenden Sie die Anzeige mit den Warteschlangeneinstellungen des IBM MQ-Messaging-Providers in der Administrationskonsole, um die folgenden Arten von Eigenschaften zu definieren:

  – Allgemeine Eigenschaften, einschließlich Verwaltungseigenschaften und IBM MQ-Warteschlangeneigenschaften.

  – Verbindungseigenschaften, die angeben, wie eine Verbindung zu dem Warteschlangenmanager hergestellt werden soll, der die Warteschlange enthält.

  – Erweiterte Eigenschaften, die das Verhalten von Verbindungen zu Zielen des IBM MQ-Messaging-Providers steuern.

  – Alle angepassten Eigenschaften für das Warteschlangenziel.

Weitere Informationen hierzu finden Sie im Abschnitt *Warteschlange für den IBM MQ-Messaging-Provider konfigurieren*.

Gehen Sie wie folgt vor, um ein JMS-Topic-Ziel für Publish/Subscribe-Messaging mit dem IBM MQ-Messaging-Provider zu erstellen oder zu konfigurieren:

- Definieren Sie auf der Seite mit den Topic-Einstellungen des IBM MQ-Messaging-Providers die folgenden Arten von Eigenschaften:
  - Allgemeine Eigenschaften, einschließlich Verwaltungseigenschaften und IBM MQ-Topic-Eigenschaften.
  - Erweiterte Eigenschaften, die das Verhalten von Verbindungen zu Zielen des IBM MQ-Messaging-Providers steuern.
  - Alle angepassten Eigenschaften für das Warteschlangenziel.

  Weitere Informationen hierzu finden Sie im Abschnitt *Topic für den IBM MQ-Messaging-Provider konfigurieren*.

**Zugehörige Konzepte**

„Warteschlangenmanager mit mehreren Instanzen" auf Seite 546
Multi-Instanz-Warteschlangenmanager sind Instanzen desselben Warteschlangenmanagers, die auf verschiedenen Servern konfiguriert sind. Eine Instanz des Warteschlangenmanagers ist als aktive Instanz definiert, die andere ist eine Standby-Instanz. Wenn die aktive Instanz ausfällt, wird der Multi-Instanz-Warteschlangenmanager automatisch auf dem Standby-Server gestartet.

**Zugehörige Tasks**

„CCDT im Binärformat konfigurieren" auf Seite 45
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

„Publish/Subscribe-Messaging konfigurieren" auf Seite 465
Sie können den Status des eingereihten Publish/Subscribe starten, stoppen und anzeigen. Darüber hinaus können Sie Datenströme hinzufügen und entfernen sowie Warteschlangenmanager aus einer Brokerhierarchie hinzufügen und löschen.

**WebSphere Application Server-Themen**

Aktivierungsspezifikationen des IBM MQ-Messaging-Providers
Aktivierungsspezifikation für den IBM MQ-Messaging-Provider erstellen
Aktivierungsspezifikation für den IBM MQ-Messaging-Provider konfigurieren
Verbindungsfactory für den IBM MQ-Messaging-Provider erstellen
Einheitliche Verbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren
Warteschlangenverbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren
Topic-Verbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren
Warteschlange für den IBM MQ-Messaging-Provider konfigurieren
Topic für den IBM MQ-Messaging-Provider konfigurieren

## JMS 2.0 JMS 2.0-Ressourcen mit wsadmin-Scripting-Befehlen konfigurieren

Mit den wsadmin-Scripting-Befehlen von WebSphere Application Server können Sie Informationen zu Aktivierungsspezifikationen von JMS, Verbindungsfactorys, Warteschlangen und Themen erstellen, ändern, löschen oder anzeigen. Sie können außerdem die Einstellungen für den IBM MQ-Ressourcenadapter anzeigen und verwalten.

### Informationen zu diesem Vorgang

Die folgenden Schritte geben einen Überblick über die Möglichkeiten, wie Sie WebSphere Application Server-wsadmin-Befehle verwenden können, um JMS-Ressourcen für die Verwendung mit dem IBM

MQ-Messaging-Provider zu konfigurieren. Weitere Informationen zur Verwendung dieser Befehle finden Sie in den *Zugehörigen Links* zur Produktdokumentation von WebSphere Application Server.

Wenn Sie einen Befehl ausführen möchten, verwenden Sie das Objekt "AdminTask" des Scripting-Clients "wsadmin".

Nachdem Sie einen Befehl zum Erstellen eines neuen Objekts oder zum Erstellen von Änderungen verwendet haben, speichern Sie die Änderungen in der Masterkonfiguration. Verwenden Sie beispielsweise folgenden Befehl:

```
AdminConfig.save()
```

Geben Sie in der wsadmin-Eingabeaufforderung den folgenden Befehl ein, um eine Liste der verfügbaren Verwaltungsbefehle des IBM MQ-Messaging-Providers sowie eine kurze Beschreibung zu jedem Befehl anzuzeigen:

```
print AdminTask.help('WMQAdminCommands')
```

Geben Sie an der wsadmin-Eingabeaufforderung den folgenden Befehl ein, um eine Übersicht über die Hilfe zu einem bestimmten Befehl anzuzeigen:

```
print AdminTask.help('command_name')
```

## Prozedur

Verwenden Sie die folgenden Befehle, um alle Ressourcen des IBM MQ-Messaging-Providers aufzulisten, die in dem Bereich definiert sind, in dem ein Befehl abgesetzt wird.

- Verwenden Sie zum Auflisten der Aktivierungsspezifikationen den Befehl **listWMQActivationSpecs** .
- Verwenden Sie zum Auflisten der Verbindungsfactorys den Befehl **listWMQConnectionFactories** .
- Verwenden Sie den Befehl **listWMQQueues** , um die Warteschlangenziele aufzulisten.
- Verwenden Sie den Befehl **listWMQTopics** , um die Topicziele aufzulisten.

Verwenden Sie die folgenden Befehle, um eine JMS-Ressource für den Messaging-Provider von IBM MQ in einem bestimmten Bereich zu erstellen.

- Verwenden Sie den Befehl **createWMQActivationSpec** , um eine Aktivierungsspezifikation zu erstellen.

  Sie können entweder eine Aktivierungsspezifikation erstellen, indem Sie alle Parameter angeben, die für die Herstellung einer Verbindung verwendet werden sollen, oder Sie können die Aktivierungsspezifikation so erstellen, dass sie eine Clientkanaldefinitionstabelle (CCDT) verwendet, um den Warteschlangenmanager zu lokalisieren, zu dem eine Verbindung hergestellt werden soll.
- Verwenden Sie zum Erstellen einer Verbindungsfactory den Befehl **createWMQConnectionFactory** mit dem Parameter **-type** , um den Typ der zu erstellenden Verbindungsfactory anzugeben:
  - Wenn Ihre JMS-Anwendung nur Punkt-zu-Punkt-Messaging verwenden soll, erstellen Sie eine domänenspezifische Verbindungsfactory für die Punkt-zu-Punkt-Messaging-Domäne, die zum Erstellen von Verbindungen speziell für Punkt-zu-Punkt-Messaging verwendet werden kann.
  - Wenn Ihre JMS-Anwendung nur Publish/Subscribe-Messaging verwenden soll, erstellen Sie eine domänenspezifische Verbindungsfactory für die Publish/Subscribe-Messaging-Domäne, die zum Erstellen von Verbindungen speziell für Publish/Subscribe-Messaging verwendet werden kann.
  - Erstellen Sie für JMS 1.1 oder höher eine domänenunabhängige Verbindungsfactory, die sowohl für Punkt-zu-Punkt-Messaging als auch für Publish/Subscribe-Messaging verwendet werden kann, sodass Ihre Anwendung sowohl Punkt-zu-Punkt- als auch Publish/Subscribe-Arbeiten im Rahmen derselben Transaktion ausführen kann.

  Der Standardtyp ist domänenunabhängige Verbindungsfactory.

- Verwenden Sie den Befehl **createWMQQueue** , um ein Warteschlangenziel zu erstellen.
- Verwenden Sie zum Erstellen eines Topicziels den Befehl **createWMQTopic** .

Verwenden Sie die folgenden Befehle, um eine JMS-Ressource für den Messaging-Provider von IBM MQ in einem bestimmten Bereich zu ändern.

- Verwenden Sie zum Ändern einer Aktivierungsspezifikation den Befehl **modifyWMQActivationSpec** .

  Sie können den Typ einer Aktivierungsspezifikation nicht ändern. Sie können z. B. nicht eine Aktivierungsspezifikation erstellen, in der Sie alle Konfigurationsdaten manuell eingeben und anschließend eine CCDT-Datei ändern.
- Verwenden Sie zum Ändern einer Verbindungsfactory den Befehl **modifyWMQConnectionFactory** .
- Verwenden Sie den Befehl **modifyWMQQueue** , um ein Warteschlangenziel zu ändern.
- Verwenden Sie den Befehl **modifyWMQTopic** , um ein Topicziel zu ändern.

Verwenden Sie die folgenden Befehle, um eine JMS-Ressource für den Messaging-Provider von IBM MQ in einem bestimmten Bereich zu löschen.

- Verwenden Sie zum Löschen einer Aktivierungsspezifikation den Befehl **deleteWMQActivationSpec** .
- Verwenden Sie zum Löschen einer Verbindungsfactory den Befehl **deleteWMQConnectionFactory** .
- Verwenden Sie den Befehl **deleteWMQQueue** , um ein Warteschlangenziel zu löschen.
- Verwenden Sie zum Löschen eines Topicziels den Befehl **deleteWMQTopic** .

Verwenden Sie die folgenden Befehle, um Informationen zu einer bestimmten Ressource des IBM MQ-Messaging-Providers anzuzeigen.

- Zum Anzeigen aller Parameter und ihrer Werte, die einer bestimmten Aktivierungsspezifikation zugeordnet sind, verwenden Sie den Befehl **showWMQActivationSpec** .
- Um alle Parameter und ihre Werte anzuzeigen, die einer bestimmten Verbindungsfactory zugeordnet sind, verwenden Sie den Befehl **showWMQConnectionFactory** .
- Verwenden Sie den Befehl **showWMQQueue** , um alle Parameter und ihre Werte anzuzeigen, die einem bestimmten Warteschlangenziel zugeordnet sind.
- Verwenden Sie den Befehl **deleteWMQTopic** , um alle Parameter und ihre Werte anzuzeigen, die einem Topicziel zugeordnet sind.

Verwenden Sie die folgenden Befehle, um die Einstellungen für den IBM MQ-Ressourcenadapter oder den IBM MQ-Messaging-Provider zu verwalten.

- Verwenden Sie den Befehl **manageWMQ** , um die Einstellungen des IBM MQ -Ressourcenadapters zu verwalten, der in einem bestimmten Geltungsbereich installiert ist.
- Verwenden Sie zum Anzeigen aller Parameter und ihrer Werte, die mit dem Befehl **manageWMQ** festgelegt werden können, den Befehl **showWMQ** . Diese Einstellungen beziehen sich entweder auf den IBM MQ-Ressourcenadapter oder den IBM MQ-Messaging-Provider. Der Befehl **showWMQ** zeigt außerdem auch eventuelle benutzerdefinierte Eigenschaften an, die auf dem IBM MQ-Ressourcenadapter eingestellt sind.

**Zugehörige Konzepte**

„Warteschlangenmanager mit mehreren Instanzen” auf Seite 546
Multi-Instanz-Warteschlangenmanager sind Instanzen desselben Warteschlangenmanagers, die auf verschiedenen Servern konfiguriert sind. Eine Instanz des Warteschlangenmanagers ist als aktive Instanz definiert, die andere ist eine Standby-Instanz. Wenn die aktive Instanz ausfällt, wird der Multi-Instanz-Warteschlangenmanager automatisch auf dem Standby-Server gestartet.

**Zugehörige Tasks**

„CCDT im Binärformat konfigurieren” auf Seite 45
Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms wird beim Erstellen des Warteschlangenmanagers automatisch eine binäre CCDT mit Standardeinstellungen erstellt. Mit dem Befehl **runmqsc** kann eine binäre CCDT aktualisiert werden.

„Publish/Subscribe-Messaging konfigurieren” auf Seite 465
Sie können den Status des eingereihten Publish/Subscribe starten, stoppen und anzeigen. Darüber hinaus können Sie Datenströme hinzufügen und entfernen sowie Warteschlangenmanager aus einer Brokerhierarchie hinzufügen und löschen.

**WebSphere Application Server-Themen**
**createWMQActivationSpec**-Befehl
**createWMQConnectionFactory**-Befehl
**createWMQQueue**-Befehl
**createWMQTopic**-Befehl
**deleteWMQActivationSpec**-Befehl
**deleteWMQConnectionFactory**-Befehl
**deleteWMQQueue**-Befehl
**deleteWMQTopic**-Befehl
**listWMQActivationSpecs**-Befehl
**listWMQConnectionFactories**-Befehl
**listWMQQueues**-Befehl
**listWMQTopics**-Befehl
**modifyWMQActivationSpec**-Befehl
**modifyWMQConnectionFactory**-Befehl
**modifyWMQQueue**-Befehl
**modifyWMQTopic**-Befehl
**showWMQActivationSpec**-Befehl
**showWMQConnectionFactory**-Befehl
**showWMQQueue**-Befehl
**showWMQTopic**-Befehl
**showWMQ**-Befehl
**manageWMQ**-Befehl

## JMS 2.0 Gemeinsam genutzte JMS 2.0-Subskriptionen verwenden

In WebSphere Application Server traditional 9.0 können Sie gemeinsam genutzte JMS 2.0-Subskriptionen mit IBM MQ 9.0 konfigurieren und verwenden.

### Informationen zu diesem Vorgang

Die JMS 2.0-Spezifikation führte das Konzept der gemeinsam genutzten Subskriptionen ein, mit dem eine einzelne Subskription von einem oder mehreren Konsumenten geöffnet werden kann. Die Nachrichten werden von allen diesen Konsumenten gemeinsam genutzt. Es gibt keine Einschränkung, wenn diese Konsumenten so lang sind, dass sie eine Verbindung zum selben Warteschlangenmanager herstellen.

Gemeinsam genutzte Subskriptionen können entweder permanent oder nicht permanent sein, mit derselben Semantik wie die jetzt als nicht gemeinsam genutzten Subskriptionen bezeichnet werden.

Damit ein Konsument feststellen kann, welche Subskription verwendet werden kann, muss er einen Subskriptionsnamen angeben. Dies ist vergleichbar mit nicht gemeinsam genutzten permanenten Subskriptionen, aber ein Subskriptionsname ist in allen Fällen erforderlich, in denen eine gemeinsam genutzte Subskription erforderlich ist. Eine Client-ID ist jedoch im Fall einer dauerhaften gemeinsam genutzten Subskription; nicht erforderlich; sie kann angegeben werden, aber sie ist nicht obligatorisch.

Während man sich gemeinsam genutzte Subskriptionen als Ladeausgleichsmechanismus vorstellen kann, gibt es weder in IBM MQ noch in der JMS 2.0-Spezifikation eine Zusage, wie die Nachrichten auf die Konsumenten verteilt werden.

In WebSphere Application Server traditional 9.0 ist ein IBM MQ 9.0-Ressourcenadapter bereits installiert.

Die folgenden Schritte zeigen, wie Sie eine Aktivierungsspezifikation für die Verwendung einer gemeinsam genutzten permanenten oder nicht permanenten Subskription über die Administrationskonsole von WebSphere Application Server traditional konfigurieren können.

## Vorgehensweise

Erstellen Sie zuerst die Objekte in JNDI.

1. Erstellen Sie ein Themenziel in JNDI als 'normal' (siehe „JMS 2.0-Ressourcen über die Administrationskonsole konfigurieren” auf Seite 751).
2. Erstellen Sie die Aktivierungsspezifikation (siehe „JMS 2.0-Ressourcen über die Administrationskonsole konfigurieren” auf Seite 751).

   Sie können die Aktivierungsspezifikation mit genau den Eigenschaften erstellen, die Sie benötigen. Wenn Sie eine permanente Subskription verwenden möchten, können Sie sie bei der Erstellung auswählen und einen Namen angeben. Wenn Sie eine nicht permanente Subskription verwenden möchten, können Sie zu diesem Zeitpunkt keinen Namen angeben. Stattdessen müssen Sie eine angepasste Eigenschaft für den Subskriptionsnamen erstellen.

Aktualisieren Sie die Aktivierungsspezifikation, die Sie mit den erforderlichen angepassten Eigenschaften erstellt haben. Es gibt zwei angepasste Eigenschaften, die Sie möglicherweise angeben müssen:

- In allen Fällen müssen Sie eine angepasste Eigenschaft erstellen, um anzugeben, dass diese Aktivierungsspezifikation eine gemeinsam genutzte Subskription verwenden soll.
- Wenn die Subskription als nicht permanent erstellt wurde, muss die Eigenschaft für den Subskriptionsnamen als angepasste Eigenschaft festgelegt werden.

In der folgenden Tabelle ist der gültige Wert aufgeführt, den Sie für jede angepasste Eigenschaft angeben können:

| Eigenschaftenname | Typ | Gültige Werte |
|---|---|---|
| sharedSubscription | Zeichenfolge | true, false |
| subscriptionName | Zeichenfolge | Java-Zeichenfolge für Länge ohne Nullwert |

3. Wählen Sie die Aktivierungsspezifikation aus der Liste aus, die im Formular **Aktivierungsspezifikationssammlung** angezeigt wird.

   Die Details für die Aktivierungsspezifikation werden im Formular **Aktivierungsspezifikationseinstellungen für IBM MQ-Messaging-Provider** angezeigt.
4. Klicken Sie im Formular **Aktivierungsspezifikationseinstellungen für IBM MQ-Messaging-Provider** auf **Angepasste Eigenschaften** .

   Das Formular **Angepasste Eigenschaften** wird angezeigt.
5. Wenn Sie eine nicht permanente Subskription verwenden, erstellen Sie die angepasste Eigenschaft 'subscriptionName'.

   Klicken Sie in der Anzeige **Angepasste Eigenschaften** der Aktivierungsspezifikation auf **Neu** und geben Sie die folgenden Details ein:

   **Name**

   Der Name der angepassten Eigenschaft, in diesem Fall `subscriptionName`.

   **Wert**

   Der Wert für die angepasste Eigenschaft. Sie können die JNDI-Namen im Feld **Wert** verwenden, z. B. `WASSharedSub0ne`.

   **Typ**

   Der Typ der angepassten Eigenschaft. Wählen Sie den angepassten Eigenschaftstyp aus der Liste aus, der in diesem Fall `java.lang.String` sein muss.

6. Erstellen Sie für eine gemeinsam genutzte, permanente und gemeinsam genutzte nicht permanente Subskription die angepasste Eigenschaft "sharedSubscription".

   Klicken Sie in der Anzeige **Angepasste Eigenschaften** der Aktivierungsspezifikation auf **Neu** und geben Sie die folgenden Details ein:

   **Name**

   Der Name der angepassten Eigenschaft, in diesem Fall `sharedSubscription`.

   **Wert**

   Der Wert für die angepasste Eigenschaft. Um anzugeben, dass die Aktivierungsspezifikation eine gemeinsam genutzte Subskription verwendet, setzen Sie den Wert auf `true`. Wenn Sie später die Verwendung einer gemeinsam genutzten Subskription für diese Aktivierungsspezifikation stoppen möchten, können Sie dies tun, indem Sie den Wert dieser angepassten Eigenschaft auf `false`setzen.

   **Typ**

   Der Typ der angepassten Eigenschaft. Wählen Sie den angepassten Eigenschaftstyp aus der Liste aus, der in diesem Fall `java.lang.String` sein muss.
7. Wenn die Eigenschaften festgelegt sind, starten Sie den Anwendungsserver erneut.

   Die nachrichtengesteuerten Beans (MDB) s für die Aktivierungsspezifikationen werden dann beim Eintreffen von Nachrichten gesteuert, aber nur die MDBs senden die gesendeten Nachrichten gemeinsam.

**Zugehörige Konzepte**

Klone und gemeinsam genutzte Subskriptionen

Subskriptionspermanenz

**Zugehörige Tasks**

Ressourcenadapter für eingehende Kommunikation konfigurieren

**Referenzinformationen zu WebSphere Application Server traditional 9.0**

Topic für den IBM MQ-Messaging-Provider konfigurieren

Aktivierungsspezifikationen des IBM MQ-Messaging-Providers

Aktivierungsspezifikation für den IBM MQ-Messaging-Provider erstellen

Aktivierungsspezifikation für den IBM MQ-Messaging-Provider konfigurieren

Angepasste Eigenschaften für Ressourcen des IBM MQ Messaging-Providers JMS konfigurieren

## JMS 2.0 JMS 2.0-Merkmale ConnectionFactory und DestinationLookup verwenden

In WebSphere Application Server traditional 9.0 können die ConnectionFactoryLookup- und DestinationLookup-Eigenschaften einer Aktivierungsspezifikation mit einem JNDI-Namen eines verwalteten Objekts bereitgestellt werden, die vorrangig vor den anderen Akivierungsspezifikationseigenschaften verwendet werden sollen.

### Informationen zu diesem Vorgang

Die JMS 2.0-Spezifikation gibt zwei zusätzliche Eigenschaften in der verwendeten Aktivierungs spezifikation an, die für die Steuerung von nachrichtengesteuerten Beans (Message-Driven Beans, MDBs) verwendet werden. Früher musste jeder Anbieter angepasste Eigenschaften in der Aktivierungsspezifikation angeben, um Details bereitzustellen, die für die Verbindung zu einem Messaging-System erforderlich sind, und um das Ziel zu definieren, von dem Nachrichten abgerufen werden sollen.

Mit den Eigenschaften connectionFactoryLookup und destinationLookup, die inzwischen Standardeigenschaften sind, kann ein JNDI-Name des relevanten Objekts für die Suche und Verwendung angegeben werden. Innerhalb von WebSphere Application Server traditional 9.0 ist ein IBM MQ 9.0-Ressourcenadapter vorinstalliert.

In den folgenden Schritten wird gezeigt, wie diese beiden Eigenschaften über die Administrationskonsole von WebSphere Application Server traditional angepasst und verwendet werden.

## Vorgehensweise

Erstellen Sie zuerst die Objekte in JNDI.

1. Erstellen Sie die ConnectionFactory in JNDI wie üblich (siehe „JMS 2.0-Ressourcen über die Administrationskonsole konfigurieren” auf Seite 751).
2. Erstellen Sie das Ziel (Destination) in JNDI wie üblich (siehe „JMS 2.0-Ressourcen über die Administrationskonsole konfigurieren” auf Seite 751).

   Das Zielobjekt muss die korrekten Werte haben.
3. Erstellen Sie die Aktivierungsspezifikation mit allen erforderlichen Werten (siehe „JMS 2.0-Ressourcen über die Administrationskonsole konfigurieren” auf Seite 751).

   Sie können die Aktivierungsspezifikation mit genau den Eigenschaften erstellen, die Sie benötigen. Sie sollten jedoch die folgenden Aspekte berücksichtigen:

   - Wenn Sie möchten, dass der IBM MQ-Ressourcenadapter die Java EE-Eigenschaften ConnectionFactory und DestinationLookup verwenden soll, ist es weniger relevant, welche Eigenschaften verwendet werden, wenn Sie die Aktivierungsspezifikation erstellen (siehe ActivationSpec-Eigenschaften ConnectionFactoryLookup und DestinationLookup).
   - Eine Eigenschaft, die in der Verbindungs-Factory oder in der Destination noch nicht definiert ist, muss jedoch in der Aktivierungsspezifikation angegeben werden. Daher müssen Sie die Merkmale für Verbindungskonsumenten und zusätzliche Merkmale sowie die Authentifizierungsinformationen definieren, die bei der eigentlichen Erstellung einer Verbindung verwendet werden.
   - Von den Eigenschaften, die in der Verbindungsfactory definiert sind, hat die Eigenschaft "ClientID" eine Sonderverarbeitung. Dies liegt daran, dass ein allgemeines Szenario eine einzige Verbindungs-Factory mit mehreren Aktivierungsspezifikationen verwendet. Dies vereinfacht die Verwaltung, aber die JMS-Spezifikation fordert eindeutige Client-IDs, daher muss die Aktivierungsspezifikation die Möglichkeit haben, alle in der Verbindungsfactory festgelegten Werte außer Kraft zu setzen. Wenn keine Client-ID in der Aktivierungsspezifikation festgelegt ist, wird jeder Wert in der Verbindungsfactory verwendet.

Aktualisieren Sie entweder die Aktivierungsspezifikation, die Sie mit den beiden neuen angepassten Eigenschaften unter Verwendung der WebSphere Application Server-Verwaltungskonsole wie in Schritt „4” auf Seite 759 beschrieben erstellt haben, oder verwenden Sie stattdessen Annotationen, wie in Schritt „5” auf Seite 760 beschrieben.

4. Aktualisieren Sie die Aktivierungsspezifikation in der Verwaltungskonsole von WebSphere Application Server.

   Diese beiden Eigenschaften müssen in der Anzeige "Angepasste Eigenschaften" der Aktivierungsspezifikation festgelegt werden. Diese Eigenschaften sind in den Hauptangaben für die Aktivierungsspezifikation oder im Erstellungsassistenten der Aktivierungsspezifikation nicht vorhanden.

   a) Wählen Sie die Aktivierungsspezifikation aus der Liste aus, die im Formular **Aktivierungsspezifikationssammlung** angezeigt wird.

      Die Details für die Aktivierungsspezifikation werden im Formular **Aktivierungsspezifikationseinstellungen für IBM MQ-Messaging-Provider** angezeigt.

   b) Klicken Sie im Formular **Aktivierungsspezifikationseinstellungen für IBM MQ-Messaging-Provider** auf **Angepasste Eigenschaften** .

      Das Formular **Angepasste Eigenschaften** wird angezeigt.

   c) Erstellen Sie im Formular **Angepasste Eigenschaften** zwei neue angepasste Eigenschaften, die beide den Typ 'java.lang.String' haben.

      Klicken Sie in jedem Fall auf **Neu** und geben Sie dann die folgenden Details für die angepasste Eigenschaft ein:

**Name**

Der Name der angepassten Eigenschaft, entweder `connectionFactoryLookup` oder `destinationLookup`

**Wert**

Der Wert für die angepasste Eigenschaft. Sie können die JNDI-Namen im Feld **Wert** verwenden, z. B. `QuoteCF` und `QuoteQ`.

**Typ**

Der Typ der angepassten Eigenschaft. Wählen Sie den angepassten Eigenschaftstyp aus der Liste aus, der in diesem Fall `java.lang.String` sein muss.

Die implementierte MDB verwendet nun diese Werte, um die Verbindungsfactory und das Ziel zu erstellen. Bei der Implementierung der MDB ist es nicht erforderlich, die JNDI-Wertkonfiguration festzulegen.

5. Verwenden Sie Annotationen anstelle der Aktivierungsspezifikation.

   Es ist möglich, Annotationen im MDB-Code zu verwenden, um auch Werte anzugeben. Wenn Sie z. B. die JNDI-Namen `QuoteCF` und `QuoteQ`verwenden, sieht der Code wie folgt aus:

```
@MessageDriven(activationConfig = {
        @ActivationConfigProperty(propertyName =  "destinationType" , propertyValue =  "ja
vax.jms.Topic" ),
        @ActivationConfigProperty(propertyName =  "destinationLookup" , propertyValue =
"QuoteQ" ),
        @ActivationConfigProperty(propertyName =  "connectionFactoryLookup" , propertyValue
=  "QuoteCF" )}, mappedName =  "LookupMDB" )
        @TransactionAttribute(TransactionAttributeType.REQUIRED)
        @TransactionManagement(TransactionManagementType.CONTAINER)
        publicclass LookupMDB implements MessageListener {
```

**Zugehörige Tasks**

Ressourcenadapter für eingehende Kommunikation konfigurieren

**Referenzinformationen zu WebSphere Application Server traditional 9.0**

Einheitliche Verbindungsfactory für den IBM MQ-Messaging-Provider konfigurieren

Topic für den IBM MQ-Messaging-Provider konfigurieren

Aktivierungsspezifikationen des IBM MQ-Messaging-Providers

Aktivierungsspezifikation für den IBM MQ-Messaging-Provider erstellen

Aktivierungsspezifikation für den IBM MQ-Messaging-Provider konfigurieren

Angepasste Eigenschaften für Ressourcen des IBM MQ Messaging-Providers JMS konfigurieren

# WebSphere Application Server für die Verwendung der neuesten Wartungsstufe für Ressourcenadapter konfigurieren

Um sicherzustellen, dass der IBM MQ-Ressourcenadapter automatisch auf die neueste verfügbare Wartungsstufe aktualisiert wird, wenn Sie WebSphere Application Server-Fixpacks anwenden, können Sie alle Server in Ihrer Umgebung so konfigurieren, dass sie die neueste Version des Ressourcenadapters verwenden, die in dem Fixpack für WebSphere Application Server enthalten ist, das Sie auf die Installation der einzelnen Knoten angewendet haben.

## Vorbereitende Schritte

**Wichtig:**

- JM 3.0 WebSphere Application Server traditional unterstützt derzeit nicht Jakarta EE. Weitere Informationen finden Sie unter IBM MQ resource adapter statement of support.
- Wenn Sie WebSphere Application Server 8.5 oder eine frühere Version auf einer beliebigen Plattform verwenden, installieren Sie den Ressourcenadapter von IBM MQ 8.0 oder höher nicht in den Anwendungsserver. Der Ressourcenadapter von IBM MQ 8.0 oder höher kann nur in einem Anwendungsserver

implementiert werden, der JMS 2.0 unterstützt. WebSphere Application Server 8.5 oder früher unterstützt jedoch nur JMS 1.1.

## Informationen zu diesem Vorgang

Verwenden Sie diese Task, wenn eine der folgenden Bedingungen für Ihre Konfiguration gilt und Sie alle Server in Ihrer Umgebung so konfigurieren möchten, dass sie die neueste Version des IBM MQ-Ressourcenadapters verwenden:

- In den JVM-Protokollen eines beliebigen Anwendungsservers in Ihrer Umgebung werden die folgenden Versionsinformationen für IBM MQ-Ressourcenadapter angezeigt, nachdem WebSphere Application Server 7.0.0 Fix Pack 1 oder höher angewendet wurde:

  ```
  WMSG1703I:RAR implementation Version 7.0.0.0-k700-L080820
  ```

- Die JVM-Protokolle eines beliebigen Anwendungsservers in Ihrer Umgebung enthalten den folgenden Eintrag:

  ```
  WMSG1625E: It was not possible to detect
  der Code des IBM MQ -Messaging-Providers im angegebenen Pfad < null>
  ```

- Ein oder mehrere Knoten wurden zuvor manuell aktualisiert, um eine bestimmte Wartungsstufe für den IBM MQ-Ressourcenadapter zu verwenden, der jetzt durch die neueste Version des Ressourcenadapters, der in der aktuellen Wartungsstufe von WebSphere Application Server enthalten ist, abgelöst wird.

Das Verzeichnis *profile_root*, auf das sich das Beispiel bezieht, ist das Ausgangsverzeichnis für das WebSphere Application Server-Profil. Beispiel: `C:\Program Files\IBM\WebSphere\AppServer1`.

Wenn Sie die folgenden Schritte für alle Zellen und Einzelserverinstallationen in Ihrer Umgebung ausgeführt haben, empfangen Ihre Server automatisch die Wartung für den IBM MQ-Ressourcenadapter, wenn ein neues Fixpack für WebSphere Application Server angewendet wird.

## Vorgehensweise

1. Starten Sie den Anwendungsserver. Wenn das Profil Teil einer Network Deployment-Konfiguration ist, starten Sie den Deployment Manager und alle Knotenagenten. Wenn das Profil einen Verwaltungsagenten enthält, starten Sie den Verwaltungsagenten.
2. Überprüfen Sie die Wartungsstufe des IBM MQ-Ressourcenadapters.
   a) Öffnen Sie ein Fenster mit Eingabeaufforderung und wechseln Sie in das Verzeichnis *profile_root*`\bin` .

      Geben Sie beispielsweise `cd C:\Program Files\IBM\WebSphere\AppServer1\bin` ein.

   b) Starten Sie das Tool "wsadmin", indem Sie `wsadmin.bat -lang jython`eingeben. Wenn Sie dazu aufgefordert werden, geben Sie Ihren Benutzernamen und Ihr Kennwort ein.

   c) Geben Sie den folgenden Befehl ein, und drücken Sie zweimal die Eingabetaste:

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WMQInfo,*")
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean, 'ge
tInfo', '')
```

      Sie können diesen Befehl auch in Jacl ausführen. Weitere Informationen hierzu finden Sie im Abschnitt *Ensuring that servers use the latest available IBM MQ resource adapter maintenance level* in der Produktdokumentation zu WebSphere Application Server.

   d) Suchen Sie die Nachricht `WMSG1703I` in der angezeigten Ausgabe des Befehls, und überprüfen Sie die Ressourcenadapterebene.

      Bei WebSphere Application Server 7.0.1 Fix Pack 5 sollte die Nachricht beispielsweise wie folgt lauten:

      ```
      WMSG1703I: RAR implementation Version 7.0.1.3-k701-103-100812
      ```

      Diese Nachricht zeigt, dass die Version 7.0.1.3-k701-103-100812 ist. Dies ist die korrekte Ressourcenadapterebene für dieses Fixpack. Wenn stattdessen die folgende Nachricht angezeigt wird,

müssen Sie den Ressourcenadapter an die richtige Wartungsstufe für WebSphere Application Server 7.0.1 Fix Pack 5 anpassen.

```
WMSG1703I: RAR implementation Version 7.0.0.0-k700-L080820
```

3. Kopieren Sie das folgende Jython-Script in eine Datei mit dem Namen `convertWMQRA.py` und speichern Sie sie anschließend in das Profilstammverzeichnis, z. B. `C:\Program Files\IBM\WebSphere\AppServer1\bin`.

```
ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
  desc = AdminConfig.showAttribute(ra, "description")
  if (desc == "WAS 7.0 Built In MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In MQ Resource
Adapter"):
    print "Updating archivePath and classpath of " + ra
    AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
    AdminConfig.unsetAttributes(ra, ['classpath'])
    AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
    AdminConfig.save()
  #end if
#end for
```

**Tipp:** Wenn Sie die Datei speichern, stellen Sie sicher, dass sie als Python-Datei und nicht als Textdatei gespeichert wird.

4. Verwenden Sie das wsadmin-Tool von WebSphere Application Server, um das soeben erstellte Jython-Script auszuführen.

   Öffnen Sie eine Eingabeaufforderung und navigieren Sie zum Verzeichnis `\bin` im Ausgangsverzeichnis für das WebSphere Application Server, z. B. `C:\Program Files\IBM\WebSphere\AppServer1\bin`. Geben Sie dann den folgenden Befehl ein und drücken Sie die Eingabetaste:

   ```
   wsadmin -lang jython -f convertWMQRA.py
   ```

   Wenn Sie dazu aufgefordert werden, geben Sie Ihren Benutzernamen und Ihr Kennwort ein.

   **Anmerkung:** Wenn Sie das Script für ein Profil ausführen, das Teil einer Network Deployment-Konfiguration ist, aktualisiert das Script alle Profile, die in dieser Konfiguration aktualisiert werden müssen. Es kann eine vollständige Resynchronisation erforderlich sein, wenn Sie bereits vorhandene Konfigurationsdateiinkonsistenzen haben.

5. Wenn Sie in einer Network Deployment-Konfiguration ausgeführt werden, stellen Sie sicher, dass die Knotenagenten vollständig neu synchronisiert sind. Weitere Informationen finden Sie unter Synchronisieren von Knoten mit dem Scripting-Tool wsadmin oder Hinzufügen, Verwalten und Entfernen von Knoten.
6. Stoppen Sie alle Server im Profil. Wenn das Profil Teil einer Network Deployment-Konfiguration ist, stoppen Sie auch alle Cluster-Member in der Konfiguration, stoppen Sie alle Knotenagenten in der Konfiguration und stoppen Sie den Deployment Manager. Wenn das Profil einen Verwaltungsagenten enthält, stoppen Sie den Verwaltungsagenten.
7. Führen Sie den Befehl **osgiCfgInit** im Verzeichnis *profile_root*`/bin` aus.

   Mit dem Befehl `osgiCfgInit` wird der Klassencache, der von der OSGi-Laufzeitumgebung verwendet wird, neu festgelegt. Wenn das Profil Teil einer Network Deployment-Konfiguration ist, führen Sie den Befehl **osgiCfgInit** im Verzeichnis `profile_root/bin` jedes Profils aus, das Teil der Konfiguration ist.
8. Starten Sie alle Server im Profil erneut. Wenn das Profil Teil einer Netzimplementierungskonfiguration ist, starten Sie auch alle Cluster-Member in der Konfiguration erneut, starten Sie alle Knotenagenten in der Konfiguration erneut, und starten Sie den Deployment Manager erneut. Wenn das Profil einen Verwaltungsagenten enthält, starten Sie den Verwaltungsagenten erneut.
9. Wiederholen Sie Schritt 2, um zu überprüfen, ob der Ressourcenadapter jetzt die richtige Stufe aufweist.

### Nächste Schritte

Wenn Sie nach der Ausführung der in diesem Abschnitt beschriebenen Schritte weiterhin Probleme haben und Sie zuvor die Schaltfläche **Ressourcenadapter aktualisieren** in der Anzeige JMS-Providereinstellungen in der Administrationskonsole von WebSphere Application Server verwendet haben, um den IBM MQ -Ressourcenadapter auf allen Knoten in Ihrer Umgebung zu aktualisieren, ist es möglich, dass das Problem auftritt, das in APAR PM10308 beschrieben ist.

**Zugehörige Konzepte**

IBM MQ-Ressourcenadapter verwenden

**Referenzinformationen für WebSphere Application Server 8.5.5**

Sicherstellen, dass die Server die neueste verfügbare Wartungsstufe des IBM MQ-Ressourcenadapters verwenden

Knoten mit dem Scripting-Tool wsadmin synchronisieren

Knoten hinzufügen, verwalten und entfernen

JMS-Providereinstellungen

## Eigenschaft JMS PROVIDERVERSION konfigurieren

Der IBM MQ-Messaging-Provider hat drei Betriebsmodi, den Normalmodus, den Normalmodus mit Einschränkungen und den Migrationsmodus. Sie können die Eigenschaft JMS **PROVIDERVERSION** festlegen, um auszuwählen, welche dieser Modi eine JMS -Anwendung verwendet, um sie zu veröffentlichen und zu abonnieren.

### Informationen zu diesem Vorgang

Die Auswahl des IBM MQ-Messaging-Provider-Modus kann in erster Linie durch das Festlegen der Eigenschaft PROVIDERVERSION der Verbindungs-Factory gesteuert werden. Der Modus der Operation kann auch automatisch ausgewählt werden, wenn kein Modus angegeben wurde.

Die Eigenschaft **PROVIDERVERSION** unterscheidet zwischen den drei Betriebsmodi des IBM MQ-Messaging-Providers:

**Normalmodus des IBM MQ-Messaging-Providers**
Der normale Modus verwendet alle Funktionen eines IBM MQ-Warteschlangenmanagers, um JMS zu implementieren. Dieser Modus ist für die Verwendung der JMS 2.0-API und der neuen Funktionen optimiert.

**Normalmodus des IBM MQ-Messaging-Providers mit Einschränkungen**
Der normale Modus mit Einschränkungen verwendet die JMS 2.0-API, jedoch nicht die neuen Funktionen, d. h. gemeinsam genutzte Subskriptionen, verzögerte Zustellung und asynchrone Sendebereitstellung.

**Migrationsmodus des IBM MQ-Messaging-Providers**
Im Migrationsmodus können Sie eine Verbindung zu einem Warteschlangenmanager von IBM MQ 8.0 oder höher herstellen, es werden aber keine der Funktionen eines Warteschlangenmanagers von IBM WebSphere MQ 7.0 oder höher, wie z. B. Read-Ahead-und Streaming-Warteschlangenmanager, verwendet.

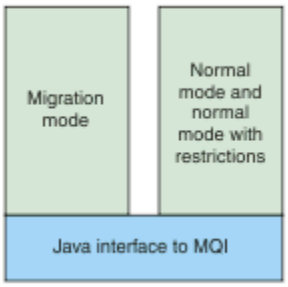



*Abbildung 89. Messaging-Provider-Modi*

### Prozedur

Gehen Sie wie folgt vor, um die Eigenschaft **PROVIDERVERSION** für eine bestimmte Verbindungsfactory zu konfigurieren

- Informationen zum Konfigurieren der Eigenschaft **PROVIDERVERSION** mit IBM MQ Explorer finden Sie im Abschnitt Warteschlangenmanager und Objekte konfigurieren.
- Informationen zum Konfigurieren der **PROVIDERVERSION** -Eigenschaft mit dem JMS -Verwaltungstool finden Sie im Abschnitt Warteschlangenmanager und Objekte konfigurieren .
- Informationen zum Konfigurieren der Eigenschaft **PROVIDERVERSION** in einer JMS-Anwendung mit den IBM JMS-Erweiterungen oder den IBM MQ-JMS-Erweiterungen finden Sie im Abschnitt Verbindungsfactorys und Ziele in einer Anwendung der IBM MQ classes for JMS erstellen und konfigurieren.

Gehen Sie wie folgt vor, um die Einstellungen des Verbindungsfactorys für alle Verbindungsfactorys in der JVM zu überschreiben

- Verwenden Sie die Eigenschaft `com.ibm.msg.client.wmq.overrideProviderVersion` , um die Einstellungen für den Verbindungs-Factor-Modus zu überschreiben.

  Wenn Sie die von Ihnen verwendete Verbindungsfactory nicht ändern können, können Sie mit der Eigenschaft `com.ibm.msg.client.wmq.overrideProviderVersion` alle Einstellungen in der Verbindungsfactory überschreiben. Diese Überschreibung gilt für alle Verbindungsfactorys in der Java Virtual Machine (JVM), die eigentlichen Verbindungsfactory-Objekte werden jedoch nicht geändert.

**Zugehörige Konzepte**

Fehlerbehebung bei JMS -Providerversionen

**Zugehörige Verweise**

PROVIDERVERSION

Verbindungsfactoryeigenschaften

Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten

## Betriebsmodi des IBM MQ-Messaging-Providers

Sie können auswählen, welchen IBM MQ-Messaging-Provider-Modus eine JMS-Anwendung zum Veröffentlichen und Subskribieren verwendet, indem Sie die Eigenschaft PROVIDERVERSION für die Verbindungsfactory auf den entsprechenden Wert setzen. In einigen Fällen wird die Eigenschaft PROVIDERVERSION als nicht angegeben festgelegt. In diesem Fall verwendet der JMS-Client einen Algorithmus, um den zu verwendenden Modus zu bestimmen.

### PROVIDERVERSION-Eigenschaftswerte

Sie können die Eigenschaft **PROVIDERVERSION** der Verbindungsfactory auf einen der folgenden Werte setzen:

**8 - normaler Modus**
Die JMS-Anwendung verwendet den normalen Modus. Dieser Modus nutzt alle Funktionen eines IBM MQ-Warteschlangenmanagers, um JMS zu implementieren.

**7 - normaler Modus mit Einschränkungen**
Die JMS-Anwendung verwendet den normalen Modus mit Einschränkungen. In diesem Modus wird die JMS 2.0-API verwendet; die neuen Funktionen (gemeinsam genutzte Subskriptionen, verzögerte Zustellung oder asynchrones Senden) werden hingegen nicht verwendet.

**6 - Migrationsmodus**
Die JMS-Anwendung verwendet den Migrationsmodus. Im Migrationsmodus nutzt IBM MQ classes for JMS die Funktionen und Algorithmen, die weitgehend den mit IBM WebSphere MQ 6.0 gelieferten entsprechen.

**`nicht angegeben (der Standardwert)`**
Der JMS-Client verwendet einen Algorithmus, um festzustellen, welcher Betriebsmodus verwendet wird.

Der für die Eigenschaft **PROVIDERVERSION** angegebene Wert muss eine Zeichenfolge sein. Um die Option 8, 7 oder 6 anzugeben, kann eines der folgenden Formate verwendet werden:

- V.R.M.F
- V.R.M
- V.R
- V

, wobei V, R, M und F Ganzzahlen größer oder gleich Null sind. Die zusätzlichen Werte R, M und F sind optional und können für eine differenzierte Angabe verwendet werden. Wenn Sie z. B. eine **PROVIDERVERSION** -Version von `7` verwenden möchten, können Sie **PROVIDERVERSION** = `7` , `7.0` , `7.0.0` oder `7.0.0.0` festlegen.

## Typen von Verbindungsfactory-Objekten

Sie können die Eigenschaft **PROVIDERVERSION** für die folgenden Typen von Verbindungsfactory-Objekten festlegen:

- MQConnectionFactory
- MQQueueConnectionFactory
- MQTopicConnectionFactory
- MQXAConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXATopicConnectionFactory

Weitere Informationen zu diesen verschiedenen Verbindungsfactory-Typen finden Sie in „JMS -und Jakarta Messaging -Objekte mit den Verwaltungstools konfigurieren” auf Seite 739.

**Zugehörige Konzepte**
IBM MQ-Messaging-Provider

## *PROVIDERVERSION Normalmodus*

Der normale Modus verwendet alle Funktionen eines IBM MQ-Warteschlangenmanagers, um JMS zu implementieren. Dieser Modus ist für die Verwendung der JMS 2.0-API und der neuen Funktionen optimiert.

Das folgende Ablaufdiagramm zeigt die Prüfungen, die der JMS-Client durchführt, um festzustellen, ob eine Verbindung im Normalmodus hergestellt werden kann.

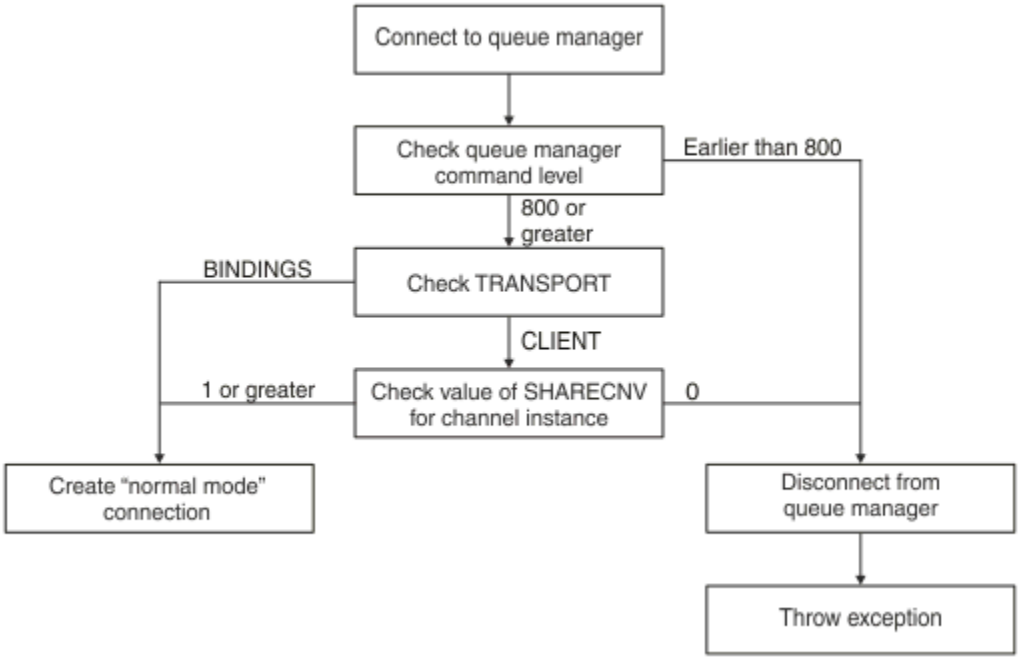



*Abbildung 90. PROVIDERVERSION-Normalmodus*

Wenn der in den Einstellungen der Verbindungs-Factory angegebene Warteschlangenmanager über eine Befehlsebene von 800 oder höher verfügt und die Eigenschaft **TRANSPORT** der Verbindungs-Factory auf BINDINGS gesetzt ist, wird eine normale Modusverbindung erstellt, ohne weitere Merkmale zu überprüfen.

Wenn der in den Einstellungen der Verbindungs-Factory angegebene Warteschlangenmanager über eine Befehlsebene von 800 oder höher verfügt und die Eigenschaft **TRANSPORT** auf CLIENT gesetzt ist, wird die Eigenschaft **SHARECNV** auf dem Serververbindungskanal ebenfalls überprüft. Diese Prüfung ist erforderlich, da der Messaging-Provider von IBM MQ eine gemeinsame Dialognutzung verwendet. Daher muss die Eigenschaft **SHARECNV** , die die Anzahl der Dialoge steuert, die gemeinsam genutzt werden können, für einen erfolgreichen Verbindungsversuch im normalen Modus einen Wert von 1 oder größer haben.

Wenn alle Prüfungen, die im Ablaufdiagramm angezeigt werden, erfolgreich sind, wird eine Verbindung im normalen Modus zum Warteschlangenmanager hergestellt, und alle APIs und Funktionen von JMS 2.0, also asynchrone Sendebereitstellung, verzögerte Zustellung und gemeinsame Subskription, können dann verwendet werden.

Der Versuch, eine normale Modusverbindung zu erstellen, schlägt aus einem der folgenden Gründe fehl:

- Der in den Einstellungen der Verbindungs-Factory angegebene WS-Manager hat eine Befehlsebene, die älter als 800 ist. In diesem Fall schlägt die Methode `createConnection` mit der Ausnahme `JMSFMQ0003` fehl.
- Die Eigenschaft **SHARECNV** auf dem Serververbindungskanal wird auf `0` gesetzt. Wenn diese Eigenschaft keinen Wert 1 oder höher hat, schlägt die Methode `createConnection` mit einer Ausnahme `JMSCC5007` fehl.

**Zugehörige Verweise**

Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten
DEFINE CHANNEL (Eigenschaft SHARECNV)
TRANSPORT

### *PROVIDERVERSION Normalmodus mit Einschränkungen*

Der normale Modus mit Einschränkungen verwendet die JMS 2.0-API, aber nicht die neuen Features IBM MQ 8.0 oder höher, wie z. B. gemeinsam genutzte Abonnements, verzögerte Zustellung oder asynchrones Senden.

Das folgende Ablaufdiagramm zeigt die Prüfungen, die der JMS-Client vornimmt, um zu ermitteln, ob ein normaler Modus mit einer Einschränkung der Verbindung erstellt werden kann.

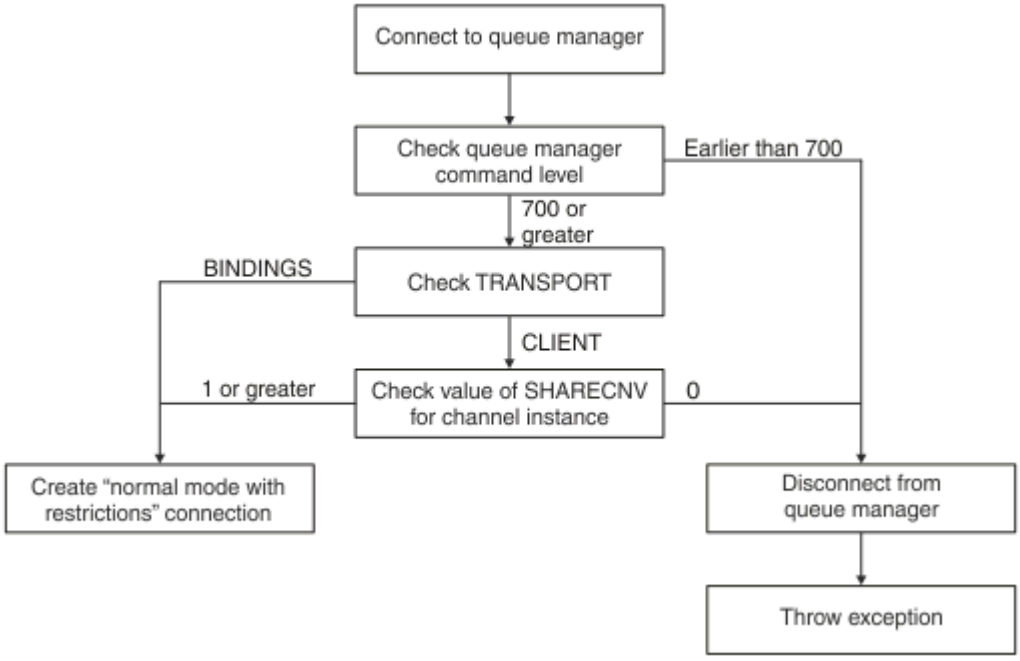



*Abbildung 91. PROVIDERVERSION-normaler Modus mit Einschränkungen*

Wenn der in den Einstellungen der Verbindungs-Factory angegebene Warteschlangenmanager über eine Befehlsebene von 700 oder höher verfügt und die Eigenschaft **TRANSPORT** der Verbindungs-Factory auf BINDINGS gesetzt ist, wird eine normale Modusverbindung erstellt, ohne weitere Merkmale zu überprüfen.

Wenn der in den Einstellungen der Verbindungs-Factory angegebene Warteschlangenmanager über eine Befehlsebene von 700 oder höher verfügt und die Eigenschaft **TRANSPORT** auf CLIENT gesetzt ist, wird die Eigenschaft **SHARECNV** auf dem Serververbindungskanal ebenfalls überprüft. Diese Prüfung ist erforderlich, da der normale Modus des IBM MQ-Messaging-Providers mit Einschränkungen die Funktion für gemeinsame Nutzung von Datenaustausch verwendet. Daher muss die Eigenschaft **SHARECNV** , die die Anzahl der Dialoge steuert, die gemeinsam genutzt werden können, für einen normalen Modus mit Einschränkungen der Verbindungseinschränkung einen Wert von 1 oder größer aufweisen.

Wenn alle Prüfungen, die im Ablaufdiagramm angezeigt werden, erfolgreich sind, wird eine Verbindung im normalen Modus mit Einschränkungen zum Warteschlangenmanager hergestellt, und Sie können dann die JMS 2.0 -API, aber nicht die asynchronen Sende-, Verzögerungs- oder gemeinsam genutzten Subskriptionsfunktionen verwenden.

Der Versuch, einen normalen Modus mit einer Einschränkung der Verbindung zu erstellen, scheitert aus einem der folgenden Gründe:

- Der in den Einstellungen der Verbindungs-Factory angegebene WS-Manager hat eine Befehlsebene, die älter als 700 ist. In diesem Fall schlägt die Methode `createConnection` mit der Ausnahmebedingung `JMSFCC5008` fehl.
- Die Eigenschaft **SHARECNV** auf dem Serververbindungskanal wird auf `0` gesetzt. Wenn diese Eigenschaft keinen Wert 1 oder höher hat, schlägt die Methode `createConnection` mit einer Ausnahme `JMSCC5007` fehl.

**Zugehörige Verweise**

Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten
DEFINE CHANNEL (Eigenschaft SHARECNV)
TRANSPORT

### *Migrationsmodus für PROVIDERVERSION*

Für den Migrationsmodus verwenden die IBM MQ classes for JMS ähnliche Funktionen und Algorithmen, wie mit IBM WebSphere MQ 6.0 bereitgestellt, also beispielsweise eingereihtes Publish/Subscribe, auf der Clientseite implementierte Auswahl, Nicht-Multiplex-Kanäle und Polling zur Implementierung der Listener.

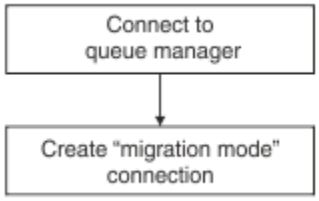



*Abbildung 92. Migrationsmodus PROVIDERVERSION*

Wenn Sie eine Verbindung zu WebSphere Message Broker 6.0 oder WebSphere Message Broker 6.1 mithilfe von IBM MQ Enterprise Transport Version 6.0 herstellen möchten, müssen Sie den Migrationsmodus verwenden.

Sie können auch eine Verbindung zu einem Warteschlangenmanager von IBM MQ 8.0 mit dem Migrationsmodus herstellen, es wird jedoch keine der neuen Funktionen eines IBM MQ classes for JMS-Warteschlangenmanagers verwendet, wie z.B. Vorauslesen oder Streaming. Wenn ein Client für IBM MQ 8.0 oder höher eine Verbindung zu einem Warteschlangenmanager für IBM MQ 8.0 oder höher auf einer verteilten Plattform z/OS oder zu einem Warteschlangenmanager für IBM MQ for z/OS 8.0 oder höher herstellt, wird die Nachrichtenauswahl nicht auf dem Clientsystem, sondern vom Warteschlangenmanager ausgeführt.

Wenn der Migrationsmodus des IBM MQ-Messaging-Providers angegeben ist und die IBM MQ classes for JMS versuchen, eine der JMS 2.0-APIs zu verwenden, schlägt der API-Methodenaufruf mit der Ausnahme `JMSCC5007` fehl.

**Zugehörige Verweise**

Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten

TRANSPORT

### *PROVIDERVERSION nicht angegeben*

Wenn die Eigenschaft **PROVIDERVERSION** einer Verbindungsfactory nicht angegeben wird, verwendet der JMS-Client einen Algorithmus, um festzustellen, welcher Betriebsmodus für die Verbindung zum Warteschlangenmanager verwendet wird. Eine Verbindungsfactory, die im JNDI-Namensbereich mit einer früheren Version von IBM MQ classes for JMS erstellt wurde, nimmt den nicht angegebenen Wert ein, wenn die Verbindungsfactory mit der neuen Version von IBM MQ classes for JMS verwendet wird.

Wenn die Eigenschaft **PROVIDERVERSION** nicht angegeben wird, wird der Algorithmus verwendet, wenn die Methode `createConnection` aufgerufen wird. Der Algorithmus prüft eine Reihe von Verbindungsfactoryeigenschaften, um zu ermitteln, ob der normale Modus des IBM MQ-Messaging-Providers, der normale Modus mit Einschränkungen oder der Migrationsmodus des IBM MQ-Messaging-Providers erforderlich ist. Der normale Modus wird immer zuerst versucht, dann der normale Modus mit Einschränkungen. Wenn keine dieser Verbindungstypen hergestellt werden kann, trennt der JMS-Client die Verbindung zum Warteschlangenmanager und stellt sie dann wieder her, um eine Verbindung im Migrationsmodus zu versuchen.

## BROKERVER-, BROKERQMGR-, PSMODE-und BROKERCONQ -Eigenschaften überprüfen

Die Überprüfung von Eigenschaftswerten beginnt mit der Eigenschaft **BROKERVER** wie in Abbildung 1 dargestellt.

Wenn die Eigenschaft **BROKERVER** auf V1 gesetzt ist, wird die Eigenschaft **TRANSPORT** als Nächstes überprüft, wie in Abbildung 2 dargestellt. Wenn die Eigenschaft **BROKERVER** jedoch auf V2 gesetzt ist, wird

die zusätzliche Prüfung, die in Abbildung 1 angezeigt wird, ausgeführt, bevor die Eigenschaft **TRANSPORT** überprüft wird.

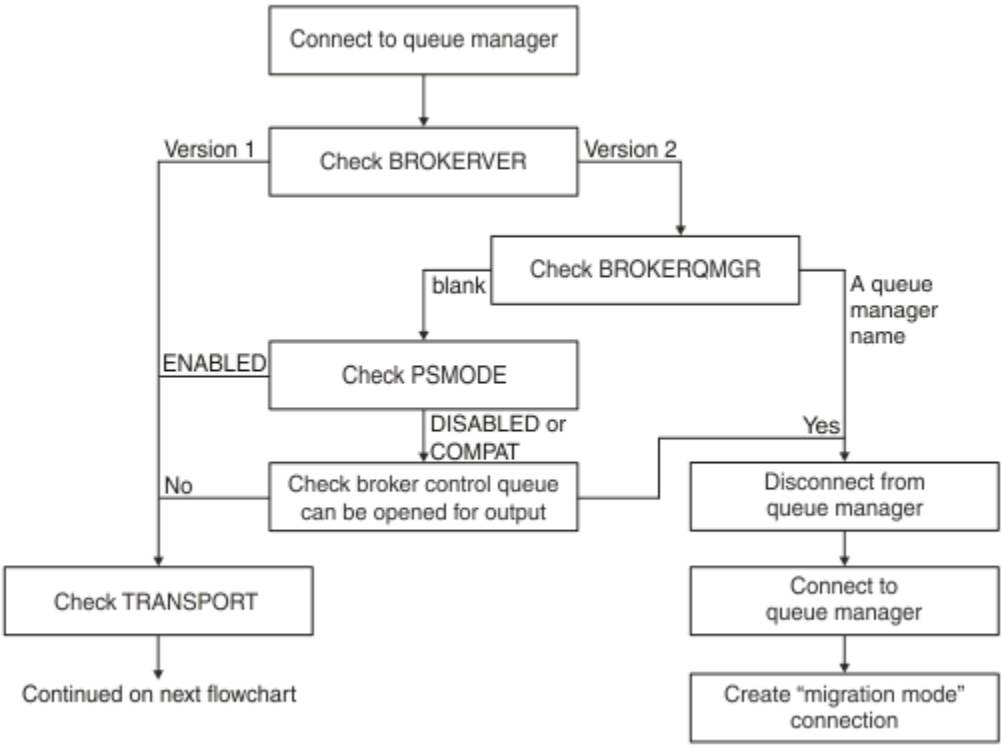



*Abbildung 93. PROVIDERVERSION nicht angegeben*

Wenn die Eigenschaft **BROKERVER** auf V2 gesetzt ist, muss die Eigenschaft **BROKERQMGR** `leer` sein, damit eine Verbindung im normalen Modus möglich ist. Darüber hinaus muss entweder das Attribut **PSMODE** auf dem Warteschlangenmanager auf ENABLED gesetzt sein, oder die durch die Eigenschaft **BROKERCONQ** angegebene Steuerwarteschlange des Brokers darf nicht für die Ausgabe geöffnet werden.

Wenn die Eigenschaftswerte als erforderlich für eine Verbindung im normalen Modus festgelegt sind, wird die nächste Überprüfung mit der Eigenschaft **TRANSPORT** fortgesetzt (siehe Abbildung 2).

Wenn die Eigenschaftswerte nicht für eine Verbindung im normalen Modus festgelegt werden, trennt der JMS-Client die Verbindung zum Warteschlangenmanager und stellt anschließend eine Verbindung im Migrationsmodus her. Dies geschieht in den folgenden Fällen:

- Wenn die Eigenschaft **BROKERQMGR** `leer` ist und das Attribut **PSMODE** auf dem Warteschlangenmanager auf `COMPAT` oder `DISABLED` gesetzt ist und die von der Eigenschaft **BROKERCONQ** angegebene Steuerwarteschlange für den Broker für die Ausgabe geöffnet werden kann (d. h. `MQOPEN` für die Ausgabe erfolgreich ist).
- Gibt an, dass die Eigenschaft **BROKERQMGR** einen Warteschlangennamen angibt.

## Überprüfen der TRANSPORT-Eigenschaft und der Befehlsebene

In Abbildung 2 sind die Prüfungen aufgeführt, die für die **TRANSPORT**-Eigenschaft und die Befehlsebene des Warteschlangenmanagers vorgenommen wurden.

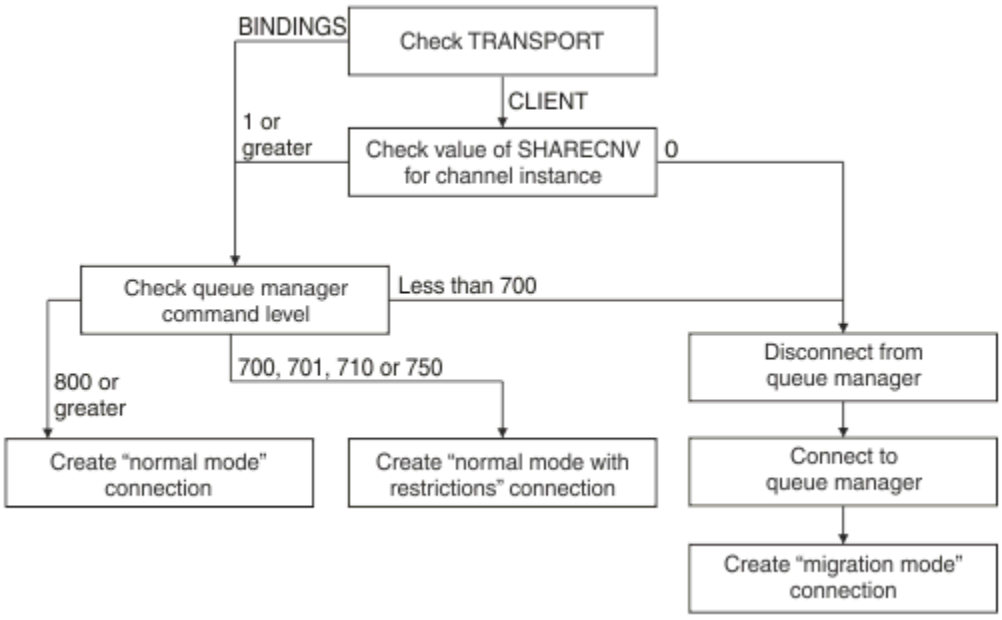



*Abbildung 94. PROVIDERVERSION nicht angegeben (Fortsetzung)*

Eine Verbindung im normalen Modus wird in einem der folgenden Fälle erstellt:

- Die Eigenschaft **TRANSPORT** der Verbindungsfactory ist auf BINDINGS gesetzt, und der Warteschlangenmanager hat eine Befehlsebene von 800 oder höher.
- Die Eigenschaft **TRANSPORT** ist auf CLIENT gesetzt, die Eigenschaft **SHARECNV** auf dem Serververbindungskanal hat den Wert 1 oder höher, und der Warteschlangenmanager hat eine Befehlsebene von 800 oder höher.

Wenn der Warteschlangenmanager die Befehlsebene 750 hat, wird ein normaler Modus mit Einschränkungen für die Verbindung zum Warteschlangenmanager erstellt.

Eine Verbindung im Migrationsmodus wird auch erstellt, wenn die Eigenschaft **TRANSPORT** auf CLIENT gesetzt ist und die Eigenschaft **SHARECNV** auf dem Serververbindungskanal den Wert 0 hat.

**Zugehörige Verweise**

Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten
ALTER QMGR (Attribut PSMODE)
BROKERCONQ
BROKERQMGR
BROKERVER
DEFINE CHANNEL (Eigenschaft SHARECNV)
TRANSPORT

## Informationen zur Providerversion in WebSphere Application Server konfigurieren

Zum Konfigurieren von Informationen zur Providerversion in WebSphere Application Server können Sie die Administrationskonsole oder wsadmin-Befehle verwenden.

### Vorgehensweise

Wenn Sie Informationen zur Providerversion für eine IBM MQ-Verbindungsfactory oder ein Aktivierungsspezifikationsobjekt in WebSphere Application Server konfigurieren möchten, beachten Sie die *Referenzinformationen*. Dort finden Sie Links zu weiteren Informationen in der Produktdokumentation zu WebSphere Application Server.

**Referenzinformationen für WebSphere Application Server 8.5.5**
Einstellungen für die Verbindungsfactory des IBM MQ-Messaging-Providers
**createWMQConnectionFactory**-Befehl
Einstellungen der Aktivierungsspezifikation des IBM MQ-Messaging-Providers
**createWMQActivationSpec**-Befehl
**Referenzinformationen für WebSphere Application Server 8.0.0**
Einstellungen für die Verbindungsfactory des IBM MQ-Messaging-Providers
**createWMQConnectionFactory**-Befehl
Einstellungen der IBM MQ-Aktivierungsspezifikation
**createWMQActivationSpec**-Befehl
**Referenzinformationen für WebSphere Application Server 7.0.0**
Einstellungen für die Verbindungsfactory des IBM MQ-Messaging-Providers
**createWMQConnectionFactory**-Befehl
Einstellungen der IBM MQ-Aktivierungsspezifikation
**createWMQActivationSpec**-Befehl

# Permanente WebSphere Application Server-Subskriptionen entfernen

Wenn Sie den IBM MQ-Messaging-Provider mit WebSphere Application Server 7.0 und WebSphere Application Server 8.0 verwenden, werden permanente Subskriptionen, die von nachrichtengesteuerten Bean-Anwendungen erstellt wurden, die an Aktivierungsspezifikationen gebunden sind, nicht entfernt. Permanente Subskriptionen können entweder mit dem IBM MQ Explorer oder einem IBM MQ-Befehlszeilendienstprogramm entfernt werden.

## Informationen zu dieser Task

Eine nachrichtengesteuerte Bean-Anwendung, die eine permanente Subskription entfernt, kann so konfiguriert werden, dass sie entweder einen Listener-Port oder eine Aktivierungsspezifikation verwendet, vorausgesetzt, dass die Anwendung innerhalb einer WebSphere Application Server 7.0- oder WebSphere Application Server 8.0-Instanz ausgeführt wird, die Normalmodus des IBM MQ-Messaging-Providers verwendet, um eine Verbindung zu IBM MQherzustellen.

Wenn die nachrichtengesteuerte Bean-Anwendung an einen Listener-Port gebunden ist, erstellt der Messaging-Provider von IBM MQ die permanente Subskription für die Anwendung, wenn diese zum ersten Mal gestartet wird. Die permanente Subskription wird entfernt, wenn die nachrichtengesteuerte Bean-Anwendung von einem Anwendungsserver deinstalliert wird und der Anwendungsserver erneut gestartet wird.

Eine nachrichtengesteuerte Bean-Anwendung, die an eine Aktivierungsspezifikation gebunden ist, funktioniert in einer etwas anderen Weise. Die permanente Subskription wird für die Anwendung erstellt, wenn die Anwendung zum ersten Mal gestartet wird. Die permanente Subskription wird jedoch nicht entfernt, wenn die Anwendung deinstalliert und der Anwendungsserver erneut gestartet wird.

Dies kann zu einer Reihe von permanenten Subskriptionen führen, die auf einer Publish/Subscribe-Engine von IBM MQ für Anwendungen verbleiben, die nicht mehr in einem WebSphere Application Server-System installiert sind. Diese Subskriptionen werden als "verwaischere Subskriptionen" bezeichnet und können zu Problemen auf dem Warteschlangenmanager führen, wenn die Publish/Subscribe-Steuerkomponente ausgeführt wird.

Wenn eine Nachricht zu einem Thema veröffentlicht wird, erstellt die IBM MQ Publish/Subscribe-Engine für jede permanente Subskription, die für dieses Thema registriert ist, eine Kopie der Nachricht und stellt sie in eine interne Warteschlange. Die Anwendungen, die diese permanente Subskription verwenden, werden dann die Nachricht aus dieser internen Warteschlange aufnehmen und konsumieren.

Wenn die nachrichtengesteuerte Bean-Anwendung, die diese permanente Subskription verwendet hat, nicht mehr installiert ist, werden die Kopien der veröffentlichten Nachrichten für die Anwendung weiterhin erstellt. Diese Nachrichten werden jedoch nie verarbeitet, was bedeutet, dass eine große Anzahl von Nachrichten in der internen Warteschlange verbleiben kann, die nie entfernt werden.

## Vorbereitungen

Bei der IBM MQ Publish/Subscribe-Engine registrierten Subskriptionen ist ein Subskriptionsname zugeordnet.

Permanente Subskriptionen, die vom WebSphere Application Server IBM MQ-Messaging-Provider für nachrichtengesteuerte Beans erstellt wurden, die an Aktivierungsspezifikationen gebunden sind, weisen einen Subskriptionsnamen im folgenden Format auf:

```
JMS:queue manager name:client identifier:subscription name
```

Dabei gilt Folgendes:

***queue manager name***
Der Name des IBM MQ-Warteschlangenmanagers, auf dem die Publish/Subscribe-Engine ausgeführt wird.

***client identifier***
Dies ist der Wert der Eigenschaft 'Client-ID' der Aktivierungsspezifikation, an die die nachrichtengesteuerte Bean gebunden ist.

***subscription name***
Dies ist der Wert der Aktivierungsspezifikationseigenschaft Subskriptionsname für die Aktivierungsspezifikation, die die nachrichtengesteuerte Bean-Anwendung für die Verwendung konfiguriert hat.

Angenommen, es ist eine Aktivierungsspezifikation vorhanden, die für die Herstellung einer Verbindung zum Warteschlangenmanager testQM eingerichtet wurde. Für die Aktivierungsspezifikation sind die folgenden Eigenschaften festgelegt:

- Client-ID = testClientID
- Subskriptionsname = durableSubscription1

Wenn ein Message-driven Bean, das eine permanente Subskription ausgibt, an diese Aktivierungsspezifikation gebunden ist, erstellt das Bean im Warteschlangenmanager "testQM" der IBM MQ Publish/Subscribe-Engine eine Subskription mit dem folgenden Subskriptionsnamen:

- JMS:testQM:testClientID:durableSubscription1

Die für einen bestimmten Warteschlangenmanager auf der IBM MQ Publish/Subscribe-Engine registrierten Subskriptionen können wie folgt angezeigt werden:

- Die erste Option ist die Verwendung des MQ-Explorers. Wenn MQ Explorer mit einem Warteschlangenmanager verbunden wurde, der für Publish/Subscribe-Arbeit verwendet wird, kann die Liste der Subskribenten, die derzeit bei der Publish/Subscribe-Steuerkomponente registriert sind, angezeigt werden, indem im Navigationsfenster auf den Eintrag `IBM WebSphere MQ ->queue manager name-> Subscriptions` geklickt wird.
- Die andere Möglichkeit, die bei einer Publish/Subscribe-Engine registrierten Subskriptionen anzuzeigen, besteht darin, das IBM MQ Befehlszeilendienstprogramm **runmqsc** zu verwenden und den Befehl **display sub**auszuführen. Rufen Sie dazu eine Eingabeaufforderung auf, wechseln Sie zum Verzeichnis *WebSphere MQ*\bin und geben Sie den folgenden Befehl ein, um **runmqsc**zu starten:

  – `runmqsc queue manager name`

  Geben Sie nach dem Start des Dienstprogramms **runmqsc** den folgenden Befehl ein, um alle permanenten Subskriptionen aufzulisten, die derzeit bei der Publish/Subscribe-Engine registriert sind, die auf dem Warteschlangenmanager ausgeführt wird, mit dem **runmqsc** verbunden ist:

  – `display sub(*) durable`

Gehen Sie wie folgt vor, um zu überprüfen, ob die mit den Publish/Subscribe-Steuerkomponenten registrierten permanenten Subskriptionen noch aktiv

1. Generieren Sie die Liste der permanenten Subskriptionen, die bei der Publish/Subscribe-Engine registriert wurden.
2. Für jede dauerhafte Subskription:

- Sehen Sie sich den Subskriptionsnamen für den permanenten Subskribenten an, und notieren Sie den Wert *client identifier* und *subscription name* .
- Sehen Sie sich die WebSphere Application Server-Systeme an, die eine Verbindung mit dieser Publish/Subscribe-Engine herstellen. Sehen Sie sich an, ob Aktivierungsspezifikationen definiert sind, die die Eigenschaft 'Client-ID' aufweisen, die mit dem Wert von *client identifier* und der Eigenschaft 'Subskriptionsname' übereinstimmt, die mit dem *subscription name* übereinstimmt.
- Wenn keine Aktivierungsspezifikationen gefunden werden, die die Eigenschaften für die Client-ID und den Subskriptionsnamen aufweisen, die mit den Feldern *client identifier* und *subscription name* im IBM MQ-Subskriptionsnamen übereinstimmen, gibt es keine Aktivierungsspezifikationen, die diese permanente Subskription verwenden. Die permanente Subskription kann gelöscht werden.
- Wenn eine Aktivierungsspezifikation definiert ist, die mit dem Namen der permanenten Subskription übereinstimmt, muss die endgültige Prüfung, die vorgenommen werden muss, angezeigt werden, wenn eine nachrichtengesteuerte Bean-Anwendung mit dieser Aktivierungsspezifikation vorhanden ist. Gehen Sie dazu wie folgt vor:
  - Notieren Sie sich den JNDI-Namen der Aktivierungsspezifikation, die die permanente Subskription ausgibt, die Sie gerade untersuchen.
  - Öffnen Sie für jede installierte Message-driven Bean-Anwendung das Konfigurationsfenster der WebSphere Application Server-Administrationskonsole.
  - Klicken Sie im Konfigurationsteilfenster auf den Link Nachrichten-Driven-Bean-Listener-Bindungen.
  - Es wird eine Tabelle mit Informationen zur nachrichtengesteuerten Bean-Anwendung angezeigt. Wenn das Optionsfeld Aktivierungsspezifikation in der Spalte "Bindungen" ausgewählt ist und das Namensfeld Zielressource JNDI den JNDI-Namen für die Aktivierungsspezifikation enthält, die die permanente Subskription ausgeführt hat, wird die Subskription immer noch verwendet und kann nicht gelöscht werden.
  - Wenn keine nachrichtengesteuerten Bean-Anwendungen gefunden werden können, die die Aktivierungsspezifikation verwenden, kann die permanente Subskription gelöscht werden.

## Verfahren

Wenn Sie eine verwaiste permanente Subskription gefunden haben, können Sie diese löschen. Dazu verwenden Sie entweder IBM MQ Explorer oder das IBM MQ-Befehlszeilendienstprogramm **runmqsc**.

So löschen Sie eine verwaiste permanente Subskription mit IBM MQ Explorer:

1. Markieren Sie den Eintrag für die Subskription.
2. Klicken Sie mit der rechten Maustaste auf den Eintrag und wählen Sie **Löschen ...** aus. aus dem Menü. Ein Bestätigungsfenster wird angezeigt.
3. Überprüfen Sie, ob der im Bestätigungsfenster angezeigte Subskriptionsname korrekt ist, und klicken Sie auf **Ja** .

IBM MQ Explorer löscht nun die Subskription von der Publish/Subscribe-Engine und bereinigt die mit ihr verbundenen internen Ressourcen (beispielsweise die noch nicht verarbeiteten Nachrichten, die zum permanent subskribierten Thema veröffentlicht wurden).

Zum Löschen einer verwaisten permanenten Subskription mit dem IBM MQ Befehlszeilendienstprogramm **runmqsc**muss der Befehl **delete sub** ausgeführt werden:

1. Öffnen Sie eine Eingabeaufforderungssitzung.
2. Navigieren Sie zum Verzeichnis *IBM MQ*\bin.
3. Geben Sie den folgenden Befehl ein, um **runmqsc**zu starten:

```
runmqsc queue manager name
```

4. Geben Sie nach dem Start des Dienstprogramms **runmqsc** Folgendes ein:

```
delete sub(Subscription name)
```

Hierbei steht *Subscription name* für den Subskriptionsnamen der permanenten Subskription, die das folgende Format hat:

- JMS:*queue manager name*:*client identifier*:*subscription name*

# Managed File Transfer konfigurieren

Nach der Installation von Managed File Transfer können Sie die Funktionen und Komponenten des Produkts konfigurieren.

Sie können IBM MQ -Hochverfügbarkeitslösungen nutzen, um die Ausfallsicherheit Ihrer Managed File Transfer -Konfiguration zu verbessern. Wenn Ihre Agenten replizierte Datenwarteschlangenmanager (RDQMs) verwenden, müssen Sie sie für die Verwendung der Funktion für variable IP-Adressen konfigurieren. Dies bedeutet, dass Agenten dieselbe IP-Adresse für die Kommunikation mit jeder der drei RDQM-Instanzen verwenden, die derzeit ausgeführt wird, und die Verbindung bei einem Failover automatisch wiederherstellen (siehe RDQM-Hochverfügbarkeit und Variable IP-Adresse erstellen und löschen). Wenn Sie die Multi-Instanz-Warteschlangenmanager-Lösung verwenden, verwenden Anwendungen eine andere IP-Adresse für die Kommunikation mit jeder Instanz, die von der Wiederherstellung der Clientverbindung bei der Übernahme verarbeitet wird (siehe Multi-Instanz-Warteschlangenmanager und Kanal und Clientverbindung).

**Zugehörige Konzepte**

Hinweise und Tipps zur Verwendung von Managed File Transfer

**Zugehörige Tasks**

MFT-Ressourcen überwachen

MFT mit Benutzerexits anpassen

MQMFTCredentials.xml konfigurieren

Managed File Transfer sichern

Programme angeben, die mit MFT ausgeführt werden sollen

Fehlerbehebung für Managed File Transfer

Managed File Transfer verwalten

**Zugehörige Verweise**

MFT-Befehle

Datei MFT `agent.properties`

MFT-Wiederherstellung und Neustart

## MFT-Konfigurationsoptionen unter Multiplatforms

In Managed File Transfer sind eine Reihe von Eigenschaftendateien bereitgestellt, die wichtige Informationen zur Konfiguration enthalten und für den Betrieb erforderlich sind. Diese Eigenschaftendateien befinden sich in dem Konfigurationsverzeichnis, das Sie bei der Installation des Produkts definiert haben.

Sie können mehrere Gruppen von Konfigurationsoptionen haben, jede Gruppe von Konfigurationsoptionen enthält eine Gruppe von Verzeichnissen und Eigenschaftendateien. Wenn in der Befehlszeile nicht explizit andere Werte angegeben werden, werden die in diesen Eigenschaftendateien definierten Werte als Standardparameter für alle Managed File Transfer-Befehle verwendet.

Zum Ändern der Standardgruppe von Konfigurationsoptionen, die Sie verwenden, können Sie den Befehl **fteChangeDefaultConfigurationOptions** verwenden. Die für einen Befehl verwendeten Konfigurationsoptionen können mit dem Parameter **-p** geändert werden, der in jedem Managed File Transfer-Befehl angegeben werden kann.

Der Name einer Gruppe von Konfigurationsoptionen ist der Name des Koordinations-WS-Managers, und es wird empfohlen, dass dies nicht geändert wird. Es ist jedoch möglich, den Namen einer Gruppe von

Konfigurationsoptionen zu ändern, aber Sie müssen den Namen der `config` -und `logs` -Verzeichnisse ändern. In den folgenden Beispielen wird der Name der Gruppe von Konfigurationsoptionen als *coordination_qmgr_name* dargestellt.

## Verzeichnisstruktur der Konfigurationsoptionen

Wenn Sie das Produkt konfigurieren, werden die Verzeichnisse und Eigenschaftendateien in der folgenden Struktur in dem Konfigurationsverzeichnis erstellt. Sie können diese Verzeichnisse und Eigenschaftendateien auch mit den Befehlen **fteSetupCoordination**, **fteSetupCommands**, **fteChangeDefaultConfiguration**und **fteCreateAgent**ändern.

```
MQ_DATA_PATH/mqft/
    config/
       coordination_qmgr_name/
            coordination.properties
            command.properties
            agents/
               agent_name/
                    agent.properties
                    exits
            loggers/
               logger_name
                    logger.properties
    installations/
       installation_name/
            installation.properties
```

Das Verzeichnis *coordination_qmgr_name* ist ein Konfigurationsoptionsverzeichnis. Im Konfigurationsverzeichnis kann es mehrere Konfigurationsoptionsverzeichnisse geben. Das Verzeichnis *agent_name* ist ein Agentenverzeichnis. Neben der Datei `agent.properties` enthält dieses Verzeichnis das Verzeichnis `exits` . Dies ist die Standardposition für Benutzerexitroutinen und verschiedene XML-Dateien, die von den Befehlen **fteCreateBridgeAgent** und **fteCreateCDAgent** generiert werden. Es kann mehr als ein Agentenverzeichnis im Verzeichnis `agents` einer Gruppe von Konfigurationsoptionen geben.

## Eigenschaftendateien

**installation.properties**
: Die Datei `installation.properties` gibt den Namen der Standardgruppe von Konfigurationsoptionen an. Dieser Eintrag verweist Managed File Transfer an eine strukturierte Gruppe mit Verzeichnissen und Eigenschaftendateien, welche die zu verwendende Konfiguration enthält. Gewöhnlich ist der Name einer Gruppe von Konfigurationsoptionen der Name des zugeordneten Koordinations-WS-Managers. Weitere Informationen zu der `installation.properties`-Datei finden Sie in der MFT installation.properties.

**coordination.properties**
: Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden. Weitere Informationen zu der `coordination.properties`-Datei finden Sie in der Datei MFT coordination.properties.

**command.properties**
: Die MFT-Datei 'command.properties' gibt den Befehlswarteschlangenmanager an, zu dem die Verbindung hergestellt werden soll, wenn Sie Befehle ausgeben, und enthält die Informationen, die Managed File Transfer benötigt, um den Kontakt zu diesem Warteschlangenmanager herzustellen. Weitere Informationen zu der `command.properties`-Datei finden Sie in der Die "command.properties" von MFT.

**agent.properties**
: Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.properties`agent.properties, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, kön-

nen in der Datei `agent.properties` angegeben sein. Weitere Informationen zu der `agent.properties`-Datei finden Sie in der Datei MFT `agent.properties`.

**logger.properties**
Die `logger.properties`-Datei gibt die Konfigurationseigenschaften für die Protokollfunktionen an. Weitere Informationen zu der `logger.properties`-Datei finden Sie unter Konfigurationseigenschaften der MFT-Protokollfunktion.

### Eigenschaftendateien und Codepages

Der Inhalt aller Managed File Transfer -Eigenschaftendateien muss aufgrund einer Einschränkung von Javain amerikanischem Englisch bleiben. Wenn Sie Eigenschaftendateien auf einem anderen System als amerikanisches Englisch bearbeiten, müssen Sie Unicode-Escapezeichenfolgen verwenden.

**Zugehörige Verweise**

SSL/TLS-Eigenschaften für MFT

Java-Systemeigenschaften für MFT

fteChangeDefaultConfigurationOptions

fteSetupCommands: MFT-Datei 'command.properties' erstellen

fteSetupCoordination

fteCreateAgent

## z/OS MFT-Konfigurationsoptionen unter z/OS

Die Konfigurationsoptionen für Managed File Transfer sind unter z/OS identisch mit denjenigen für verteilte Plattformen.

Weitere Informationen zu Konfigurationsoptionen unter Multiplatforms finden Sie unter „MFT-Konfigurationsoptionen unter Multiplatforms” auf Seite 774.

Unter z/OS bestimmt die Umgebungsvariable BFG_DATA den Speicherort der Konfiguration. Falls in dem von BFG_DATA referenzierten z/OS UNIX System Services-Verzeichnis noch keine Konfiguration vorhanden ist, generiert das JCL-Script BFGCUSTM eines PDSE-Bibliotheks-Datasets für MFT-Befehle die benötigten Jobs für die Erstellung der Konfiguration. Die Konfiguration wird dann erstellt, wenn Sie diese generierten Jobs ausführen. Die Konfigurationserstellung basiert auf BFG_DATA, die auf ein vorhandenes Verzeichnis verweist, auf das zugegriffen werden kann.

Sie können auch eine Konfiguration erstellen und verwalten, indem Sie dieselben **`fte`** -Befehle verwenden, die auf Multiplatforms und z/OSverfügbar sind. Eine Liste der **`fte`**-Befehle finden Sie unter MFT-Befehle.

**Zugehörige Konzepte**

„MFT-Konfigurationsoptionen unter Multiplatforms” auf Seite 774
In Managed File Transfer sind eine Reihe von Eigenschaftendateien bereitgestellt, die wichtige Informationen zur Konfiguration enthalten und für den Betrieb erforderlich sind. Diese Eigenschaftendateien befinden sich in dem Konfigurationsverzeichnis, das Sie bei der Installation des Produkts definiert haben.

„Agenten erstellen” auf Seite 794
Kopieren Sie die PDSE, um eine agentenspezifische PDSE wie *user.MFT.AGENT1* zu erstellen. Kopieren Sie die PDSE aus einem vorherigen Agenten oder einer vorherigen Konfiguration der Protokollfunktion, falls vorhanden. Wenn es sich um Ihre erste Konfiguration handelt, kopieren Sie die mit MFT bereitgestellte PDSE.

„Koordinationswarteschlangenmanager definieren” auf Seite 792
Managed File Transfer erfordert die Erstellung eines Warteschlangenmanagers, der als Koordinationswarteschlangenmanager fungiert.

**Zugehörige Tasks**

z/OS `MQMFTCredentials.xml` unter z/OS konfigurieren

„Vorhandenes Dataset für MFT-Agenten- oder -Protokollfunktionsbefehle unter z/OS aktualisieren” auf Seite 796

Sie können ein aus dem Managed File Transfer-BefehlsvorlagenDataset erstelltes PDSE-Bibliotheks-Dataset für Managed File Transfer-Befehle aktualisieren.

## Windows Linux Redistributable Managed File Transfer components herunterladen und konfigurieren

Der Redistributable Managed File Transfer package stellt die Redistributable Managed File Transfer Agent bereit, die Sie so konfigurieren können, dass sie eine Verbindung zu einer vorhandenen IBM MQ-Infrastruktur herstellen und die Benutzer in die Lage versetzen, Dateien zu übertragen, ohne dass IBM MQinstalliert werden muss. Ab IBM MQ 9.3.0enthält das weiterverteilbare Paket auch die Redistributable Managed File Transfer Logger.

### Vorbereitende Schritte

Informationen zu Redistributable-Lizenzbedingungen für den Redistributable Managed File Transfer Agent und den Redistributable Managed File Transfer Logger finden Sie unter IBM MQ Redistributable Components.

Die Redistributable Managed File Transfer package-Komponenten stellen die Funktionalität von Managed File Transfer bereit, mit Ausnahme der folgenden Funktionen:

- Für den Redistributable Managed File Transfer Agent werden Bindungsmodusverbindungen zu den Koordinations-, Befehls- und Agentenwarteschlangenmanagern nicht unterstützt. Sie müssen die Clientmodusverbindung verwenden. Bei der Eingabe von Befehlen müssen Sie die Parameter angeben, die bei Verwendung des Managed File Transfer, das als Komponente von IBM MQ installiert ist, optional sind: Host, Port und Name des Warteschlangenmanagers und Kanalname.
- V 9.3.0 Der Redistributable Managed File Transfer Logger unterstützt nur Protokollfunktionen des Typs FILE, die nur im Clientmodus eine Verbindung zum Koordinationswarteschlangenmanager herstellen. Clientmodusverbindungen mit dem Koordinationswarteschlangenmanager für eine Datenbankprotokollfunktion werden nicht unterstützt. Wenn eine Verbindung im Bindungsmodus erforderlich ist, müssen Sie eine Standardinstallation von IBM MQ verwenden.
- V 9.3.0 Ab IBM MQ 9.3.0ist der Befehl **`fteCreateCDAgent.cmd`** nicht mehr enthalten. Eine vollständige Liste der verfügbaren Befehle finden Sie im Abschnitt Installierte MFT-Befehlssätze .
- Managed File Transfer Connect:Direct wird nicht unterstützt.
- IBM MQ Explorer wird nicht bereitgestellt.

Windows Sie müssen die Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019-Bibliotheken, die von Microsoft verfügbar sind, auf Ihrem System installieren, um den Redistributable Managed File Transfer Agent verwenden zu können. Siehe The latest supported Visual C++ downloads.

V 9.3.0 Ab IBM MQ 9.3.0sind auch die Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 -Bibliotheken für Redistributable Managed File Transfer Loggererforderlich.

**Anmerkung:** Advanced Message Security wird mit dem Redistributable Managed File Transfer package nicht unterstützt.

### Informationen zu diesem Vorgang

Sie können optional das Redistributable Managed File Transfer package herunterladen und den Redistributable Managed File Transfer Agent so konfigurieren, dass er eine Verbindung zur vorhandenen IBM MQ-Infrastruktur herstellt, um es Benutzern zu ermöglichen, Dateien zwischen ihrer lokalen Umgebung und der vorhandenen IBM MQ-Infrastruktur zu übertragen, ohne dass sie IBM MQ installieren müssen, um die Managed File Transfer-Funktionalität zu erhalten.

V 9.3.0 Ab IBM MQ 9.3.0 enthält das Redistributable Managed File Transfer package auch die Redistributable Managed File Transfer Logger, mit der Sie eine Dateiprotokollfunktion einrichten können, um eine Verbindung im Clientmodus mit dem Koordinationswarteschlangenmanager herzustellen.

## Vorgehensweise

1. Laden Sie das IBM MQ redistributable Managed File Transfer Agent package von Fix Central herunter.
   a) Wählen Sie das Paket für Ihr Betriebssystem aus.
      Die Namen der Archiv- bzw. ZIP-Dateien beschreiben den Dateiinhalt und geben die entsprechenden Wartungsstufen an. Die Dateinamen haben das folgende Format:
      - Windows *V.R.M.F*-IBM-MQFA-Redist-Win64
      - Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxX64
      - Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxS390X
      - Linux *V.R.M.F*-IBM-MQFA-Redist-LinuxPPC64LE

      Dabei steht *V.R.M.F* für die Versionsnummer, z. B. 9.2.0.0 oder 9.2.1.0.
   b) Geben Sie das Verzeichnis an, in dem das Paket extrahiert werden soll, z. B.:
      - Windows C:\MFTZ
      - Linux /home/MFTZ
2. Extrahieren Sie den Inhalt des heruntergeladenen Pakets:
   - Windows Extrahieren Sie es unter Windows mit den Tools des Windows Explorer.
   - Linux Extrahieren und entpacken Sie es unter Linux wie folgt:

     ```
     gunzip V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar.gz
     ```

     und dann

     ```
     tar xvf V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar
     ```

     Dabei steht *V.R.M.F* für die Versionsnummer, z. B. 9.3.0.0 oder 9.3.1.0.
     Die folgenden Verzeichnisse werden erstellt:
     - Windows Linux bin: Enthält alle erforderlichen MFT-Befehle.
     - Windows bin64: Enthält erforderliche Bibliotheken, die für die 64-Bit-Betriebssystemunterstützung von Windows erforderlich sind.
     - Windows Linux java: Enthält die IBM-JRE- und IBM MQ-Bibliotheken.
     - Windows Linux licenses: Enthält die Lizenzdateien.
     - Windows V 9.3.0 META-INF: Enthält Dateien mit Codesignierinformationen
     - Windows Linux mqft: Enthält ant- und lib-Verzeichnisse, die für die Unterstützung von Ant und für die Unterstützung der Kernfunktionen von MFT erforderlich sind.
     - Windows Linux swtag: Enthält die Datei swidtag, die von Lizenzmanagern für die Identifizierung der Installationen auf der Maschine benötigt wird

## Nächste Schritte

Sie können den Managed File Transfer Agent jetzt konfigurieren. Informationen zu den folgenden Schritten finden Sie unter „Erstkonfiguration für den Redistributable Managed File Transfer Agent erstellen” auf Seite 779.

V 9.3.0 Ab IBM MQ 9.3.0 können Sie auch einen Managed File Transfer Logger konfigurieren. Informationen zu den nächsten Schritten zur Konfiguration der Protokollfunktion finden Sie im Abschnitt „Erstkonfiguration für den Redistributable Managed File Transfer Logger erstellen” auf Seite 781.

**Zugehörige Verweise**

Mögliche Fehler beim Konfigurieren der Redistributable Managed File Transfer components

# Windows Linux Erstkonfiguration für den Redistributable Managed File Transfer Agent erstellen

Sie können einen Managed File Transfer Agent so konfigurieren, dass er eine Verbindung zu einer vorhandenen IBM MQ-Konfiguration herstellt.

## Vorbereitende Schritte

Der Inhalt des Redistributable Managed File Transfer Agent-Pakets muss heruntergeladen und extrahiert worden sein. Weitere Informationen finden Sie unter „Redistributable Managed File Transfer components herunterladen und konfigurieren” auf Seite 777.

## Informationen zu diesem Vorgang

Erstellen Sie zuerst die Umgebung, die für den Redistributable Managed File Transfer Agent erforderlich ist. Anschließend können Sie die Verbindung zum Warteschlangenmanager einrichten, der auf dem IBM MQ-Server ausgeführt wird, und anschließend einen Agenten und den Warteschlangenmanager des Agenten konfigurieren, bevor Sie den Agenten starten und verifizieren.

V 9.3.0 Ab IBM MQ 9.3.0 wird die Umgebung, die Sie erstellen, gemeinsam mit dem Redistributable Managed File Transfer Logger genutzt. Weitere Informationen finden Sie unter „Erstkonfiguration für den Redistributable Managed File Transfer Logger erstellen” auf Seite 781.

## Vorgehensweise

1. Erstellen Sie die Umgebung für den Redistributable Managed File Transfer Agent.

   Wenn Sie den **fteCreateEnvironment** -Befehlausführen, wird das MFT -Datenverzeichnis mit den Konfigurationsinformationen für MFT -Agenten erstellt. Stellen Sie sicher, dass Sie sich in dem `bin`-Verzeichnis befinden, das beim Extrahieren der heruntergeladenen Redistributable Managed File Transfer Agent-Komponente erstellt wurde. Führen Sie den folgenden Befehl aus:

   - Windows

     ```
     fteCreateEnvironment.cmd -d datapath location
     ```

   - Linux

     ```
     . ./fteCreateEnvironment -d datapath location
     ```

   In diesem Befehl werden die folgenden optionalen Parameter verwendet:

   **-d**
   Mit diesem Parameter wird die Position für den Datenpfad angegeben, in dem die MFT-Konfiguration erstellt, gespeichert und verwaltet wird. Wenn Sie den Befehl **fteCreateEnvironment** ohne Angabe der Datenposition ausführen, wird das Verzeichnis `mftdata` an der Position erstellt, an der der Redistributable Managed File Transfer Agent extrahiert wird.

**Anmerkung:** Wenn der weiterverteilbare Agent als Windows-Dienst ausgeführt wird, muss die Umgebungsvariable **BFG_DATA** in der Systemumgebung festgelegt werden, damit der Dienst funktioniert.

**-n *Installationsname***
Mit diesem Parameter wird der Name einer IBM MQ-Installation oder ein eindeutiger Name angegeben.

Beispiele für Situationen, in denen dieser Parameter verwendet werden kann:

- Zum schnellen Testen einer neuen Funktion mithilfe des weiterverteilbaren Pakets in der vorhandenen Konfiguration, in der Agenten so konfiguriert wurden, dass sie Verbindungen zu einem Warteschlangenmanager nur im Clientmodus herstellen können. (Beachten Sie, dass dieser Parameter nicht für einen Agenten gilt, der für die Verbindung zu einem Warteschlangenmanager im Bindungsmodus konfiguriert ist.)
- Bei der Migration aus einer Managed File Transfer-Standardinstallation in ein Redistributable Managed File Transfer Agent-Paket, bei der Sie die gleiche Konfiguration verwenden möchten, die von der Standardinstallation erstellt wurde. Dies ist der Fall, wenn die Standardversion von Managed File Transfer installiert wurde, aber eine Verbindung zu einem Agentenwarteschlangenmanager hergestellt wird, der auf einer anderen Maschine ausgeführt wird.

Die Standardvariable für den Installationsnamen ist **BFG_INSTALLATION_NAME**.

Weitere Informationen zum Befehl **fteCreateEnvironment** finden Sie unter fteCreateEnvironment (Umgebung für Redistributable Managed File Transfer Agent einrichten).

Sie können die Umgebungsvariable *BFG_DATA* auch mit der Datenpfadposition festlegen:

```
BFG_DATA=Datapath location
```

Bevor Sie einen Agenten oder andere Befehle erstellen, starten und stoppen, müssen Sie sicherstellen, dass die Variable *BFG_DATA* auf die richtige Datenpfadposition gesetzt ist.

2. Konfigurieren Sie die IBM MQ-Konnektivität.
   a) Richten Sie den Koordinationswarteschlangenmanager mit dem Befehl **fteSetupCoordination** ein.

   Der Befehl **fteSetupCoordination** erstellt die Gruppe, die für Koordinationswarteschlangenmanager erforderlich ist, und die Verzeichnisse, die für die weitere Konfiguration erforderlich sind. Der Redistributable Managed File Transfer Agent arbeitet im Clientmodus, daher müssen Sie in diesem Befehl zusätzliche Parameter angeben, um Fehler zu vermeiden, da der Bindungsmodus nicht unterstützt wird.

   ```
   fteSetupCoordination -coordinationQMgr PRMFTDEMO2
                -coordinationQMgrHost 9.121.59.233 -coordinationQMgrPort 3002
                -coordinationQMgrChannel SYSTEM.DEF.SVRCONN
   ```

   Weitere Informationen und Schritte für die Verwendung des Befehls **fteSetupCoordination** finden Sie unter fteSetupKoordination. Informationen zur Konfiguration des Koordinationswarteschlangenmanagers finden Sie im Abschnitt „Koordinationswarteschlangenmanager für MFT konfigurieren” auf Seite 824.

   b) Erstellen und konfigurieren Sie den Befehlswarteschlangenmanager:

   ```
   fteSetupCommands -p PRMFTDEMO2 -connectionQMgrHost 9.121.59.233
                -connectionQMgrPort 3002 -connectionQMgrChannel SYSTEM.DEF.SVRCONN
                -connectionQMgr PRMFTDEMO2 -f
   ```

   Weitere Informationen und Schritte für die Verwendung des Befehls **fteSetupCommands** finden Sie unter fteSetupCommands: Erstellen der MFT-Datei 'command.properties'.

3. Erstellen Sie eine MFT-Agentendefinition für einen Endpunkt.

```
fteCreateAgent -p PRMFTDEMO2 -agentQMgrHost 9.121.59.233
            -agentQMgrPort 3002 -agentQMgrChannel SYSTEM.DEF.SVRCONN
            -agentName AGENT.TRI.BANK -agentQMgr PRMFTDEMO2 -f
```

Weitere Informationen zur Verwendung des Befehls **fteCreateAgent** zum Konfigurieren eines Agenten und des Agentenwarteschlangenmanagers finden Sie im Abschnitt fteCreateAgent.

**Anmerkung:** Sie müssen die MQSC-Befehle verwenden, die als Teil der Befehlsausgabe angezeigt werden, um die Agentenobjekte auf dem Agentenwarteschlangenmanager zu definieren. Andernfalls funktionieren die Anweisungen in Schritt „4" auf Seite 781 nicht.

In den Schritten „2" auf Seite 780 und „3" auf Seite 780 für jeden Agenten erstellen Sie Warteschlangen- und Themendefinitionen auf dem Agentenwarteschlangenmanager.

4. Starten Sie den Agenten, und Sie sind bereit, Dateien zu übertragen.

```
fteStartAgent -p PRMFTDEMO2 AGENT.TRI.BANK
```

Sie können den Status des Agenten überprüfen, indem Sie den folgenden Befehl ausführen:

```
fteListAgents
```

Weitere Informationen zur Verwendung des Befehls **fteListAgents** finden Sie im Abschnitt fteListAgenten.

## Nächste Schritte

V 9.3.0 Wenn Sie den Redistributable Managed File Transfer Logger konfigurieren möchten, führen Sie die Schritte in „Erstkonfiguration für den Redistributable Managed File Transfer Logger erstellen" auf Seite 781 aus.

**Zugehörige Konzepte**

„Managed File Transfer konfigurieren" auf Seite 774
Nach der Installation von Managed File Transfer können Sie die Funktionen und Komponenten des Produkts konfigurieren.

„MFT-Konfigurationsoptionen unter Multiplatforms" auf Seite 774
In Managed File Transfer sind eine Reihe von Eigenschaftendateien bereitgestellt, die wichtige Informationen zur Konfiguration enthalten und für den Betrieb erforderlich sind. Diese Eigenschaftendateien befinden sich in dem Konfigurationsverzeichnis, das Sie bei der Installation des Produkts definiert haben.

**Zugehörige Verweise**

**fteCreateTransfer**: Neue Dateiübertragung starten

# Windows Linux V 9.3.0 Erstkonfiguration für den Redistributable Managed File Transfer Logger erstellen

Sie können einen Managed File Transfer Logger mit dem Typ FILE so konfigurieren, dass er im Clientmodus eine Verbindung zu einem Koordinationswarteschlangenmanager herstellt.

## Vorbereitende Schritte

Der Inhalt des Redistributable Managed File Transfer Agent-Pakets muss heruntergeladen und extrahiert worden sein. Ab IBM MQ 9.3.0 enthält dieses Paket auch den Redistributable Managed File Transfer Logger. Weitere Informationen finden Sie unter „Redistributable Managed File Transfer components herunterladen und konfigurieren" auf Seite 777.

## Informationen zu diesem Vorgang

Die Redistributable Managed File Transfer Agent und die Redistributable Managed File Transfer Logger haben dieselbe Umgebung. Nachdem diese Umgebung erstellt und die IBM MQ-Konnektivität konfiguriert wurde, können Sie die Protokollfunktion erstellen und starten.

## Vorgehensweise

1. Stellen Sie sicher, dass die gemeinsam genutzte Umgebung für die Redistributable Managed File Transfer Agent und Redistributable Managed File Transfer Logger wie in Schritt „1" auf Seite 779 beschrieben erstellt wurde und die Konnektivität von IBM MQ wie in Schritt „2" auf Seite 780 von „Erstkonfiguration für den Redistributable Managed File Transfer Agent erstellen" auf Seite 779 beschrieben eingerichtet wurde.

   .
2. Erstellen Sie eine Dateiprotokollfunktion mit dem Befehl **fteCreateLogger**.

   For example:

   ```
   fteCreateLogger FILELOGGER -loggerType FILE -loggerQMgr PRMFTDEMO2
   -loggerQMgrHost 9.121.59.233 -loggerQMgrPort 3003 -loggerQMgrChannel SYSTEM.DEF.SVRCONN
   -fileSize 20MB -fileCount 10 -fileLoggerMode CIRCULAR
   ```

   Weitere Informationen zur Verwendung des Befehls **fteCreateLogger** finden Sie im Abschnitt fteCreateLogger.
3. Starten Sie die Protokollfunktion mit dem Befehl **fteStartLogger**.

   Weitere Informationen zum Befehl **fteStartLogger** finden Sie im Abschnitt fteStartLogger.

**Zugehörige Konzepte**

„Managed File Transfer konfigurieren" auf Seite 774
Nach der Installation von Managed File Transfer können Sie die Funktionen und Komponenten des Produkts konfigurieren.

„MFT-Konfigurationsoptionen unter Multiplatforms" auf Seite 774
In Managed File Transfer sind eine Reihe von Eigenschaftendateien bereitgestellt, die wichtige Informationen zur Konfiguration enthalten und für den Betrieb erforderlich sind. Diese Eigenschaftendateien befinden sich in dem Konfigurationsverzeichnis, das Sie bei der Installation des Produkts definiert haben.

# Windows Linux Upgrade von Redistributable Managed File Transfer components

Sie können ein Upgrade für Redistributable Managed File Transfer components durchführen, indem Sie ein neues Redistributable Managed File Transfer packageherunterladen.

## Vorbereitende Schritte

Informationen zu Redistributable-Lizenzbedingungen für den Redistributable Managed File Transfer Agent und den Redistributable Managed File Transfer Logger finden Sie unter IBM MQ Redistributable Components.

**Anmerkung:** Advanced Message Security wird mit dem Redistributable Managed File Transfer package nicht unterstützt.

## Informationen zu diesem Vorgang

Wenn Sie Redistributable Managed File Transfer componentsbereits installiert haben, können Sie ein Upgrade durchführen, indem Sie ein neues verteilbares Paket herunterladen und den Inhalt an derselben Position extrahieren.

## Vorgehensweise

1. Laden Sie IBM MQ redistributable Managed File Transfer -Agentenpaket für Ihr Betriebssystem von Fix Centralherunter.
2. Stoppen Sie alle Managed File Transfer -Agenten und die Protokollfunktion warten, bis alle aktiven Managed File Transfer -Befehle abgeschlossen sind.

3. Aktualisieren Sie die Dateien für Ihre vorhandene Installation von Redistributable Managed File Transfer components , indem Sie den Inhalt des neuen weiterverteilbaren Pakets, das Sie heruntergeladen haben, in dasselbe Verzeichnis extrahieren, in dem bereits Redistributable Managed File Transfer components installiert ist.

# z/OS Dataset für MFT-Agenten- oder -Protokollfunktionsbefehle erstellen

Aus dem Managed File Transfer-Befehlsvorlagen-Dataset können Sie für eine bestimmte Koordination ein PDSE-Dataset mit Befehlen für einen bestimmten Managed File Transfer Agent oder Managed File Transfer Logger erstellen.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus:

## Vorgehensweise

1. Erstellen Sie eine Kopie der PDSE-Bibliotheksdatei SCSQFCMD der MFT -Befehlsschablone.

   SCSQFCMD muss in eine neue Bibliothek kopiert werden, z. B. *prefix.agent*.JCL. Sie können eine aktualisierte Version des SCSQFCMD-Members (BFGCOPY) mit den folgenden Ersetzungen verwenden:

   - Ersetzen Sie *+ + supplied-library + +* durch den vollständig qualifizierten Namen der SCSQFCMD-PDSE.
   - z/OS Ersetzen Sie *++service-library++* durch den vollständig qualifizierten Namen des neuen PDSE-Bibliotheks-Datasets mit MFT-Befehlen. *++Servicebibliothek++* ist das Ausgabedataset für den Agenten- oder Protokollfunktionsservice, der erstellt wird.
2. Bearbeiten Sie im neuen PDSE-Bibliotheks-Dataset mit den MFT-Befehlen die Teildatei BFGCUSTM (ein JCL-Script), um die Befehle an den Agenten oder die Protokollfunktion anzupassen. Jede Variable ist im Format *++Variablenname++* angegeben. Den Variablennamen müssen Sie durch den erforderlichen Wert ersetzen. Eine Beschreibung der einzelnen JCL-Variablen finden Sie im Abschnitt „z/OS-JCL-Variablen” auf Seite 796. Mit der Datendefinitionsanweisung BFGSTDIN werden Variablen in drei Kategorien definiert: Variables (Variablen), Properties (Eigenschaften) und Environment (Umgebung). Die Anweisung hat das folgende Format:

   ```
   [Variables]
   variable1=value1
   variable2=value2
   ....
   variableN=valueN
   [Properties]
   property1=property value1
   property2=property value2
   ...
   propertyN=property valueN
   [Environment]
   custom_variable1=value1
   custom_variable2=value2
   ....
   custom_variableN=valueN
   ```

   Variablen legen die für jeden Befehl erforderlichen Einrichtungs- und Umgebungsvariablen fest.

   Eigenschaften definieren Überschreibungen der MFT-Konfigurationseigenschaften. Sie können nach Bedarf auch Eigenschaften für Agenten und Protokollfunktionen hinzufügen, um diese in Ihrer Umgebung anzupassen. Eine Liste aller Eigenschaften finden Sie im Abschnitt „Konfigurationseigenschaftendateien” auf Seite 808. Diese Funktion wird bereitgestellt, um Ihnen den Zugriff auf die Eigenschaftendateien der MFT-Konfiguration zu ersparen, die als z/OS UNIX System Services-Dateien vorliegen.

   'Environment' definiert alle weiteren erforderlichen angepassten Umgebungsvariablen.

3. Übergeben Sie den Job BFGCUSTM für das neue PDSE-Bibliotheks-Dataset für MFT-Befehle. Dieser Job generiert den JCL-Befehlssatz passend für den Agenten bzw. die Protokollfunktion in PDSE-Teildateien neu. Eine vollständige Liste der Befehle finden Sie im Abschnitt „JCL-Scripts für Agenten- oder Protokollfunktionsbefehle unter z/OS" auf Seite 800.

   Der Job BFGCUSTM aktualisiert die Bibliothek, die die JCL mit einer Datendefinitionsanweisung mit DISP=OLD enthält. Damit der Job ausgeführt werden kann, müssen Sie den Editor nach der Übergabe des Jobs schließen.

   Prüfen Sie im Ausgabeprotokoll des Jobs, ob das JCL-Script erfolgreich ausgeführt wurde. Falls Fehler ausgegeben wurden, korrigieren Sie diese und übergeben Sie den Job BFGCUSTM erneut.

   Das JCL-Script BFGCUSTM aktualisiert auch die z/OS UNIX System Services MFT -Konfigurationseigenschaftendateien, um die Dateien in Schritt zu halten. Wenn die durch die Eigenschaft CoordinationQMgr definierte Konfiguration nicht vorhanden ist, werden entsprechende Warnungen ausgegeben, und Sie müssen die generierten Jobs BFGCFCR und BFGCMCR ausführen, um die Konfigurationseigenschaftendateien zu erstellen. Sie müssen BFGAGCR für einen Agenten ausführen und BFGLGCRS für eine Protokollfunktionsbearbeitung ausführen. Andernfalls wird die Konfiguration mit den im JCL-Script BFGCUSTM definierten Eigenschaften aktualisiert.

**Zugehörige Konzepte**

„MFT-Konfigurationsoptionen unter z/OS" auf Seite 776
Die Konfigurationsoptionen für Managed File Transfer sind unter z/OS identisch mit denjenigen für verteilte Plattformen.

**Zugehörige Tasks**

„Vorhandenes Dataset für MFT-Agenten- oder -Protokollfunktionsbefehle unter z/OS aktualisieren" auf Seite 796
Sie können ein aus dem Managed File Transfer-BefehlsvorlagenDataset erstelltes PDSE-Bibliotheks-Dataset für Managed File Transfer-Befehle aktualisieren.

## z/OS Managed File Transfer for z/OS konfigurieren

Bei Managed File Transfer for z/OS ist eine Anpassung der Komponente erforderlich, damit diese ordnungsgemäß funktioniert.

### Informationen zu diesem Vorgang

Sie müssen wie folgt vorgehen:

1. Bearbeiten Sie eine PDSE-Teildatei, um Konfigurationsdaten anzugeben
2. Definieren Sie den Koordinationswarteschlangenmanager
3. Definieren Sie den Befehlswarteschlangenmanager
4. Konfigurieren Sie einen oder mehrere Agenten
5. Optional: Konfigurieren Sie eine Protokollierungstask zum Speichern von Daten in Db2

Die Reihenfolge der Tasks, die Sie ausführen müssen, wird in den folgenden Abschnitten ausführlich behandelt.

**Zugehörige Konzepte**

„MFT-Konfiguration überprüfen" auf Seite 784
Vor Beginn müssen Sie die Konfiguration überprüfen.

**Zugehörige Tasks**

Installieren von IBM MQ Advanced for z/OS

## z/OS MFT-Konfiguration überprüfen

Vor Beginn müssen Sie die Konfiguration überprüfen.

Bei Managed File Transfer (MFT) ist für jede MFT-Konfiguration mindestens ein Warteschlangenmanager für die folgenden Rollen erforderlich:

- Ein Koordinationswarteschlangenmanager, der Informationen zum Status der einzelnen Agenten in der Konfiguration enthält, die zu einem Thema im Koordinator veröffentlicht wurden.
- Mindestens ein Befehls- oder Verbindungswarteschlangenmanager, der für MFT-Befehle als Einstiegspunkt in das IBM MQ-Netz dient.
- Mindestens ein Agentenwarteschlangenmanager, der die Kommunikation zwischen einem MFT-Agenten und dem IBM MQ-Netz ermöglicht.

Jede der oben aufgeführten Rollen kann von einem eigenen Warteschlangenmanager übernommen werden; diese Rollen können jedoch auch zusammengefasst werden und in der einfachsten Konfiguration können alle Rollen von einem einzigen Warteschlangenmanager übernommen werden.

Wenn einer bereits vorhandenen MFT-Umgebung ein z/OS-Warteschlangenmanager hinzugefügt wird, müssen Sie die Verbindung zwischen diesem z/OS-Warteschlangenmanager und den anderen Warteschlangenmanagern in der Konfiguration definieren. Dies ist entweder über manuell definierte Übertragungswarteschlangen oder über Clustering möglich.

Jeder MFT-Agent kommuniziert mit einem einzigen Warteschlangenmanager. Kommunizieren mehrere Agenten mit demselben Warteschlangenmanager, müssen in dem Agentenwarteschlangenmanager für jeden Agenten mehrere Warteschlangen definiert sein:

- SYSTEM.FTE.COMMAND.*Agentenname*
- SYSTEM.FTE.DATA.*Agentenname*
- SYSTEM.FTE.REPLY.*Agentenname*
- SYSTEM.FTE.STATE.*Agentenname*
- SYSTEM.FTE.EVENT.*Agentenname*
- SYSTEM.FTE.AUTHAGT1.*Agentenname*
- SYSTEM.FTE.AUTHTRN1.*Agentenname*
- SYSTEM.FTE.AUTHOPS1.*Agentenname*
- SYSTEM.FTE.AUTHSCH1.*Agentenname*
- SYSTEM.FTE.AUTHMON1.*Agentenname*
- SYSTEM.FTE.AUTHADM1.*Agentenname*

Sie können generische Sicherheitsprofile unter Verwendung eines Profils wie `SYSTEM.FTE.COMMAND.*` definieren oder Sie können für jeden Agenten ein eigenes Profil definieren.

**Zugehörige Konzepte**

„Bevor Sie mit der Konfiguration von MFT für z/OS beginnen.” auf Seite 785
Die Managed File Transfer-Konfiguration (MFT-Konfiguration) verwendet Dateien in z/OS UNIX System Services (z/OS UNIX) und PDSE-Datasets.

**Zugehörige Verweise**

MFT-Systemwarteschlangen und der Systemabschnitt

## z/OS Bevor Sie mit der Konfiguration von MFT für z/OS beginnen.

Die Managed File Transfer-Konfiguration (MFT-Konfiguration) verwendet Dateien in z/OS UNIX System Services (z/OS UNIX) und PDSE-Datasets.

Die meisten Konfigurationen und Operationen werden mit JCL aus einer PDSE ausgeführt, und Sie müssen sich mit der Arbeit in einer z/OS UNIX-Umgebung vertraut machen.

Sie können auf OMVS über ISPF zugreifen, oder Sie können eine Telnet-Typ-Sitzung mit Befehlen auf Ihrer Workstation verwenden, z. B. Telnet Putty oder SSH.

Wenn Sie OMVS über ISPF verwenden, können Sie den ISPF und die Anzeigebefehle **oedit** und **obrowse**verwenden.

Sie müsse mit den folgenden z/OS UNIX-Befehlen vertraut sein:

*Tabelle 40. Allgemeine z/OS UNIX-Befehle*

| **Befehl** | **Funktion** |
|---|---|
| chmod xxx-Pfad | Ändern Sie die Zugriffsberechtigungen für Dateien. |
| df -k Pfad | Dokumentiert, wie viel freier Speicherbereich im Dateisystem verbleibt. -k gibt den freien Speicherbereich in KB an. |
| du -kt Pfad | Dokumentiert die Größe von Verzeichnissen unter dem Pfad. Die Größe wurde in KB angegeben. |
| find path -name xxx | Suchen Sie die Datei xxxx im Pfadverzeichnis. Bei xxx wird die Groß-/Kleinschreibung beachtet und kann wie *zzz angegeben werden. |
| ls -ltrd Verzeichnis | Listet Informationen zu dem angegebenen Verzeichnis und nicht zu den Dateien im Verzeichnis auf. |
| ls -ltr Pfad | Listet Informationen zu den Dateien im Pfad auf. |
| obrowse dateiname | Durchsuchen Sie den Dateinamen. |
| oedit-Dateiname | Bearbeiten Sie eine Datei in OMVS. |

Überprüfen Sie die Elemente in der folgenden Tabelle und führen Sie die Tabelle mit den entsprechenden Einträgen für Ihr Unternehmen aus. Sie benötigen diese Werte, wenn Sie Member BFGCUSTM bearbeiten.

*Tabelle 41. Parameter für Member BFGCUSTM erforderlich*

| **Name** | **Beispieldaten** | **Kommentare** |
|---|---|---|
| ADMIN_JOB1 | | Jobkarte. Alle Jobs werden mit derselben JCL-Karte generiert. |
| armELEMENT | Wenn ARM verwendet wird, verwenden Sie den ARM-ELEMENT-Wert, der in der ARM-Richtlinie für diesen Agenten oder diese Protokollfunktion angegeben ist. Wenn ARM nicht verwendet wird, setzen Sie diesen Parameter auf ein Leerzeichen, z. B. armELEMENT= | |
| armELEMTYPE | Wenn ARM verwendet wird, verwenden Sie den ARM-ELEMTYPE, der in der ARM-Richtlinie angegeben ist. Beispiel: armELEMTYPE=SYSBFGAG für einen Agenten oder armELEMTYPE=SYSBFGLG für eine Protokollfunktion. Wenn ARM nicht verwendet wird, setzen Sie diesen Parameter auf ein Leerzeichen, z. B. armELEMTYPE= | |
| BFG_DATA | | Führen Sie die erforderlichen Schritte |
| BFG_GROUP_NAME | MQM | |

| *Tabelle 41. Parameter für Member BFGCUSTM erforderlich (Forts.)* | | |
|---|---|---|
| **Name** | **Beispieldaten** | **Kommentare** |
| BFG_JAVA_HOME | `/java/java71_bit64_GA/J7.1_64/` | |
| BFG_JVM_PROPERTIES | | Führen Sie die erforderlichen Schritte |
| BFG_PROD | `/mqm/V9R2M0/mqft` | Der vollständige Pfad zum Verzeichnis `mqft` im Verzeichnis IBM MQ for z/OS UNIX System Services Components . |
| BFG_WTO | JA | Zum Abrufen einer MFT-Nachricht im Systemprotokoll. |
| CLEAN_AGENT_PROPS | -trs | Dieser Parameter gibt die Optionen an, die verwendet werden, um einen Agenten zu bereinigen, wenn das BFGAGCL-Member ausgeführt wird. Weitere Informationen zu den gültigen Werten für diesen Parameter finden Sie unter fteCleanAgent: MFT-Agenten bereinigen. |
| coordinationQMgr | MQPV | Obligatorische Konfiguration |
| CREDENTIAL_PATH | | Wird bei der Migration verwendet. |
| Db2_HLQ | SYS2.Db2.V10 | |
| DB_PROPS_PATH | | Wird bei der Migration verwendet. |
| FTE_CONFIG | | Wird bei der Migration verwendet. |
| JOBCARD1 | | Dies ist die Jobkarte für die Tasks mit langer Laufzeit, Agenten und Protokollfunktionen. |
| SPEICHERARCHIV | SCEN.FTE.JCL | Name der MFT-PDSE. Sie benötigen eine Kopie für jede Agent-oder Protokollfunktionstask. |
| MQ_HLQ | Das übergeordnete Qualifikationsmerkmal für IBM MQ-Datasets. Beispiel: `MQM.V920` | |
| MQ_LANG | E | |
| MQ_PATH | `/mqm/V9R2M0` | Der vollständige Verzeichnispfad zur Installation von IBM MQ for z/OS UNIX System Services Components. |
| NAME | AGENT1 | |
| OUTPUT_CLASS | * | |
| PATH | `bin:/usr/bin:/usr/sbin` | |

| *Tabelle 41. Parameter für Member BFGCUSTM erforderlich (Forts.)* | | |
|---|---|---|
| **Name** | **Beispieldaten** | **Kommentare** |
| productId | ADVANCEDVUE | Dieser Parameter dient dazu, den Produkttyp festzulegen, für den die Nutzung von Managed File Transfer aufgezeichnet werden soll. Informationen zu den gültigen Werten für diesen Parameter finden Sie in fteSetProductId: set z/OS SCRT recording product id. |
| QMGR | MQPV | |
| SERVICE_TYP | AGENT oder LOGGER | |
| TMPDIR | `/tmp` | z/OS UNIX-Pfad mit Lese- und Schreibzugriff für temporäre Dateien. |

Zudem müssen die folgenden Variablen überprüft und ggf. Werte angegeben werden:

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=
- connectionQMgrChannel=

Diese Eigenschaften werden häufig mit AGENT oder LOGGER verwendet.

**Anmerkung:** Host, Port und Kanal sind für die Clientverbindung erforderlich, sollten aber für eine Bindungsverbindung auf der lokalen Maschine leer gelassen werden.

**Zugehörige Konzepte**

„Zu prüfende Elemente" auf Seite 788
Stellen Sie sicher, dass Sie über einen ausreichenden Festplattenspeicher und ein Verzeichnis zur Datenspeicherung verfügen und die erforderlichen Dateien vorhanden sind.

„Teildatei BFGCUSTM bearbeiten" auf Seite 791
Vor der Ausführung des Jobs müssen Sie die Teildatei BFGCUSTM bearbeiten und die Parameterwerte Ihres Unternehmens eingeben.

## z/OS Zu prüfende Elemente

Stellen Sie sicher, dass Sie über einen ausreichenden Festplattenspeicher und ein Verzeichnis zur Datenspeicherung verfügen und die erforderlichen Dateien vorhanden sind.

### Größe des vorhandenen Festplattenspeichers überprüfen

Überprüfen Sie, ob auf dem Dateisystem, in dem die konfigurationsspezifischen Dateien gespeichert werden sollen, ausreichend Festplattenspeicher vorhanden ist.

Wenn ein Agententrace aktiviert ist, kann dieser standardmäßig einen Speicherplatz von 100 MB einnehmen.

Die Konfigurationsdateien selbst sind lediglich wenige KB groß.

Bei Verwendung von zwei Agenten und einer Protokollfunktion sind mindestens 30 MB erforderlich. Sie können den Befehl **df -k** `path`verwenden, wobei `path` die Position der installationsspezifischen Dateien ist. Dies gibt den verfügbaren und den gesamten Speicherbereich in KB an.

300 MB entsprechen 307.200 KB, daher sollte ein Speicherplatz von mindestens 310.000 KB vorhanden sein.

## Verzeichnis zur Speicherung der Managed File Transfer-Daten erstellen und überprüfen

Sie benötigen ein Verzeichnis zur Speicherung der Managed File Transfer-Daten (MFT-Daten).

Überprüfen Sie, ob Sie genügend Speicherplatz im Dateisystem `df -k /var` haben. Dieses Dateisystem sollte über mindestens 310.000 KB freien Speicher verfügen.

Wenn Sie dieses Dateisystem noch nicht erstellt haben, können Sie dies mithilfe des Befehls **mkdir**, beispielsweise mit **mkdir /var/mft** tun.

Zeigen Sie mit dem Befehl **ls -ltrd /var/mft**an, welche Berechtigungen Benutzer für dieses Verzeichnis haben.

Wenn der Eigner oder die Gruppe nicht korrekt ist, verwenden Sie den Befehl **chown owner:group /var/mft**.

Wenn die Berechtigungen für die Gruppe nicht korrekt sind, können Sie dem Berechtigungsinhaber und der Gruppe mithilfe des folgenden Befehls Lese- und Schreibzugriff bzw. eine Ausführungsberechtigung gewähren. Hinweis: mit dem folgenden Befehl wird auch allen Benutzern Lesezugriff und eine Ausführungsberechtigung erteilt: **chmod 775 /var/mft**.

## Vorhandensein der Dateien und Dateizugriff prüfen

Prüfen Sie die Dateien, die während der Anpassung verwendet werden sollen, mithilfe des Befehls **ls -ltr**. For example:

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

ergibt

```
drwxr-xr-x  4 SYSTASK TSOUSER   8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

wobei `drwxr-xr-x` für Folgendes steht:

**d**
: Das Verzeichnis.

**rwx**
: Der Berechtigungsinhaber *SYSTASK* verfügt für das Verzeichnis über Lese-, Schreib und Ausführungszugriff.

**r-x**
: Personen in der Gruppe *TSOUSER* verfügen über Lese- und Ausführungszugriff auf die Dateien des Verzeichnisses.

**r-x**
: Uneingeschränkter Zugriff, d. h., alle Benutzer verfügen über Lese- und Ausführungszugriff auf die Dateien des Verzeichnisses.

Prüfen Sie die folgenden Dateien:

| *Tabelle 42. Erforderlicher Benutzerzugriff auf bestimmte Dateien* | |
|---|---|
| **Pfad** | **Erforderlicher Zugriff durch Benutzer, die die Konfiguration durchführen** |
| BFG_JAVA_HOME | Lese- und Ausführungszugriff |
| /tmp | Lese- und Schreibzugriff |
| BFG_PROD | Lesen |
| BFG_DATA | Schreibzugriff |
| MQ_PATH | Lesen |

**Zugehörige Konzepte**

„Bevor Sie mit der Konfiguration von MFT für z/OS beginnen." auf Seite 785
Die Managed File Transfer-Konfiguration (MFT-Konfiguration) verwendet Dateien in z/OS UNIX System Services (z/OS UNIX) und PDSE-Datasets.

„Gängige Konfigurationen für MFT für z/OS" auf Seite 790
Ein Überblick über die verschiedenen Managed File Transfer-Konfigurationsmöglichkeiten

## z/OS Gängige Konfigurationen für MFT für z/OS

Ein Überblick über die verschiedenen Managed File Transfer-Konfigurationsmöglichkeiten

Managed File Transfer überträgt Daten mithilfe von an Warteschlangenmanager angehängten Agenten.

MFT kann mehrere Warteschlangenmanager verwenden:

- Einen oder mehrere Warteschlangenmanager zum Übertragen der Daten.
- Einen Befehlswarteschlangenmanager, der Anforderungen ausgibt. So wird beispielsweise an diesen Warteschlangenmanager eine Anforderung zum Start einer Übertragung gesendet und die zugehörigen Befehle werden an die MFT-Agenten weitergeleitet.
- Einen Koordinationswarteschlangenmanager zum Verwalten der Arbeit.

Es gibt drei gängige Konfigurationen von Managed File Transfer (MFT):

1. Ein einzelner Warteschlangenmanager mit einem oder mehreren Agenten, der lokale Verbindungen verwendet. Diese Konfiguration kann verwendet werden, um den Inhalt einer Datei in IBM MQ-Warteschlangen einzureihen.
2. Ein einzelner Warteschlangenmanager mit einem MFT-Client auf einem verteilten System, der Clientbindungen verwendet.
3. Zwei durch Kanäle verbundene Warteschlangenmanager und ein oder mehrere Agenten auf jedem System. Bei diesen Agenten kann es sich um Clientbindungen oder lokale Bindungen handeln.

Beachten Sie dabei Folgendes:

1. MFT ist in Java geschrieben; einige Shell-Scripts und JCL dienen zur MFT-Konfiguration und dem MFT-Betrieb.
2. Der Status und die Aktivität von Db2 können protokolliert und in Db2-Tabellen gespeichert werden.
3. Die Person, die MFT konfiguriert, muss mit z/OS UNIX System Services (z/OS UNIX) vertraut sein. Zum Beispiel:
   - Die Verzeichnisstruktur mit Dateien mit Namen wie `/u/userID/myfile.txt2`
   - z/OS UNIX-Befehle, wie z. B.:

     **cd** (Verzeichnis wechseln)
     **ls** (Liste)
     **chmod** (Dateiberechtigungen ändern)

**chown** (Dateieigentumsrecht oder Gruppen, die auf die Datei oder das Verzeichnis zugreifen können, ändern)

4. Folgende Produkte sind in z/OS UNIX erforderlich, um MFT konfigurieren und ausführen zu können:
   - Java; Beispiel: `/java/java71_bit64_GA/J7.1_64/`
   - IBM MQ V920 (z. B. `/mqm/V9R2M0`)
   - Db2-JDBC-Bibliotheken, wenn Db2 für Status und Protokoll genutzt werden soll (z. B. `/db2/db2v12/jdbc/lib`)

Sie benötigen einen Koordinationswarteschlangenmanager. Zur Ausführung von Agenten, der Verarbeitung von Befehlen und der Koordination kann jedoch derselbe Warteschlangenmanager verwendet werden. Bei Verwendung mehrerer Warteschlangenmanager muss einer als Koordinator festgelegt werden.

**Prüfen Sie Ihre IBM MQ-Konnektivität.**

Wenn Sie bereits über einen als MFT-Koordinator fungierenden Warteschlangenmanager verfügen, ist eine Verbindung zwischen dem Warteschlangenmanager, auf dem die Konfiguration durchgeführt wird, und den Koordinations- und Befehlswarteschlangenmanagern erforderlich.

## z/OS Kopieren Sie SCSQFCMD, um eine JCL-Bibliothek zu erstellen

Für alle Agenten und Protokollfunktionen muss eine JCL-Bibliothek erstellt werden. Die JCL enthält die Konfiguration und die Jobs, die zur Erstellung und Ausführung der Agenten bzw. Protokollfunktionen verwendet werden.

Erstellen Sie für jeden Agenten und jede Protokollfunktion eine Kopie der von IBM bereitgestellten Bibliothek SCSQFCMD, indem Sie die Teildatei BFGCOPY bearbeiten und ausführen.

Mithilfe dieser Bibliothek wird die Konfiguration für den Agenten oder die Protokollfunktion definiert. Nach der Anpassung enthält sie Jobs, die zur Erstellung der erforderlichen Managed File Transfer-Konfiguration sowie des Agenten bzw. der Protokollfunktion verwendet werden können.

Die Erstellung der Teildatei BFGCUSTM erfolgt im Rahmen dieses Prozesses.

**Anmerkung:** Wenn Sie mit z/OS UNIX-Befehlen vertraut sind, können Sie z/OS mit denselben Befehlen konfigurieren, die Sie auch auf anderen Plattformen verwenden.

**Zugehörige Konzepte**

„Gängige Konfigurationen für MFT für z/OS" auf Seite 790
Ein Überblick über die verschiedenen Managed File Transfer-Konfigurationsmöglichkeiten

„Teildatei BFGCUSTM bearbeiten" auf Seite 791
Vor der Ausführung des Jobs müssen Sie die Teildatei BFGCUSTM bearbeiten und die Parameterwerte Ihres Unternehmens eingeben.

## z/OS Teildatei BFGCUSTM bearbeiten

Vor der Ausführung des Jobs müssen Sie die Teildatei BFGCUSTM bearbeiten und die Parameterwerte Ihres Unternehmens eingeben.

Eine Liste der Parameter, die bestimmte Werte erfordern, finden Sie im Abschnitt Für Teildatei BFGCUSTM erforderliche Parameter.

Zudem müssen die folgenden Variablen überprüft und ggf. Werte angegeben werden:

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=

- connectionQMgrChannel=

Diese Eigenschaften werden häufig mit AGENT oder LOGGER verwendet.

**Anmerkung:** Host, Port und Kanal sind für die Clientverbindung erforderlich, sollten aber für eine Bindungsverbindung auf der lokalen Maschine leer gelassen werden.

Wenn es sich um den ersten Warteschlangenmanager in Ihrer Managed File Transfer-Umgebung handelt und derselbe Warteschlangenmanager für Koordination, Befehle und die Ausführung von Agenten verwendet werden soll, setzen Sie die Werte auf den Namen des lokalen Warteschlangenmanagers.

```
coordinationQMgr=MQPV
connectionQMgr=MQPV
```

MQPV ist hierbei der Name Ihres lokalen Warteschlangenmanagers.

Übergeben Sie den Job, der die PDSE aktualisiert und im angegebenen Pfad eine Verzeichnisstruktur erstellt.

Hinweis: Dieser Job erfordert exklusive Nutzung, d. h., Sie müssen die Verwendung der PSDE beenden, bis der Job ausgeführt ist.

**Tipp:** Wenn Sie den Job BFGCUSTM übergeben, werden alle JCL-Dateien ersetzt. Daher sollten Sie alle Teildateien umbenennen, die Sie ändern.

**Zugehörige Konzepte**

„Bevor Sie mit der Konfiguration von MFT für z/OS beginnen.” auf Seite 785
Die Managed File Transfer-Konfiguration (MFT-Konfiguration) verwendet Dateien in z/OS UNIX System Services (z/OS UNIX) und PDSE-Datasets.

„Agenten erstellen” auf Seite 794
Kopieren Sie die PDSE, um eine agentenspezifische PDSE wie *user.MFT.AGENT1* zu erstellen. Kopieren Sie die PDSE aus einem vorherigen Agenten oder einer vorherigen Konfiguration der Protokollfunktion, falls vorhanden. Wenn es sich um Ihre erste Konfiguration handelt, kopieren Sie die mit MFT bereitgestellte PDSE.

## z/OS Koordinationswarteschlangenmanager definieren

Managed File Transfer erfordert die Erstellung eines Warteschlangenmanagers, der als Koordinationswarteschlangenmanager fungiert.

Abhängig von der gewählten Konfiguration befindet sich dieser Warteschlangenmanager auf dem lokalen MVS-System oder einer anderen Maschine. Im ersteren Fall handelt es sich bei den Verbindungen mit dem Warteschlangenmanager um Verbindungen im Binding-Modus, im letzteren Fall um Clientverbindungen.

Nach der erfolgreichen Ausführung des Konfigurationsschritts befinden sich konfigurierte Teildateien in der PDSE.

Die Teildatei BFGCFCR definiert den Koordinationswarteschlangenmanager. Dieser Job führt die folgenden Tasks aus:

1. Erstellen einer Verzeichnisstruktur im Managed File Transfer-Verzeichnis (MFT-Verzeichnis) und Erstellen von Konfigurationsdateien.
2. Ausführen von CSQUTIL zur Definition der IBM MQ-Ressourcen.

Wenn sich der Koordinationswarteschlangenmanager auf einer fernen Maschine befindet, schlägt dieser Jobschritt fehl.

Die Teildatei BCFCFCR erstellt Dateien in z/OS UNIX System Services und MQ-Definitionen. Dieser Job führt die folgenden Tasks aus:

1. Erstellen eines MFT-Themas,
2. Erstellen einer MFT-Warteschlange

3. Ändern von *NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* zu *NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)*
4. Ausführen von *ALTER QMGR PSMODE(ENABLED)*

   Vor der Durchführung der Änderung wird der Befehl *DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)* ausgegeben. Wenn es sich bei NAMELIST nicht um Ihren Standardwert handelt, sollten Sie SYSTEM.FTE zu Ihrer Namensliste hinzufügen

Benennen Sie die Teildatei BCFCFCR unter Verwendung eines eigenen Präfixes um (Beispiel: CCPCFCR), da diese Datei bei der erneuten Anpassung ersetzt wird.

Bearbeiten Sie diese umbenannte Teildatei, indem Sie den Namen Ihrer Berechtigungsnachweisdatei einfügen. For example:

```
%BFGCMD CMD=fteSetupCoordination +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

Speichern und übergeben Sie den Job. Hinweis: Wenn Sie den Job erneut übergeben möchten, müssen Sie die Option *-f* hinzufügen.

Bei der Ausführung dieses Jobs werden die durch den Job erstellten IBM MQ-Ressourcen aufgelistet. Diese Ressourcen müssen geschützt werden.

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
```

**Zugehörige Tasks**

„Befehlswarteschlangenmanager definieren” auf Seite 793
Sie können entweder denselben Warteschlangenmanager wie die Koordinations- und Befehlswarteschlangenmanager verwenden oder einen neuen Befehlswarteschlangenmanager erstellen.

## z/OS Befehlswarteschlangenmanager definieren

Sie können entweder denselben Warteschlangenmanager wie die Koordinations- und Befehlswarteschlangenmanager verwenden oder einen neuen Befehlswarteschlangenmanager erstellen.

### Informationen zu diesem Vorgang

Sie benötigen einen Befehlswarteschlangenmanager. Allerdings kann derselbe Warteschlangenmanager als Koordinations- und Befehlswarteschlangenmanager verwendet werden. Anderenfalls muss ein neuer Befehlswarteschlangenmanager erstellt werden. Dies kann auf derselben Maschine geschehen, auf der sich auch der Koordinationswarteschlangenmanager befindet, ist jedoch nicht Voraussetzung.

### Vorgehensweise

1. Benennen Sie die Teildatei BFGCMCR unter Verwendung eines eigenen Präfixes um, z. B. CCPCMCR.

   Sie müssen BFGCMCR umbenennen, weil sie beim erneuten Anpassen der Datei ersetzt wird.
2. Bearbeiten Sie die umbenannte Teildatei, indem Sie den Namen Ihrer Berechtigungsnachweisdatei einfügen.

   For example:

```
%BFGCMD CMD=fteSetupCommands +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
```

3. Speichern und übergeben Sie den Job.

   Hinweis: Wenn Sie den Job erneut übergeben möchten, müssen Sie die Option *-f* hinzufügen.

   Dieser Warteschlangenmanager wird für Befehle wie **ftePingAgent** verwendet.
4. Prüfen Sie diese Teildatei, übergeben Sie sie und prüfen Sie die Ausgabe.

## Nächste Schritte

Informationen zur Vorgehensweise bei der Erstellung eines Agenten finden Sie im Abschnitt „Agenten erstellen” auf Seite 794.

**Zugehörige Konzepte**

„Koordinationswarteschlangenmanager definieren” auf Seite 792
Managed File Transfer erfordert die Erstellung eines Warteschlangenmanagers, der als Koordinationswarteschlangenmanager fungiert.

**Zugehörige Tasks**

MQMFTCredentials.xml konfigurieren

**Zugehörige Verweise**

MFT-Berechtigungsnachweisdateiformat

## z/OS Agenten erstellen

Kopieren Sie die PDSE, um eine agentenspezifische PDSE wie *user.MFT.AGENT1* zu erstellen. Kopieren Sie die PDSE aus einem vorherigen Agenten oder einer vorherigen Konfiguration der Protokollfunktion, falls vorhanden. Wenn es sich um Ihre erste Konfiguration handelt, kopieren Sie die mit MFT bereitgestellte PDSE.

Prüfen Sie die Teildatei BFGCUSTM und erstellen Sie bei Bedarf eine weitere Berechtigungsnachweisdatei.

Ein Großteil des Inhalts, der im Abschnitt „Teildatei BFGCUSTM bearbeiten” auf Seite 791 ausführlich beschriebenen Anpassung bleibt unverändert.

Sie müssen folgende Änderungen vornehmen:

- //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- Passen Sie LIBRARY an die PDSE des Agenten an
- SERVICE_TYPE=AGENT
- Ändern Sie NAME in den Namen des Agenten JOBCARD (in Übereinstimmung mit der PDSE)
- Ändern Sie BFG_JVM_PROPERTIES="-Xmx1024M"

Übergeben Sie diesen Job und bedenken Sie, dass für den Job exklusiver Zugriff auf das Dataset erforderlich ist.

Sämtliche Jobs für den Agenten haben Namen im Format *BFGAG**

Benennen Sie das Element *BFGAGCR* um. Dieser Job aktualisiert Dateien im Managed File Transfer-Verzeichnis und erstellt mithilfe von CSQUTIL agentenspezifische Warteschlangen im lokalen Warteschlangenmanager. Geben Sie den Namen Ihrer Berechtigungsnachweisdatei an, z. B. `-credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD)`. Wenn Sie keinen Namen angeben, verwendet der Job, der den Agenten startet, keine Berechtigungsnachweisdatei.

Überprüfen Sie die Ausgabe, um sicherzustellen, dass der Prozess erfolgreich durchgeführt wurde.

**Tipp:** Kopieren Sie den Pfadnamen der Datei *agent.properties* von der Ausgabe des Jobs in eine Teildatei in der PDSE des Agenten.

Kopieren Sie `/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties` zum Beispiel in das Mitglied AGENT.

Dies ist nützlich, wenn Sie die Merkmaldatei anzeigen und die Zeile `/u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs` hinzufügen müssen.

Hier werden Tracedateien gespeichert.

**Zugehörige Konzepte**

„Koordinationswarteschlangenmanager definieren" auf Seite 792
Managed File Transfer erfordert die Erstellung eines Warteschlangenmanagers, der als Koordinationswarteschlangenmanager fungiert.

„Agenten verwenden" auf Seite 795
So gewährleisten Sie den ordnungsgemäßen Betrieb des Agenten mithilfe verschiedener Befehle.

**Zugehörige Tasks**

„Befehlswarteschlangenmanager definieren" auf Seite 793
Sie können entweder denselben Warteschlangenmanager wie die Koordinations- und Befehlswarteschlangenmanager verwenden oder einen neuen Befehlswarteschlangenmanager erstellen.

# z/OS Agenten verwenden

So gewährleisten Sie den ordnungsgemäßen Betrieb des Agenten mithilfe verschiedener Befehle.

## Agenten starten

Benennen Sie die Teildatei BFGAGST um, überprüfen Sie sie und übergeben Sie den Job.

Wenn dies funktioniert, erhalten Sie die Nachricht BFGAG0059I: Der Agent wurde erfolgreich gestartet.

## Aktive Agenten anzeigen

Benennen Sie die Teildatei BFGAGLI um, überprüfen Sie sie und übergeben Sie den Job, der den koordinierenden Warteschlangenmanager verwendet.

Beheben Sie ggf. vorhandene Konnektivitätsprobleme

## Agenten mit Ping überprüfen, um sicherzustellen, dass er ordnungsgemäß funktioniert

Benennen Sie die Teildatei BFGAGPI um, überprüfen Sie sie und übergeben Sie den Job, der den Befehlswarteschlangenmanager verwendet.

Beheben Sie ggf. vorhandene Konnektivitätsprobleme

## Testübertragung durchführen

Weitere Informationen finden Sie im Abschnitt „Verifizierungsübertragung durchführen" auf Seite 802.

## Agenten stoppen

Benennen Sie die Teildatei BFGAGSP um, überprüfen Sie sie und übergeben Sie den Job.

Starten Sie den Agenten mit der Teildatei BFGAGST neu.

**Zugehörige Konzepte**

„Agenten erstellen" auf Seite 794
Kopieren Sie die PDSE, um eine agentenspezifische PDSE wie *user.MFT.AGENT1* zu erstellen. Kopieren Sie die PDSE aus einem vorherigen Agenten oder einer vorherigen Konfiguration der Protokollfunktion, falls vorhanden. Wenn es sich um Ihre erste Konfiguration handelt, kopieren Sie die mit MFT bereitgestellte PDSE.

## z/OS Vorhandenes Dataset für MFT-Agenten- oder -Protokollfunktions-befehle unter z/OS aktualisieren

Sie können ein aus dem Managed File Transfer-BefehlsvorlagenDataset erstelltes PDSE-Bibliotheks-Dataset für Managed File Transfer-Befehle aktualisieren.

### Vorgehensweise

1. Öffnen Sie die Teildatei mit dem JCL-Script BFGCUSTM und aktualisieren Sie die darin enthaltenen Variablen und Eigenschaften in der Datendefinitionsanweisung BFGSTDIN.

   Wenn Sie eine bereits definierte Eigenschaft entfernen möchten, löschen Sie den Eintrag nicht komplett, sondern setzen Sie den Wert einfach auf einen Leerwert. Bei der Ausführung des JCL-Scripts BFGCUSTM werden die angegebenen Eigenschaften als Update auf die z/OS UNIX System Services-Eigenschaftendateien (die eigentliche Konfiguration) des Agenten bzw. der Protokollfunktion angewendet. Script-Eigenschaften mit einem Leerwert werden in der tatsächlichen Konfiguration entfernt.
2. Übergeben Sie den Job BFGCUSTM. Dieser Job generiert den JCL-Befehlssatz passend für den Agenten bzw. die Protokollfunktion neu. Eine vollständige Liste der Befehle finden Sie im Abschnitt „JCL-Scripts für Agenten- oder Protokollfunktionsbefehle unter z/OS” auf Seite 800. Prüfen Sie im Ausgabeprotokoll des Jobs, ob das JCL-Script erfolgreich ausgeführt wurde. Falls Fehler ausgegeben wurden, korrigieren Sie diese und übergeben Sie den Job BFGCUSTM erneut.

### Ergebnisse

Sie können die generierten JCL-Scripts bearbeiten und Ihre eigene Logik hinzufügen. Achten Sie allerdings bei einer erneuten Ausführung des Jobs BFGCUSTM darauf, dass Ihre eigene Logik nicht überschrieben wird.

**Zugehörige Konzepte**

„MFT-Konfigurationsoptionen unter z/OS” auf Seite 776
Die Konfigurationsoptionen für Managed File Transfer sind unter z/OS identisch mit denjenigen für verteilte Plattformen.

**Zugehörige Tasks**

„Dataset für MFT-Agenten- oder -Protokollfunktionsbefehle erstellen” auf Seite 783
Aus dem Managed File Transfer-Befehlsvorlagen-Dataset können Sie für eine bestimmte Koordination ein PDSE-Dataset mit Befehlen für einen bestimmten Managed File Transfer Agent oder Managed File Transfer Logger erstellen.

## z/OS *z/OS-JCL-Variablen*

Sie können Substitutionswerte, JCL-Variablen und Konfigurationseigenschaften im Script BFGCUSTM verwenden.

In der folgenden Tabelle sind die Substitutionswerte für das JCL-Script BFGCUSTM eines PDSE-Bibliotheks-Datasets für MFT-Befehle zusammengefasst. Sie müssen diese Substitutionswerte durch geeignete Werte ersetzen, bevor Sie den BFGCUSTM-Job übergeben.

*Tabelle 43. Substitutionswerte für das JCL-Script BFGCUSTM*

| Substitutionsvariable | Wert |
|---|---|
| + + Bibliothek + + | Der Dataset-Name des zugehörigen PDSE-Bibliothek für MFT-Befehle. |
| + + + bfg_java_home + + | Das Installationsverzeichnis der Java-Installation. |
| ++mq_path++ | Der Pfad zum IBM MQ for z/OS UNIX System Services Components-Verzeichnis. Beispiel: `/mqm/V9R2M0`. Dieser Befehl wird verwendet, um den vollständigen Pfad zur MFT-Installation anzugeben, z. B. `/mqm/V9R2M0/mqft`. |

In der folgenden Tabelle sind die Umgebungsvariablen für die Datendefinitionsanweisung BFGSTDIN des JCL-Scripts BFGCUSTM eines PDSE-Bibliotheks-Datasets für MFT-Befehle zusammengefasst (für den Abschnitt [Variables]). Sie müssen alle Variablen ersetzen, die durch Substitutionswerte (d. a. in zwei Pluszeichen eingeschlossene Pluszeichen, + +) mit geeigneten Werten angegeben werden, bevor Sie den BFGCUSTM-Job übergeben.

*Tabelle 44. Umgebungsvariablen*

| **Umgebungsvariable** | **Wert** |
|---|---|
| SPEICHERARCHIV | Der Dataset-Name des zugehörigen PDSE-Bibliothek für MFT-Befehle. |
| TMPDIR | z/OS UNIX System Services-Verzeichnis für temporäre Dateien. |
| BFG_PROD | Der vollständige Pfad zum Verzeichnis `mqft` unter dem Verzeichnis IBM MQ for z/OS UNIX System Services Components. Beispiel: `/mqm/V9R2M0/mqft`. |
| BFG_DATA | Die Position des Managed File Transfer-Datenverzeichnisses für z/OS, d. h. der Pfad zu *DATENVERZEICHNIS*. |
| BFG_JAVA_HOME | Das Installationsverzeichnis der Java-Installation. |
| BFG_JVM_PROPERTIES | Optional. Legt einen Wert für die Umgebungsvariable BFG_JVM_PROPERTIES fest. Diese Eigenschaften werden an die Java Virtual Machine (JVM) übergeben. |

*Tabelle 44. Umgebungsvariablen (Forts.)*

| **Umgebungsvariable** | **Wert** |
|---|---|
| BFG_GROUP_NAME | In der Regel ist den Dateien und Befehlen für die Konfigurationsdaten von MFT die Dateigruppe 'mqm' zugeordnet. Daher können alle Benutzer der Gruppe 'mqm' auf die MFT-Konfiguration zugreifen und diese ändern. Weitere Informationen finden Sie unter Dateisystemberechtigungen für MFT in IBM MQ.<br><br>Unter z/OS ist eine Dateigruppe eine Dateisystementität in z/OS UNIX System Services (z/OS UNIX) und die Dateigruppe 'mqm' ist nicht in jedem Fall definiert. Sie können mithilfe der Umgebungsvariablen BFG_GROUP_NAME eine z/OS UNIX -Dateisystemgruppe für MFT -Konfigurationsdatendateien zuordnen. Geben Sie beispielsweise an der z/OS UNIX-Shelleingabeaufforderung Folgendes ein:<br><br>`export BFG_GROUP_NAME=FTEGB`<br><br>Definiert die Gruppe *FTEGB* , die allen nachfolgend erstellten Konfigurationsdateien für die aktuelle z/OS UNIX -Sitzung zugeordnet werden soll.<br><br>Sie können BFG_GROUP_NAME auf einen leeren Wert setzen, oder entfernen Sie ihn.<br><br>**Anmerkung:** Wenn BFGCUSTM zum ersten Mal ausgeführt wird und die MFT -Konfiguration von mehreren Benutzer-IDs verwendet werden soll, muss BFG_GROUP_NAME auf eine Gruppe gesetzt werden, auf die alle erforderlichen Benutzer-IDs zugreifen können. Wenn BFGCUSTM erneut ausgeführt wird, darf BFG_GROUP_NAME nicht geändert werden (andernfalls müssen die z/OS UNIX-Gruppendateiberechtigungen für alle Dateien und Verzeichnisse in dem Verzeichnis, auf das BFG_DATA verweist, geändert werden, um die neue Einstellung BFG_GROUP_NAME wiederzugeben). |
| BFG_WTO | Bei YES, ON oder TRUE ist die z/OS-Protokollierung aktiviert. Dadurch wird gesteuert, ob Nachrichten, die in das Ereignisprotokoll des Agenten geschrieben werden, auch an die Operatorprotokollfunktion von z/OS gesendet werden. Dies erleichtert den Zugriff für Automationsprodukte, wenn ein Agent über die Jobsteuersprache (JCL) ausgeführt wird. Der Routing-Code ist Programmer Information (11) und der Deskriptorcode ist Informational (12). |
| SERVICE_TYP | Legt fest, ob die MFT-Befehlsbibliothek für einen Agenten oder eine Protokollfunktion vorgesehen ist. Die gültigen Werte sind AGENT oder LOGGER. |
| NAME | Der Name des Agenten oder der Protokollfunktion für den Wert SERVICE_TYPE. |

| *Tabelle 44. Umgebungsvariablen (Forts.)* | |
|---|---|
| **Umgebungsvariable** | **Wert** |
| QMGR | Der Name des lokalen Warteschlangenmanagers, der dem Agenten oder der Protokollfunktion für den Wert SERVICE_TYPE zugeordnet ist. |
| OUTPUT_CLASS | Die Output-Klasse für SYSOUT-Dateien. Nimmt standardmäßig den Wert * an, der dieselbe Ausgabeklasse wie der Parameter MSGCLASS aus der Jobanweisung anfordert. |
| MQ_PATH | Der Pfad zum Verzeichnis der IBM MQ for z/OS UNIX-Komponente. |
| MQ_HLQ | Das übergeordnete Qualifikationsmerkmal für IBM MQ-Datasets. |
| MQ_LANG | Die Sprache, die erforderlich ist. |
| DB2_HLQ | Optional. Das übergeordnete Qualifikationsmerkmal für Db2-Datasets. |
| JOBCARD1 | Überschriftzeile 1 für einen JCL-Befehlsjob. |
| JOBCARD2 | Kopfzeile 2 für einen JCL-Befehlsjob. |
| JOBCARD3 | Kopfzeile 3 für einen JCL-Befehlsjob. |
| ADMIN_JOB1 | Überschriftzeile 1 für einen Verwaltungsjob. |
| ADMIN_JOB2 | Überschriftzeile 2 für einen Verwaltungsjob. |
| ADMIN_JOB3 | Kopfzeile 3 für einen Verwaltungsjob. |
| FTE_CONFIG | Bestehende MFT-Migrationskonfiguration. Setzen Sie diese Angabe auf einen leeren Wert, wenn die Migration nicht erforderlich ist. |
| CREDENTIAL_PATH | Pfad zur Berechtigungsnachweisdatei für die Migration, z. B. `/u/user1/agent3`. Nur für Migrationsbefehle **BFGAGMG** und **BFGLGMG** JCL-Scripts erforderlich. Setzen Sie diese Angabe auf einen leeren Wert, wenn die Migration nicht erforderlich ist. Beachten Sie auch, dass |
| DB_PROPS_PATH | Gibt die Eigenschaftendatei der Datenbankprotokollfunktion für die Migration an. Diese Option ist nur erforderlich, wenn die Merkmaldatei den folgenden Standardnamen und Pfad nicht verwendet: `config_directory/coordination_qmgr/databaselogger.properties`. Setzen Sie diese Angabe auf einen leeren Wert, wenn die Migration nicht erforderlich ist. |

In der folgenden Tabelle sind die obligatorischen Konfigurationseigenschaften von MFT für die Datendefinitionsanweisung BFGSTDIN des JCL-Scripts BFGCUSTM eines PDSE-Bibliotheks-Datasets für MFT-Befehle zusammengefasst. Sie müssen die Eigenschaften ersetzen, die durch Substitutionswerte (d. a. in zwei Pluszeichen eingeschlossene Pluszeichen, + +) mit einem geeigneten, nicht leeren Wert angegeben werden, bevor Sie den BFGCUSTM-Job übergeben. Durch diese Eigenschaften werden Überschreibungen der MFT-Konfigurationseigenschaften definiert. Sie können Agenten-und Protokollfunktionsmerkmale hinzufügen, um Agenten oder Protokollfunktionen für Ihre Umgebung anzupassen. Eine Liste aller Eigenschaften finden Sie im Abschnitt „Konfigurationseigenschaftendateien” auf Seite 808.

| *Tabelle 45. Obligatorische Konfigurationseigenschaften für die DD-Anweisung BFGSTDIN* | |
|---|---|
| **Eigenschaft** | **Wert** |
| coordinationQMgr | Der Name des Koordinations-WS-Managers für die Konfiguration, der der Agent oder die Protokollfunktion zugeordnet ist. |
| coordinationQMgrHost | Optional. Der Hostname des Systems, auf dem der Koordinations-WS-Manager ausgeführt wird. Wenn Sie den Wert für diese Eigenschaft leer lassen, wird eine Verbindung im Bindungsmodus angenommen. |
| coordinationQMgrPort | Optional. Die Nummer des Ports, an dem der Koordinationswarteschlangenmanager empfangsbereit ist. Dieser Parameter wird nur verwendet, wenn Sie auch einen Wert ohne Leerwert für die Eigenschaft "coordinationQMgrHost" angeben. |
| coordinationQMgrChannel | Optional. Kanal, der für die Verbindung zum Koordinationswarteschlangenmanager verwendet werden soll. Dieser Parameter wird nur verwendet, wenn Sie auch einen Wert ohne Leerwert für die Eigenschaft "coordinationQMgrHost" angeben. |
| connectionQMgr | Der Name des Befehlswarteschlangenmanagers für die Konfiguration, der der Agent oder die Protokollfunktion zugeordnet ist. |
| connectionQMgrHost | Optional. Der Hostname des Systems, auf dem der Befehlswarteschlangenmanager ausgeführt wird. Wenn Sie den Wert für diese Eigenschaft leer lassen, wird eine Verbindung im Bindungsmodus angenommen. |
| connectionQMgrPort | Optional. Die Nummer des Ports, an dem der Befehlswarteschlangenmanager empfangsbereit ist. Dieser Parameter wird nur verwendet, wenn Sie auch einen Wert ohne Leerwert für die Eigenschaft "connectionQMgrHost" angeben. |
| connectionQMgrChannel | Optional. Kanal, der für die Verbindung zum Befehlswarteschlangenmanager verwendet werden soll. Dieser Parameter wird nur verwendet, wenn Sie auch einen Wert ohne Leerwert für die Eigenschaft "connectionQMgrHost" angeben. |

### z/OS *JCL-Scripts für Agenten- oder Protokollfunktionsbefehle unter z/OS*

Die Gruppe verfügbarer JCL-Befehle, die in einer MFT-Befehls-PDSE-Bibliotheksdatei verfügbar sind.

| *Tabelle 46. JCL-Befehle eines PDSE-Bibliotheksdatasets für MFT-Befehle* | |
|---|---|
| **Mitglied** | **Beschreibung oder fte Befehlszeile, Befehl** |
| BFGCOPY | Job zum Erstellen einer Kopie dieser Bibliothek |
| BFGCUSTM | Job zum Anpassen dieser Bibliothek für den Agenten oder die Protokollfunktion |
| BFGZCFCR | fteSetupCoordination |

| *Tabelle 46. JCL-Befehle eines PDSE-Bibliotheksdatasets für MFT-Befehle (Forts.)* | |
|---|---|
| **Mitglied** | **Beschreibung oder fte Befehlszeile, Befehl** |
| BFGZCMCR | fteSetupCommands: die MFT-Datei 'command.properties' erstellen |
| BFGZAGCR | fteCreateAgent . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGLGCRS | fteCreateLogger . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |
| BFGZAGST | fteStartAgent . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGAGSTP | **fteStartAgent** -Prozedur. Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZAGPI | ftePingAgent . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZAGSP | fteStopAgent . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZLGST | fteStartLogger . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |
| BFGLGSTP | **fteStartLogger** -Prozedur. Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |
| BFGZLGSP | fteStopLogger . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |
| BFGZAGSH | fteShowAgentDetails. Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZLGSH | fteShowLoggerDetails. Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |
| BFGZCFDF | fteChangeDefaultConfigurationOptions |
| BFGZAGCL | fteCleanAgent . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZAGDE | fteDeleteAgent . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZLGDE | fteDeleteLogger . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |

| *Tabelle 46. JCL-Befehle eines PDSE-Bibliotheksdatasets für MFT-Befehle (Forts.)* | |
|---|---|
| **Mitglied** | **Beschreibung oder fte Befehlszeile, Befehl** |
| BFGZPRSH | fteDisplayVersion |
| BFGZAGLI | fteListAgents . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZMNLI | fteListMonitors |
| BFGZSTLI | fteListScheduledTransfers |
| BFGZTMLI | fteListTemplates |
| BFGXCROB | **fteObfuscate** Beispiel |
| BFGZRAS | **fteRAS** |
| BFGZAGTC | fteSetAgentTraceLevel . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf AGENT setzen. |
| BFGZLGTC | fteSetLoggerTraceLevel . Diese Variable wird nur erstellt, wenn Sie die Variable SERVICE_TYPE auf LOGGER setzen. |
| BFGXPRAN | **fteAnt** Beispiel |
| BFGXTRCA | **fteCancelTransfer** Beispiel |
| BFGXMNCR | **fteCreateMonitor** Beispiel |
| BFGXTMCR | **fteCreateTemplate** Beispiel |
| BFGXTRCR | **fteCreateTransfer** Beispiel |
| BFGXMNDE | **fteDeleteMonitor** Beispiel |
| BFGXSTDE | **fteDeleteScheduledTransfer** Beispiel |
| BFGXTMDE | **fteDeleteTemplate** Beispiel |

**Anmerkungen:**

- Die JCL für Befehle, die MQSC-oder Referenzlöschscripts erstellen, fordert Sie auf, ein Script auszuführen, aber das Script wurde bereits von dem Job ausgeführt.
- BFGZRAS erstellt das BFGRAS-Member, wenn der BGCUSTM-Job ausgeführt wird.

## z/OS Verifizierungsübertragung durchführen

So führen Sie eine Übertragung durch, um zu prüfen, ob das Produkt ordnungsgemäß funktioniert.

Benennen Sie die Teildatei BFGTRCRS um und bearbeiten Sie sie.

1. Fügen Sie vor `%BFGCMD CMD=fteCreateTransfer -h` ein `/*` hinzu.
2. Löschen Sie die anderen Kommentare aus der Teildatei.
3. Geben Sie den aktuellen Agentennamen für `-sa` und `-da` an.
4. Speichern Sie JCL.
5. Übergeben Sie JCL.

Diese JCL stellt eine Verbindung zum Befehlswarteschlangenmanager her.

## z/OS Protokollierungsaufgabe konfigurieren

Die Protokollierungsaufgaben müssen in demselben Image ausgeführt werden wie der Koordinationswarteschlangenmanager. Sie können die Protokollierung in Db2 vornehmen.

### Erstellen einer Protokollierungsaufgabe

Kopieren Sie die PDSE, um die logger-spezifische PDSE zu erstellen. Beispiel: `user.MFT.LOGGER`.

Wenn Sie eine andere Berechtigungsnachweisdatei verwenden müssen, erstellen Sie eine Datei. Siehe MQMFTCredentials.xml unter z/OSkonfigurieren.

Überprüfen Sie das Member BFGCUSTM . Beachten Sie, dass ein großer Teil des Inhalts von der vorherigen Anpassung gleich bleibt.

Sie müssen jedoch die folgenden Schritte ausführen:

- Ändern Sie // SYSEXEC DD DSN=SCEN.FTE.JCL ...
- Ändern Sie LIBRARY so, dass es mit dem Agenten PDSE übereinstimmt.
- Ändern Sie QMGR in den Namen des Koordinations-WS-Managers.
- SERVICE_TYPE=LOGGER erstellen
- NAME so ändern, dass er der Name der Protokollfunktion ist (entspricht der PDSE)
- Überprüfen Sie JOBCARD und ändern Sie den Jobnamen so, dass sich der Name nicht mit den Jobnamen der Agenten unterscheidet.
- Überprüfen Sie BFG_JVM_PROPERTIES="-Xmx1024M".

Bei Verwendung der Db2-Protokollfunktion ist die Erstellung einer Datei ratsam, um Db2-Traces zu erfassen und Db2-Probleme zu ermitteln.

Der Name der Datei wird in den JVM-Eigenschaften angegeben, in denen die JDBC-Trace-Eigenschaftendatei Inhalt enthält, z. B.

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

Zwei JVM-Eigenschaften festlegen

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBCA
```

`/u/.../sql.properties` ist hierbei der Name Ihrer Db2-Traceeigenschaftendatei und *DBCA* ist der Name Ihres Db2-Subsystems.

Übergeben Sie diesen Job, und stellen Sie fest, dass der Job exklusiven Zugriff auf die Datei erfordert. Die Jobs für den Agenten haben alle Namen wie *BFGLG**.

### An Dateien anmelden

Weitere Informationen zur Protokollierung in Db2 finden Sie unter „Protokollierungstask bei Protokollierung von Informationen in Db2 erstellen” auf Seite 804

Benennen Sie die Teildatei BFGLGCRS um. Dieser Job aktualisiert Dateien im Managed File Transfer-Verzeichnis (MFT-Verzeichnis) und erstellt mithilfe von CSQUTIL agentenspezifische Warteschlangen im lokalen Warteschlangenmanager.

Die Originaldatei hat den Befehl `%BFGCMD CMD=fteCreateLogger -h` , in dem die Syntax des Befehls aufgeführt ist.

Zum Erstellen der Protokollfunktionstask kommentieren Sie den `%BFGCMD CMD=fteCreateLogger -h` aus, indem Sie `/*` vor die Anweisung stellen, und stellen Sie sicher, dass die Spalte eins leer ist.

Entfernen Sie die Kommentare aus dem zweiten Befehl und konfigurieren Sie die Anweisungen. For example:

```
%BFGCMD CMD=fteCreateLogger   +
-p MQPH     +
-loggerQMgr MQPH      +
-loggerType FILE        +
-fileLoggerMode circular    +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

Überprüfen Sie die Ausgabe, um festzustellen, ob sie erfolgreich verarbeitet wurde.

**Tipp:** Kopieren Sie den Pfadnamen der `logger.properties` -Datei von der Ausgabe des Jobs in eine Teildatei in der PDSE des Agenten.

Beispiel: Kopieren in Member APATH

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

Dies ist nützlich, wenn Sie die Eigenschaftendatei anzeigen müssen.

Fügen Sie das Verzeichnis zu dieser Datei hinzu:

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

Wenn Sie sich bei der Datei anmelden, werden die Protokolldateien in diesem Verzeichnis gespeichert, z. B. `LOGGER0-20140522123654897.log`.

Tracedateien befinden sich im Protokollunterverzeichnis, z. B.

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

Sie können jetzt die Protokollierungsaufgabe starten .

## Protokollierungstask bei Protokollierung von Informationen in Db2 erstellen

Benennen Sie die Teildatei BFGLGCRS um.

Dieser Job aktualisiert Dateien im MFT-Verzeichnis und verwendet CSQUTIL, um agentenspezifische Warteschlangen im lokalen WS-Manager zu erstellen.

Sie müssen Folgendes wissen:

*Tabelle 47. Db2-Variablen*

| **Db2-Name** | **Beispiel** |
|---|---|
| *-dbName databaseName* | Sie finden diesen Namen im Standortwert der Nachricht DSNL004I für Ihr Db2-Subsystem |
| *- dbDriver filePath* | Beispiel: `/db2/db2v10/jdbc/classes/db2jcc.jar` |
| *-dbLib filePath* | Beispiel: `/db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so` |

Bearbeiten Sie die Datei. Die Originaldatei hat den Befehl `%BFGCMD CMD=fteCreateLogger -h` , in dem die Syntax des Befehls aufgeführt ist.

Entfernen Sie die Kommentare aus dem zweiten Befehl und konfigurieren Sie die Anweisungen. Beispiel:

```
%BFGCMD CMD=fteCreateLogger   +
-p MQPH     +
-loggerQMgr MQPH      +
-loggerType DATABASE      +
-dbType DB2           +
-dbName DSNDBCP         +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar  +
-dbLib /db2/db2v10/jdbc/lib/            +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

Zum Erstellen der Protokollfunktionstask kommentieren Sie den `%BFGCMD CMD=fteCreateLogger -h` aus, indem Sie `/*` vor die Anweisung stellen, und stellen Sie sicher, dass die Spalte eins leer ist.

Übergeben Sie den Job, und überprüfen Sie die Ausgabe, um festzustellen, ob er erfolgreich verarbeitet wurde.

**Tipp:** Kopieren Sie den Pfadnamen der `logger.properties` -Datei von der Ausgabe des Jobs in eine Teildatei in der PDSE der Agenten.

Kopieren Sie z. B. in die Teildatei APATH:

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

Dies ist nützlich, wenn Sie die Eigenschaftendatei anzeigen müssen.

Tracedateien befinden sich im Protokollunterverzeichnis, z. B.:

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

## Db2-Tabellen erstellen

Die Db2-Tabellen müssen erstellt werden. Die Definitionen befinden sich in der z/OS UNIX System Services-Datei `mqft/sql/ftelog_tables_zos.sql`.

Erstellen Sie die Teildatei Db2 in Ihrer PDSE. Bearbeiten Sie diese Teildatei, und verwenden Sie den Befehl COPY in der Befehlszeile. Kopieren Sie aus der z/OS UNIX System Services-Definitionsdatei.

Da sitenspezifische Anforderungen sehr unterschiedlich sein können, gibt diese Datei nur die Basisstrukturen der Tabellen und einen Tabellenbereich an, in dem sie sich befinden.

Der Tabellenbereich wird durch das SQL-Script angegeben, um sicherzustellen, dass er unter Verwendung eines Pufferpools mit einer Seitengröße erstellt wird, die ausreicht, um die größten Tabellenzeilen zu halten. Beachten Sie, dass Attribute wie z. B. LOB-Positionen usw. nicht angegeben sind.

Ihr Datenbankadministrator möchte möglicherweise eine Kopie dieser Datei ändern, um diese leistungsbezogenen Attribute zu definieren.

In dieser Datei wird auch der Standardschemaname FTELOG, der Standardtabellenbereichsname von FTELOGTS und der Datenbankname von FTELOGDB angenommen. Sie können diese Namen ändern, wenn Sie eine vorhandene Datenbank und alle lokalen Namenskonventionen anpassen müssen, indem Sie den in den Kommentaren zu Beginn der Datei beschriebenen Prozess befolgen.

**Wichtig:** Verwenden Sie Onlinefunktionen wie **SPUFI** , um die Befehle auszuführen, da die Datei Kommentare enthält und Stapelverarbeitungsprogramme wie **DSNTINAD** keine Kommentare akzeptieren.

Weitere Informationen finden Sie unter SQL mit SPUFI ausführen . Darüber hinaus enthält CSQ45STB in SCSQPROC Beispiel-JCL, die Sie anpassen können, um die Db2 SELECT-Befehle auszuführen.

## Task "logger" starten

Benennen Sie das Member BFGLGST um, überprüfen Sie es und übergeben Sie die Nachricht BFGDB0023I : Die Protokollfunktion hat die Startaktivitäten abgeschlossen und ist jetzt aktiv.

### Logger-Operationen

So zeigen Sie den Status der Protokollfunktion an: Benennen Sie die Teildatei BFGLGSH um, und übergeben Sie sie.

Um die Protokollfunktion zu stoppen, müssen Sie die Teildatei BFGLGSP umbenennen, überprüfen und übergeben.

## z/OS Umgebungsvariablen für MFT unter z/OS

Wenn Sie Befehle direkt aus der z/OS UNIX System Services-Umgebung (z/OS UNIX oder aus eigenen JCL-Scripts ausführen, müssen Sie nach der Anpassung und Konfiguration mehrere Umgebungsvariablen festlegen, bevor Sie die Konfigurations- und Verwaltungsscripts von Managed File Transfer ausführen. Sie müssen diese Variablen für jeden Benutzer und in jeder Umgebung festlegen, von der die Scripts aufgerufen werden.

Um Konflikte mit anderen Produkten zu vermeiden, können Sie ein `.wmqfterc`-Script in Ihrem Homeverzeichnis erstellen. Das Script `.wmqfterc` wird anschließend von jedem der Managed File Transfer-Scripts aufgerufen, und Sie können dieses Script verwenden, um angepasste Umgebungseinstellungen für Managed File Transfer bereitzustellen.

Es gibt auch eine optionale Umgebungsvariable (BFG_WTO), die Sie festlegen können, um Nachrichten an das Bedienerprotokoll zu senden, wenn Agenten aus JCL ausgeführt werden.

*Tabelle 48. Erforderliche z/OS-Umgebungsvariablen*

| **Umgebungsvariable** | **Wert** |
|---|---|
| BFG_JAVA_HOME | Das Installationsverzeichnis der Java-Installation. Weitere Informationen zu den unterstützten Java-Versionen finden Sie im Abschnitt Systemvoraussetzungen für IBM MQ. |
| BFG_DATA | Die Position des Datenverzeichnisses für Managed File Transfer for z/OS. Dies ist der Pfad zu *DATA_DIR*. |
| STEPLIB | Muss die folgenden IBM MQ-Datasets enthalten:<br>• SCSQAUTH<br>• SCSQANLE<br>• SCSQLADEN<br>Soll die Datenbankprotokollfunktion auf einem z/OS-System ausgeführt werden, muss STEPLIB auch noch die folgenden Db2-Datasets in der angegebenen Reihenfolge enthalten:<br>• SDSNEXIT<br>• SDSNLOD2<br>• SDSNLOAD |

Im Folgenden sehen Sie ein Beispiel für `.profile`, das die Umgebungsvariablen für Managed File Transfer ordnungsgemäß konfiguriert:

```
STEPLIB=MQM.V920.SCSQAUTH:MQM.V920.SCSQANLE:MQM.V920.SCSQLOAD
PATH=/u/fteuser/bin:/u/fteuser/J7.0/bin:/bin:/usr/bin:/u/fteuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/fteuser/J7.0
BFG_DATA=/u/fteuser/DATA_DIR
export PATH STEPLIB BFG_JAVA_HOME BFG_DATA
```

**Achtung:** Die Umgebungsvariable LIBPATH ist beim Aufruf von **`fte*`**-Befehlen aus einer z/OS UNIX-Umgebung nicht mehr erforderlich und sollte aus jedem vorhandenen `.wmqfterc`-Script entfernt werden.

Optional können Sie auch die folgenden Umgebungsvariablen festlegen:

*Tabelle 49. Optionale z/OS-Umgebungsvariablen*

| **Umgebungsvariable** | **Wert** |
|---|---|
| BFG_WTO | Mit einem der folgenden Werte wird BFG_WTO aktiviert:<br>• JA<br>• ON<br>• TRUE<br>Mit einem der folgenden Werte wird BFG_WTO inaktiviert. Bei diesen Werten wird die Groß-/Kleinschreibung nicht beachtet.<br>• NULL<br>• NEIN<br>• OFF<br>• FALSE<br>Aktiviert die Protokollierung in z/OS. Standardmäßig ist diese Umgebungsvariable inaktiviert.<br>Nachrichten, die in das Ereignisprotokoll des Agenten geschrieben werden, werden auch an die Operatorprotokollfunktion von z/OS gesendet. Dies erleichtert den Zugriff für Automationsprodukte, wenn ein Agent über die Jobsteuersprache (JCL) ausgeführt wird. Der Routing-Code ist Programmer Information (11) und der Deskriptorcode ist Informational (12). |

| *Tabelle 49. Optionale z/OS-Umgebungsvariablen (Forts.)* | |
|---|---|
| **Umgebungsvariable** | **Wert** |
| BFG_GROUP_NAME | Die Dateigruppe `mqm` ist normalerweise mit Managed File Transfer-Konfigurationsdateien und -Befehlen verknüpft. Folglich können alle Benutzer, die Mitglieder der Gruppe `mqm` sind, auf die Konfiguration von Managed File Transfer zugreifen und Änderungen vornehmen. Weitere Informationen finden Sie unter Dateisystemberechtigungen für MFT in IBM MQ.<br>Bei einem z/OS-System handelt es sich bei einer Dateigruppe um eine z/OS UNIX-Dateisystementität, und die `mqm`-Dateigruppe ist nicht unbedingt definiert. Mit der Umgebungsvariablen BFG_GROUP_NAME können Sie eine alternative, vorhandene z/OS UNIX -Dateisystemgruppe für Managed File Transfer -Konfigurationsdatendateien definieren. Geben Sie beispielsweise an der z/OS UNIX-#Shelleingabeaufforderung Folgendes ein:<br>`export BFG_GROUP_NAME=FTEGB`<br>Dieser Befehl legt fest, dass jeder nachfolgend in der aktuellen z/OS UNIX-Sitzung erstellten Konfigurationsdatei die Gruppe FTEGB zugeordnet wird.<br>Sie können BFG_GROUP_NAME auf einen leeren Wert setzen, oder entfernen Sie ihn. |

## z/OS Konfigurationseigenschaftendateien

In diesem Abschnitt finden Sie eine Übersicht über die in Managed File Transfer verwendeten Eigenschaften.

- Die MFT-Datei 'coordination.properties'
- Die MFT-Datei 'command.properties'
- Die MFT-Datei 'agent.properties'
- Logger-Konfigurationseigenschaftendatei

## z/OS MFT für z/OS Automatic Restart Manager (ARM) konfigurieren

Managed File Transfer ist eine ARM-fähige Anwendung.

### Vorbereitende Schritte

Weitere Informationen zum Aktivieren von ARM und die Definition von ARM-Richtlinien für Ihr System finden Sie im Abschnitt z/OS Automatic Restart Manager (ARM) verwenden.

Wenn Sie die MFT-Datenbankprotokollfunktion zum automatischen Neustart und zum Wiederherstellen einer Verbindung zu einer Db2-Datenbank verwenden möchten, ist ARM der einzige unterstützte Neustartmanager, der verfügbar ist.

## Informationen zu diesem Vorgang

Mit ARM können Agenten und Protokollfunktionen für den Neustart konfiguriert werden, indem die Agenten-/Protokollfunktionseigenschaften armELEMTYPE und armELEMENT festgelegt werden. Die Eigenschaft armELEMTYPE definiert den Typ des ARM-Elements und die Eigenschaft armELEMENT ist der Name des Elements, das ARM registrieren soll:

- Sie können den Agenten ELEMTYPE auf SYSBFGAG setzen, und armELEMENT kann so eingestellt werden, dass sie dem Agentennamen entspricht.
- Sie können die Protokollfunktion ELEMTYPE auf SYSBFGLG setzen, und armELEMENT kann so eingestellt werden, dass sie dem Protokollfunktionsnamen entspricht.

**Anmerkung:** Agenten und Protokollfunktionen, die für den Neustart durch ARM konfiguriert sind, können nur erfolgreich von einem Stapeljob oder einer gestarteten Task ausgeführt werden. Der Versuch, den Agenten oder die Protokollfunktion direkt über die z/OS UNIX System Services-Befehlszeile zu starten, schlägt mit einem ARM-Fehlerursachencode fehl.

### Beispiel

Das folgende Beispiel für eine Neustartrichtlinie definiert den Agenten BFGFT7CAG1 als abhängig vom WS-Manager FT7C:

```
RESTART_ORDER
   LEVEL(3)
   ELEMENT_TYPE(SYSBFGAG,SYSBFGLG)

RESTART_GROUP(GROUP7C)
   ELEMENT(SYSMQMGRFT7C)
   ELEMENT(BFGFT7CAG1)
       RESTART_ATTEMPTS(3,300)
```

# z/OS Beispiel: JCL für Managed File Transfer-Agenten unter z/OS erstellen

Diese Informationen helfen Ihnen dabei, die JCL zu generieren, mit der ein Agent unter IBM MQ for z/OS erstellt und gestartet werden kann.

## Beispielbibliothek Kopieren

Gehen Sie wie folgt vor:

1. Erstellen Sie eine Kopie der Bibliothek SCSQFCMD (siehe „Kopieren Sie SCSQFCMD, um eine JCL-Bibliothek zu erstellen” auf Seite 791), indem Sie die Bibliothek öffnen.

   Bei den meisten Einträgen (die mit BFGX, BFGY oder BFGZ beginnen) handelt es sich um Vorlagen, mit denen Sie später die angepasste JCL für den Agenten generieren.

   Der wichtige Eintrag ist BFGCOPY.
2. Öffnen Sie BFGCOPY und ersetzen Sie die folgenden Zeichenfolgen:

   **++supplied_library++**
   durch den Namen der Bibliothek SCSQFCMD, die als Teil des Produkts installiert wurde.

   **++service-library++**
   durch den Namen der Bibliothek, die für den Agenten verwendet werden soll (Zielbibliothek).
3. Übergeben Sie den Job. Sie haben nun eine neue Bibliothek, die Sie verwenden können.

## BFGCUSTM bearbeiten

Gehen Sie wie folgt vor:

1. Öffnen Sie die neue Bibliothek, damit Sie den Eintrag BFGCUSTM bearbeiten können (siehe „Teildatei BFGCUSTM bearbeiten” auf Seite 791)
2. Ändern Sie alle Parameter im Eintrag, die in ++ eingeschlossen sind, und ersetzen Sie diese durch die entsprechenden Werte. Ändern Sie beispielsweise die folgenden Zeichenfolgen:

   **++mq_path++**
   Der Pfad zum Verzeichnis der z/OS UNIX System Services-Komponente (z/OS UNIX). Beispiel: `/mqm/V9R2M0`.

   **Anmerkung:** Es gibt drei Instanzen dieser Variablen, die ersetzt werden sollten.

   **++bfg_data++**
   für den Verweis auf das z/OS UNIX-Verzeichnis, in dem Ihre IBM MQ Managed File Transfer for z/OS-Konfiguration gespeichert werden soll.

   **++service_type++**
   in das Wort AGENT

   **++agent_name++**
   in den Namen Ihres Agenten

**Anmerkungen:**

1. Einige der Einträge wie beispielsweise `++options++`, die für CLEAN_AGENT_PROPS erforderlich sind, werden nicht benötigt und sollten entfernt werden.
2. Im Abschnitt „Bevor Sie mit der Konfiguration von MFT für z/OS beginnen.” auf Seite 785 finden Sie eine vollständige Liste aller Parameter im Eintrag BFGCUSTM sowie eine Beschreibung der zugehörigen Werte.

## BFGCUSTM-JCL übergeben

Gehen Sie wie folgt vor:

1. Übergeben Sie den Job.
2. Verlassen Sie die Bibliothek in ISPF.

   Dies ist erforderlich, da die Bibliothek durch den Job BFGCUSTM aktualisiert wird, wozu die Bibliothek aber geschlossen sein muss.
3. Lesen Sie nach Abschluss des Jobs das Jobprotokoll.

   In einer Reihe von Nachrichten wird dort angezeigt, dass neue Einträge in der Bibliothek erstellt wurden.

   Jeder Eintrag enthält JCL, mit der bestimmte Tasks für Ihren Agenten ausgeführt werden können. Im Abschnitt „JCL-Scripts für Agenten- oder Protokollfunktionsbefehle unter z/OS” auf Seite 800 finden Sie eine Liste dieser Einträge sowie die IBM MQ Managed File Transfer-Befehle, denen sie entsprechen.

## BFGAGCR zum Erstellen des Agenten übergeben

Der neue Eintrag BFGAGCR enthält einige JCL zum Erstellen eines Agenten, indem der Befehl **`fteCreateAgent`** aufgerufen wird.

Gehen Sie wie folgt vor:

1. Öffnen Sie den Eintrag BFGAGCR.

   Es sollte angezeigt werden, dass BFGAGCR mit den Namen Ihrer folgenden Komponenten gefüllt wurde:

   - Agent
   - Agenten- WS- Manager
   - Koordinationswarteschlangenmanager für die MFT-Topologie

2. Übergeben Sie den Eintrag BFGAGCR.

   Bei der Ausführung des Eintrags werden folgende Aktionen vorgenommen:

   - Die erforderlichen Konfigurationsdateien für Ihren Agenten werden erstellt.
   - Die Verbindung zum Warteschlangenmanager des Agenten wird hergestellt und die für den Agenten erforderlichen Systemwarteschlangen werden mithilfe von CSQUTIL erstellt.
   - Der Agent wird beim Koordinationswarteschlangenmanager registriert.

### Agent durch Übergabe von BFGAGST starten

Gehen Sie wie folgt vor:

1. Übergeben Sie den Eintrag BFGAGST. Im Abschnitt Agent verwenden finden Sie verschiedene Befehle, mit denen gezeigt wird, dass der Agent ordnungsgemäß funktioniert.
2. Stellen Sie nach dem Abschluss des Jobs sicher, dass das Jobprotokoll die folgenden Nachrichten enthält:

   ```
   BFGAG0058I: The agent has successfully initialized.
   BFGAG0059I: The agent has been successfully started.
   ```

   Dies bedeutet, dass Ihr Agent betriebsbereit ist und verwaltete Übertragungen ausführen kann.

## z/OS MFT-Agenten in eine neue z/OS-LPAR verschieben

Manchmal muss ein IBM MQ Managed File Transfer for z/OS-Agent in eine andere LPAR verschoben werden, während der Agent in derselben IBM MQ Managed File Transfer-Topologie mit denselben Koordinations- und Befehlswarteschlangenmanagern verbleibt. Welche Schritte hierfür erforderlich sind, hängt davon ab, wie der zu migrierende Agent ursprünglich erstellt wurde.

### Informationen zu diesem Vorgang

Verschieben Sie Ihren IBM MQ Managed File Transfer for z/OS-Agenten auf eine der folgenden Arten:

- Wenn der Agent ursprünglich mit einer angepassten Version der Bibliothek SCSQFCMD erstellt wurde, verwenden Sie die Bibliothek, um sie in einer neuen LPAR erneut zu erstellen.
- Wenn der Agent ursprünglich durch Ausführung von z/OS UNIX System Services-Befehlen (z/OS UNIX erstellt wurde, verwenden Sie die Befehle, um ihn in einer neuen LPAR erneut zu erstellen.

**Anmerkung:**

Geplante Übertragungen und Übertragungsvorlagen werden auf dem Koordinationswarteschlangenmanager für eine IBM MQ Managed File Transfer-Topologie gespeichert. In dieser Task wird davon ausgegangen, dass der Koordinationswarteschlangenmanager nicht Teil der Verschiebung ist. In diesem Fall verbleiben alle geplanten Übertragungen und Übertragungsvorlagen, die dem zu verschiebenden Agenten zugeordnet sind, nach der Verschiebung auf dem vorhandenen Koordinationswarteschlangenmanager.

### Prozedur

- Versetzen Sie einen Agenten, der mit einer angepassten Version der Bibliothek SCSQFCMD erstellt wurde.

  Wenn der Agent mit einer angepassten Version der Bibliothek SCSQFCMD erstellt wurde, können Sie mit dieser Bibliothek die IBM MQ Managed File Transfer for z/OS -Umgebung und die Agentenkonfiguration in der neuen LPAR erneut erstellen. Führen Sie hierzu die folgenden Schritte aus:

  1. Kopieren Sie die angepasste Version der Bibliothek aus der ursprünglichen LPAR in die neue LPAR.
  2. Bearbeiten Sie das Member BFGCUSTM in der angepassten Version der Bibliothek in der neuen LPAR und stellen Sie sicher, dass die Parameterwerte noch gültig sind.

3. Führen Sie das Member BFGCUSTM in der neuen LPAR aus, um die gesamte JCL zu erstellen, die zum Konfigurieren der Umgebung und zum Erstellen des Agenten erforderlich ist.
4. Führen Sie das Member BFGCFCR aus, um den Koordinationswarteschlangenmanager zu definieren, der vom Agenten in der neuen LPAR verwendet werden soll, und erstellen Sie die zum Speichern der IBM MQ Managed File Transfer-Konfiguration erforderliche Verzeichnisstruktur.
5. Führen Sie als Nächstes das Member BFGCMCR aus, um den Befehlswarteschlangenmanager zu definieren, der vom Agenten in der neuen LPAR verwendet werden soll.
6. Führen Sie das Member BFGAGCR aus, um den Agenten und seine Konfiguration erneut zu erstellen.
7. Stellen Sie sicher, dass die vom Agenten verwendeten Systemwarteschlangen auf dem Warteschlangenmanager für diesen Agenten vorhanden sind.

Wenn dem zu verschiebenden Agenten Ressourcenüberwachungen zugeordnet sind, müssen Sie die Überwachungen auf dem neuen Agenten erneut erstellen. Führen Sie hierzu die folgenden Schritte aus:

1. Führen Sie in der ursprünglichen LPAR das Member BFGMNLI aus, um die Definitionen für die dem ursprünglichen Agenten zugeordnete Ressourcenüberwachung in XML-Dateien zu exportieren.
2. Kopieren Sie die XML-Dateien mit den Ressourcenüberwachungsdefinitionen in die neue LPAR.
3. Verwenden Sie das Member BFGMNCRS in der Bibliothek SCSQFCMD der neuen LPAR, um die in den XML-Dateien gespeicherten Ressourcenmonitordefinitionen zu importieren. Auf diese Weise werden die Überwachungen auf dem neuen Agenten erstellt.

- Agenten verschieben, der durch die Ausführung von Befehlen in z/OS UNIX erstellt wurde

  Wenn der Agent ursprünglich durch Ausführung von z/OS UNIX-Befehlen erstellt wurde, können Sie Befehle verwenden, um den Agenten in einer neuen LPAR erneut zu erstellen. Führen Sie hierzu die folgenden Schritte aus:

  1. Führen Sie den Befehl fteSetupCoordination in der neuen LPAR aus, um den Koordinationswarteschlangenmanager zu definieren, der von dem Agenten verwendet werden soll, und erstellen Sie die zum Speichern der IBM MQ Managed File Transfer-Konfiguration benötigte Verzeichnisstruktur.
  2. Führen Sie den Befehl fteSetupCommands aus, um den Befehlswarteschlangenmanager zu definieren, der vom Agenten in der neuen LPAR verwendet werden soll.
  3. Führen Sie den Befehl fteCreateAgent aus, um den Agenten und seine Konfiguration erneut zu erstellen.
  4. Stellen Sie sicher, dass die vom Agenten verwendeten Systemwarteschlangen auf dem Warteschlangenmanager für diesen Agenten vorhanden sind.

  Wenn dem zu verschiebenden Agenten Ressourcenüberwachungen zugeordnet sind, müssen Sie die Überwachungen auf dem neuen Agenten erneut erstellen. Führen Sie hierzu die folgenden Schritte aus:

  1. Führen Sie in der ursprünglichen LPAR den Befehl fteListMonitors unter Angabe des Parameters **-ox** aus, um die Definitionen für die dem ursprünglichen Agenten zugeordnete Ressourcenüberwachung in XML-Dateien zu exportieren.
  2. Kopieren Sie die XML-Dateien mit den Ressourcenüberwachungsdefinitionen in die neue LPAR.
  3. Führen Sie den Befehl fteCreateMonitor in der neuen LPAR aus und geben Sie dabei den Parameter **-ix** an, um die Ressourcenüberwachungsdefinitionen zu importieren, die in den XML-Dateien gespeichert sind. Auf diese Weise werden die Überwachungen auf dem neuen Agenten erstellt.

## z/OS V 9.3.0 MFT-Infrastruktur mit IBM MQ for z/OS-Gruppen mit gemeinsamer Warteschlange planen

Bei Verwendung von IBM MQ Managed File Transfer (MFT) ist Folgendes zu beachten, wenn einer oder mehrere der Agenten-, Befehls- oder Koordinationswarteschlangenmanager zu einer IBM MQ for z/OS-Gruppe mit gemeinsamer Warteschlange gehören.

Eine Beschreibung der Agenten, der Befehlswarteschlangenmanager und der Koordinationswarteschlangenmanager finden Sie in der MFT-Topologieübersicht.

## Agentenwarteschlangenmanager

Normalerweise stellt ein MFT-Agent eine Verbindung zu einem einzelnen Agentenwarteschlangenmanager her und verwendet lokale Warteschlangen, auf die nur dieser Warteschlangenmanager zugreifen kann. Dem Agenten wird mitgeteilt, mit welchem Warteschlangenmanager er eine Verbindung herstellen soll, indem ihm bei seiner ersten Erstellung der Name des Warteschlangenmanagers übergeben wird.

Mit IBM MQ for z/OS ist es möglich, den Agenten zu erstellen und den Namen des Warteschlangenmanagers durch den Namen einer Gruppe mit gemeinsamer Warteschlange (QSG) zu ersetzen. Dies bedeutet, dass der Agent eine Verbindung zu jedem verfügbaren Warteschlangenmanager in der QSG herstellen kann, um Dateiübertragungen auszuführen. Tritt bei dem Warteschlangenmanager, mit dem der Agent aktuell verbunden ist, ein Fehler auf, erkennt der Agent den Fehler und stellt eine Verbindung zu einem alternativen Warteschlangenmanager in der QSG her.

Die Verbindung eines Agenten mit einer Gruppe mit gemeinsamer Warteschlange in Kombination mit der hoch verfügbaren Agentenunterstützung, die von IBM MQ 9.2.0bereitgestellt wird (siehe „Hochverfügbarkeitsagenten in Managed File Transfer" auf Seite 837), ermöglicht die Erstellung sehr leistungsfähiger MFT -Topologien.

In der folgenden Abbildung wurde beispielsweise *Agent1* so erstellt, dass sein Warteschlangenmanager eine QSG ist, die aus den zwei Warteschlangenmanagern *QM1* und *QM2* besteht. Die Agentenwarteschlangen wurden als gemeinsam genutzte Warteschlangen definiert, die in der Coupling-Facility gespeichert sind.

Dies bedeutet, dass der Agent entweder in *LPAR 1* oder *LPAR 2* ausgeführt werden kann und eine Verbindung zu *QM1* oder *QM2* herstellen kann. Die Dateien und Datasets, aus denen der Agent Daten liest oder in die er Daten schreibt, werden gemeinsam genutzt, d. h., es kann aus jeder LPAR darauf zugegriffen werden.

Darüber hinaus wurde der Agent als hoch verfügbarer Agent konfiguriert. Im Diagramm ist der Agent in *LPAR 1* aktiv und eine Standby-Instanz des Agenten wird in *LPAR 2* ausgeführt.

Diese Topologie bietet eine hohe Ausfallsicherheit. Sollte der Agent, der in *LPAR 1* aktiv ist, oder Warteschlangenmanager *QM1* oder *LPAR 1* ausfallen, kann die Standby-Instanz des Agenten in *LPAR 2* übernehmen und die Verarbeitung von Dateiübertragungen ab dem Zeitpunkt des Ausfalls fortsetzen.

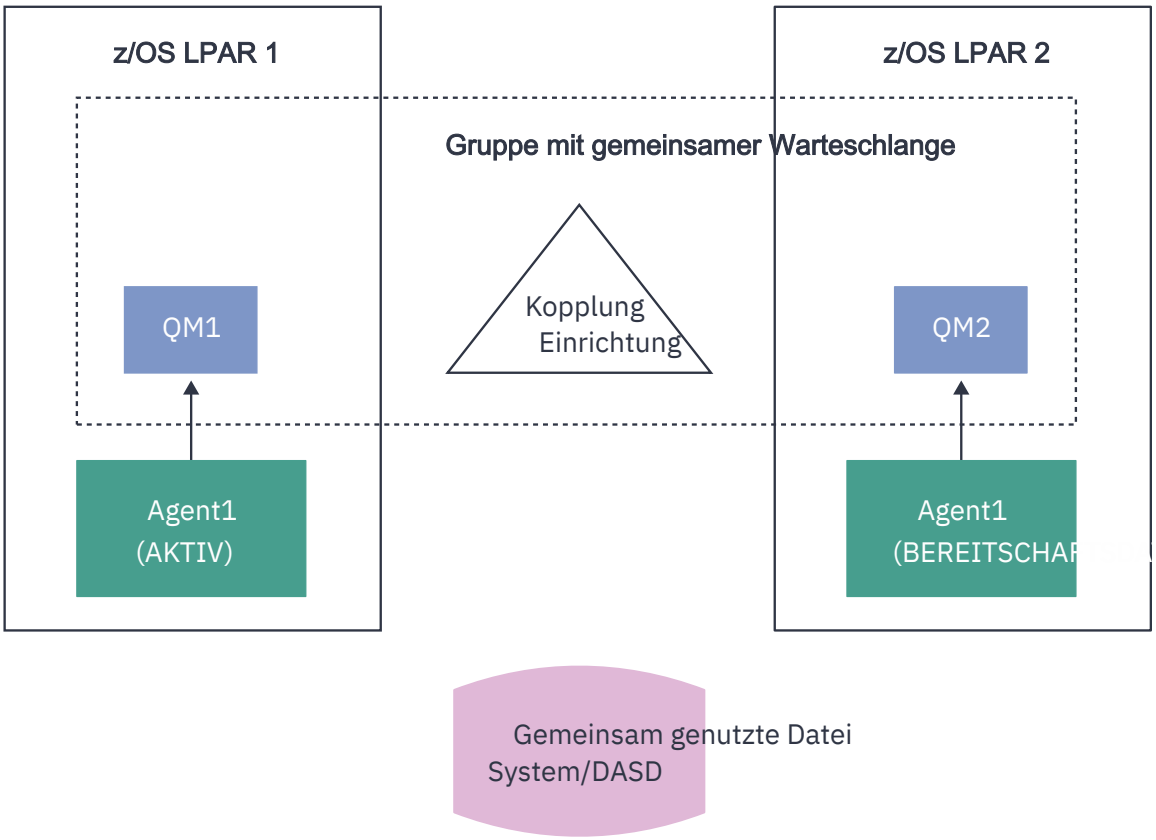



*Abbildung 95. Hoch verfügbarer MFT-Agent unter Verwendung einer Gruppe mit gemeinsamer Warteschlange*

## Agenten erstellen, der eine QSG als Agentenwarteschlangenmanager verwendet

Ein Agent wird mit dem Befehl fteCreateAgent erstellt. Bei der Erstellung wird der Name der Gruppe mit gemeinsamer Warteschlange für den Agentenwarteschlangenmanager angegeben. For example:

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1
```

Dieser Befehl erstellt einen Agenten mit dem Namen *Agent1*, der einen beliebigen Warteschlangenmanager, der Mitglied der Gruppe *QSG1* ist, als seinen Agentenwarteschlangenmanager verwendet. In dieser Konfiguration stellt der Agent eine Verbindung zum Agentenwarteschlangenmanager über eine Cross-Memory-Verbindung (Bindungsmodus) her, was bedeutet, dass sich der Agent und der Warteschlangenmanager in derselben LPAR befinden müssen. Dies entspricht genau dem Beispiel, das oben in Abbildung 1 gezeigt wird.

Wenn Sie den Befehl **fteCreateAgent** ausführen, generiert er eine Gruppe von MQSC-Befehlen zum Erstellen der notwendigen Warteschlangen auf dem Agentenwarteschlangenmanager.

Wenn der Agentenwarteschlangenmanager eine QSG ist, muss diese Befehlsgruppe so geändert werden, dass jede Warteschlange als eine gemeinsam genutzte Warteschlange erstellt wird. Das heißt, dass jede Warteschlange mit QSGDISP (SHARED) und einer entsprechenden Coupling-Facility-Struktur erstellt werden muss, die vom Attribut CFSTRUCT bereitgestellt wird.

Das folgende Beispiel zeigt, wie der MQSC-Befehl geändert werden muss, damit die Warteschlange SYSTEM.FTE.COMMAND.AGENT1 als gemeinsam genutzte Warteschlange erstellt wird. Die Änderungen der Standardwerte sind in Fettdruck dargestellt.

**Wichtig:** Sie müssen ähnliche Änderungen für alle anderen Warteschlangen vornehmen, die der Agent verwendet.

```
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.AGENT1) +
 QSGDISP(SHARED) +
 CFSTRUCT(MFTSTRUCT) +
 DEFPRTY(0) +
 DEFSOPT(SHARED) +
 GET(ENABLED) +
 INDXTYPE(CORRELID) +
 MAXDEPTH(5000) +
 MAXMSGL(4194304) +
 MSGDLVSQ(PRIORITY) +
 PUT(ENABLED) +
 RETINTVL(999999999) +
 SHARE +
 NOTRIGGER +
 USAGE(NORMAL) +
 REPLACE
```

## Agenten erstellen, der eine QSG als Agentenwarteschlangenmanager verwendet und eine Verbindung als Client herstellt

Agenten können über einen Clientkanal eine Verbindung zu ihrem Agentenwarteschlangenmanager herstellen. Sie können diesen Ansatz nutzen, um dem Agenten die Ausführung auf verteilten Plattformen zu ermöglichen, während eine Verbindung zu einer QSG hergestellt wird. Wenn alle Warteschlangenmanager in der QSG für IBM MQ Advanced for z/OS Value Unit Edition lizenziert sind, kann der Agent auch aus einer z/OS-LPAR, in der es keinen lokalen Warteschlangenmanager gibt, eine Verbindung zu ihnen herstellen.

Diese Topologie wird in der folgenden Abbildung gezeigt und ermöglicht es dem Agenten, die Ausfallsicherheit von QSGs zu nutzen. Wenn der Warteschlangenmanager in der QSG, mit der der Agent aktuell verbunden ist, ausfällt, stellt der Agent automatisch eine Verbindung zu einem anderen Mitglied der QSG her und setzt die Verarbeitung fort.

Über den Sysplex Distributor werden die Verbindungen vom Agenten auf die verfügbaren Warteschlangenmanager in der QSG verteilt.

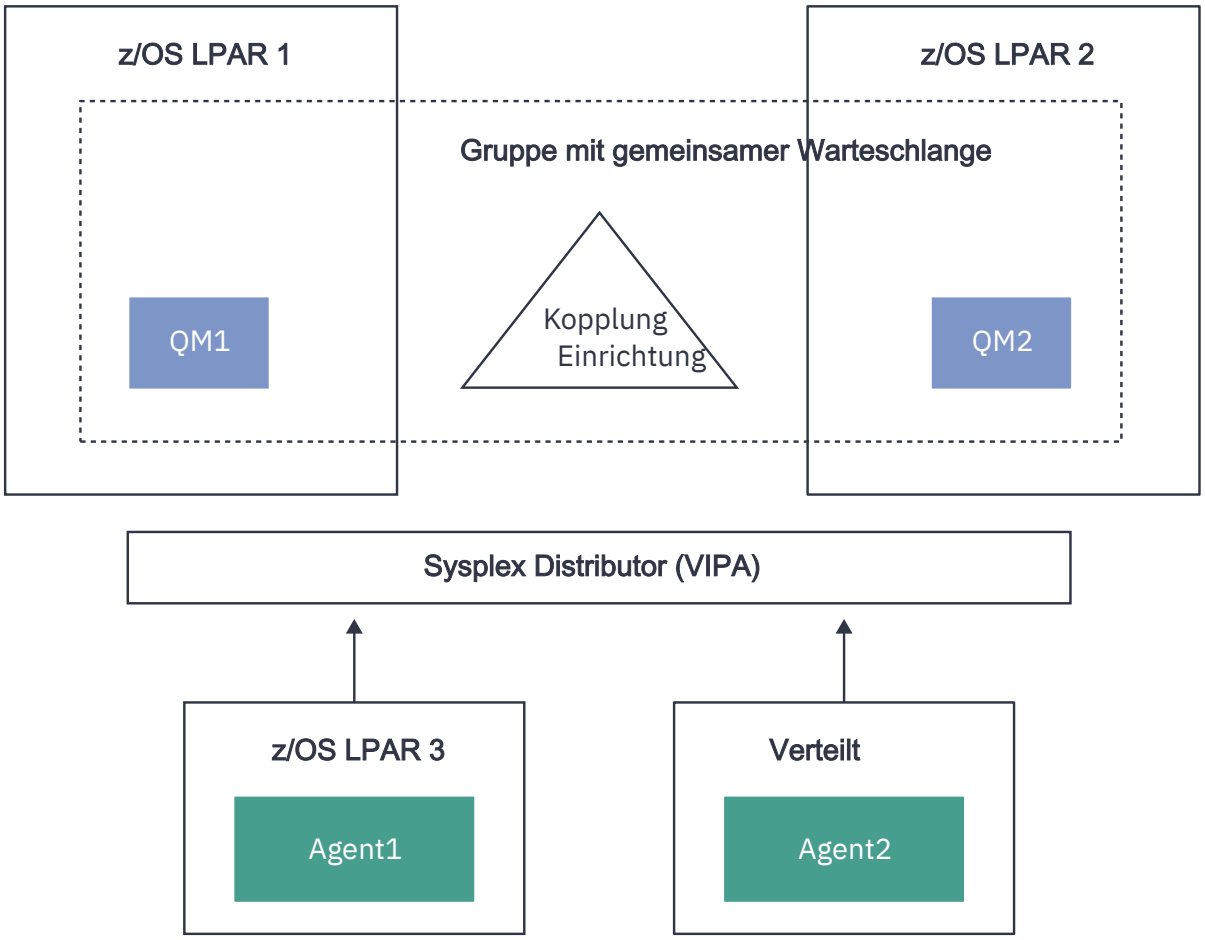



*Abbildung 96. MFT-Agenten, die als Client eine Verbindung zu einer Gruppe mit gemeinsamer Warteschlange herstellen*

Um diese Topologie verwenden zu können, muss jeder der Warteschlangenmanager in der QSG über einen Serververbindungskanal verfügen, der für die Verwendung durch den Agenten definiert ist. Informationen zur Vorgehensweise finden Sie im Abschnitt „Client mit einer Gruppe mit gemeinsamer Warteschlange verbinden” auf Seite 66.

Bei der Erstellung des Agenten müssen die Warteschlangenmanager so konfiguriert werden, dass sie den Kanal, der für die QSG definiert ist, verwenden und über den Sysplex Distributor darauf zugreifen können. For example:

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1 -agentQMgrHost vipaAddress
-agentQMgrPort sharedPort -agentQMgrChannel CHANNEL1
```

Wie bereits erwähnt, müssen die mit dem Befehl **fteCreateAgent** generierten MQSC-Befehle angepasst werden, indem QSGDISP (SHARED) und eine entsprechende Coupling-Facility-Struktur im Attribut CFSTRUCT angegeben werden.

## Befehlswarteschlangenmanager

Der MFT-Befehlswarteschlangenmanager kann Mitglied einer QSG sein. Allerdings kann bei der Angabe eines Befehlswarteschlangenmanagers nicht der Name einer QSG verwendet werden. Sie müssen einen spezifischen Warteschlangenmanagernamen verwenden.

### Koordinationswarteschlangenmanager

Der MFT-Koordinationswarteschlangenmanager kann Mitglied einer QSG sein. Allerdings kann, wie bei einem Befehlswarteschlangenmanager, bei der Angabe eines Koordinationswarteschlangenmanager nicht der Name einer QSG verwendet werden. Sie müssen einen spezifischen Warteschlangenmanagernamen verwenden.

### Befehle zum Herstellen einer Verbindung zu einer QSG

MFT stellt eine Reihe von Befehlen für verwaltete Agenten, Übertragungen und Agenten-, Befehls- oder Koordinationswarteschlangenmanager bereit. Sie können die Befehle, die eine Verbindung zu einem Agentenwarteschlangenmanager herstellen, nur verwenden, wenn sich der Warteschlangenmanager in einer QSG befindet.

Es folgt eine Liste der Befehle, die eine Verbindung zum Agentenwarteschlangenmanager herstellen:

- **fteCleanAgent**
- **fteCreateAgent**
- **fteCreateBridgeAgent**
- **fteCreateCDAgent**
- **fteDeleteAgent**

Beachten Sie, dass Sie den Namen des Warteschlangenmanagers angeben müssen, wenn Sie andere MFT-Befehle ausführen.

## Managed File Transfer for z/OS mit dem JZOS- Java -Startprogramm verwenden

Sie können die Anweisungen in diesem Abschnitt als alternative Methode für die Verwendung von Managed File Transfer in Ihrem Unternehmen auf Ihrem IBM MQ for z/OS-System anwenden.

### Übersicht

Für Managed File Transfer for z/OS (MFT) wird die z/OS-Standardinstallation durchgeführt. Eine alternative Methode zur Ausführung von MFT -Befehlen ist die Verwendung von JCL und JZOS Java Launcher.

Weitere Informationen finden Sie unter JZOS Batch Launcher and Toolkit.

Wenn Ihre JCL nicht ordnungsgemäß verarbeitet werden kann, lesen Sie die Informationen im Abschnitt Allgemeine MFT-Probleme mit JZOS .

### Beispiel-JCL

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL)   (1)
// INCLUDE MEMBER=BFGJCL8       (2)
//     DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent  (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}"  (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT  (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

Dabei gilt:

- (1) Ist die Position der eingeschlossenen JCL-Anweisungen.
- (2) Anschließen des angegebenen JCL-Members von der Position in 1)
- (2A) Dies erweitert den // STDENV-siehe unten.

- (3) Dies ist der Befehl, der ausgeführt werden soll, ohne das führende fte Präfix
- (4) Diese Zeilen sind erforderlich. Sie konfigurieren Informationen für JZOS.
- (5) Die Parameter für den Befehl
- Das Member BFGJCL8 (Sie können Ihren eigenen Namen auswählen) ruft JZOS auf. Dieses Member verfügt über die STEPLIB und weitere JCL, die für die Ausführung von MFT erforderlich sind.

### Andere JCL, die Sie einschließen müssen

Sie sollten die JCL für die IBM MQ for z/OS-Bibliotheken und (bei Verwendung der Db2-Protokollfunktion) für die Db2-Bibliotheken einschließen.

For example:

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T'  (1)
//STEPLIB  DD DSN=SYS1.SIEALNKE,DISP=SHR        (2)
//* MQ libraries
//   DD DSN=MQM.V920.SCSQAUTH,DISP=SHR MQ Bindings
//   DD DSN=MQM.V920.SCSQANLE,DISP=SHR MQ Bindings
//   DD DSN=MQM.V920.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNEXIT.DBCP
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD
//   DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOD2
//SYSOUT  DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT  DD SYSOUT=H
//STDERR  DD SYSOUT=H

//STDENV  DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR  (3)
```

Dabei gilt:

- (1) Dies ist der Name des JZOS-Programms. Suchen Sie in SYS1.SIEALNKE nach der Version auf Ihrem System. Fügen Sie PARM = '+ T' hinzu, um zusätzliche Diagnoseprogramme zu erhalten.
- (2) Dies ist der Datensatz, der mit dem JZOS-Programm definiert ist.
- (3) Dies ist der Membername eines Shell-Scripts. In diesem Script sind die für MFT erforderlichen Parameter definiert Siehe „Shell-Script zum Definieren von MFT” auf Seite 818.

  Es kann sich um eine beliebige Datei und ein beliebes Member handeln. Sie muss die letzte Datei in der Datei sein, da der JCL-Job dies erweitert. Siehe 2A in „Beispiel-JCL” auf Seite 817.

### Shell-Script zum Definieren von MFT

Im unter „Andere JCL, die Sie einschließen müssen” auf Seite 818 aufgeführten Beispiel wird das Member BFGZENV8 verwendet. Dies basiert auf dem JZOS-Profil.

Sie müssen Folgendes wissen:

- Das Verzeichnis, in dem Java installiert ist.
- Das Verzeichnis der IBM MQ for z/OS Java-Bibliotheken und der MFT-Bibliotheken.
- Eine Benutzer-ID muss zu einer bestimmten Gruppe gehören, damit sie als IBM MQ for z/OS-Administrator verwendet werden kann. Sie benötigen den Namen dieser Gruppe
- Wenn Sie nicht Englisch für die Nachrichten verwenden, müssen Sie wissen, welche Sprache angegeben werden muss.

### Beispieldatei

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
```

```
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH:"/mqm/V9R2M0/java/lib/"
export LIBPATH

export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/mqm/V9R2M0/mqft"
export BFG_CONFIG="/u/johndoe/fteconfig"
# export BFG_LANG=" -Duser.language=de "
export BFG_LANG=" "

____________________
```

Dabei gilt:

**export _BPXK_AUTOCVT = "ON "**
Ist für Unicode-Konvertierung erforderlich

**export JAVA_HOME=" /java/java71_bit64/J7.1_64/"**
Die Position des Verzeichnisses Java . Geben Sie den Namen des Pfads für Javaan. Dieses Verzeichnis enthält bin und andere Verzeichnisse.

**export PATH= "/bin: ${ JAVA_HOME } /bin/classic/"**
Definiert die Pfadanweisung für ausführbare Java -Anweisungen

**LIBPATH=" /lib:/usr/lib: ${ JAVA_HOME } /bin "**
Richtet den Bibliothekspfad für die ausführbaren Java -Anweisungen ein.

**LIBPATH=" $LIBPATH: ${ JAVA_HOME } /bin/classic "**
Fügt der Anweisung LIBPATH weitere Java -Bibliotheken hinzu

**LIBPATH=$LIBPATH:"/mqm/V9R2M0/java/lib/"**
Fügt dem Bibliothekspfad IBM MQ for z/OS-Bibliotheken hinzu. Geben Sie den Namen Ihrer IBM MQ for z/OS-Bibliotheken in z/OS UNIX System Services an.

**export LIBPATH**
Stellt den LIBPATH für JZOS zur Verfügung.

**export BFG_JAVA_HOME = "${ JAVA_HOME }"**
Setzt den Wert von 'BFG_JAVA_HOME' auf den oben angegebenen Wert von JAVA_HOME.

**export BFG_WTO = "YES "**
Wird BFG_WTO auf YES gesetzt, werden Nachrichten, die im Jobprotokoll angezeigt werden, mit WTO angezeigt

**export BFG_GROUP_NAME=MQADM**
Benutzer-IDs, die zu einer bestimmten Gruppe gehören, werden als IBM MQ for z/OS-Administratoren eingestuft.

**export BFG_PROD="/mqm/V9R2M0/mqft"**
Ist der Pfad, in dem sich der MFT-Code befindet

**export BFG_DATA= "/u/johndoe/fteconfig"**
Ist, wo die MFT-Konfigurationsdaten gespeichert sind

**# export BFG_LANG = "-Duser.language = de"**
Ist eine auskommentierte Anweisung zum Definieren der Sprache als Deutsch

**export BFG_LANG = ""**
Gibt die Sprache als Standardsprache (Englisch) an.

Der Inhalt des MFT-Produkts in `/lib/messages/BFGNVMessages_*.properties` listet die verfügbaren Sprachen auf. Standardmäßig wird der Wert leer gelassen, d. es bedeutet, dass Englisch verwendet wird.

**Zugehörige Tasks**

„Managed File Transfer for z/OS konfigurieren” auf Seite 784
Bei Managed File Transfer for z/OS ist eine Anpassung der Komponente erforderlich, damit diese ordnungsgemäß funktioniert.

Managed File Transfer planen

# IBM i MFT unter IBM i konfigurieren

Damit Managed File Transfer nach der Installation eingesetzt werden kann, müssen zunächst einige Konfigurationsschritte für den Koordinationswarteschlangenmanager und den Agenten ausgeführt werden.

## Informationen zu diesem Vorgang

Nach der Installation müssen die in Managed File Transfer für neue Koordinationswarteschlangenmanager und Agenten bereitgestellten Konfigurationsscripts ausgeführt werden, damit die Koordinationswarteschlangenmanager und Agenten für die Dateiübertragung eingesetzt werden können. Anschließend müssen die von Ihnen erstellten Agenten gestartet werden.

## Vorgehensweise

1. Für alle neuen Koordinations-WS-Manager: Führen Sie die MQSC-Befehle in der Datei *coordination_qmgr_name*.mqsc für den Koordinationswarteschlangenmanager aus. Wenn sich der Koordinations-WS-Manager nicht auf demselben Computer wie die Installation befindet, kopieren Sie die MQSC-Scriptdatei auf den Computer, auf dem sich der Warteschlangenmanager befindet, und führen Sie dann das Script aus.
   a) Starten Sie aus einer IBM i-Befehlszeile Qshell mithilfe des folgenden Befehls: `CALL QSHELL`
   b) Wechseln Sie in das folgende Verzeichnis: `/QIBM/UserData/mqm/mqft/config/`*coordination_qmgr_name*
   c) Geben Sie den folgenden Befehl aus, und ersetzen Sie *coordination_qmgr_name* durch den Namen Ihres Warteschlangenmanagers:

   ```
   /QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
   ```

   Sie können den Koordinationswarteschlangenmanager stattdessen manuell konfigurieren. Weitere Informationen finden Sie unter „Koordinationswarteschlangenmanager für MFT konfigurieren” auf Seite 824.
2. Für alle neuen Agenten: Führen Sie die MQSC-Befehle in der *agent_name*_create.mqsc-Datei für den Agentenwarteschlangenmanager aus.
   Wenn sich der Agentenwarteschlangenmanager nicht auf demselben Computer wie der Agent befindet, kopieren Sie die MQSC-Scriptdatei auf den Computer, auf dem sich der WS-Manager befindet, und führen Sie das Script aus.
   a) Starten Sie aus einer IBM i-Befehlszeile Qshell mithilfe des folgenden Befehls: `CALL QSHELL`
   b) Wechseln Sie in das folgende Verzeichnis: `/QIBM/UserData/mqm/mqft/config/`*agent_qmgr_name*`/agents`
   c) Setzen Sie den folgenden Befehl ab, indem Sie *agent_qmgr_name* durch den Namen Ihres Agentenwarteschlangenmanagers ersetzen und *agent_name* durch den Namen Ihres Agenten ersetzen:

   ```
   /QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
   ```

   Sie können den Agentenwarteschlangenmanager stattdessen manuell konfigurieren. Weitere Informationen finden Sie unter „MFT-Agentenwarteschlangenmanager konfigurieren” auf Seite 832.
3. Wenn Sie das Subsystem QMFT noch nicht als Teil der Installation gestartet haben, starten Sie das Subsystem QMFT über die IBM i -Befehlszeile mit dem folgenden Befehl: `STRSBS SBSD(QMQMMFT/QMFT)` oder `STRSBS QMQMMFT/QMFT` .
4. Starten Sie Ihre neuen Agenten mit dem Befehl **fteStartAgent** .

a) Starten Sie aus einer IBM i-Befehlszeile Qshell mithilfe des folgenden Befehls: `CALL QSHELL`

b) Wechseln Sie in das folgende Verzeichnis: `/QIBM/ProdData/mqm/bin`

c) Geben Sie den folgenden Befehl aus, und ersetzen Sie AGENT durch den Namen Ihres Agenten:

```
./fteStartAgent AGENT
```

### Nächste Schritte

Es wird empfohlen, Sandboxes einzurichten, um die Bereiche des Dateisystems zu begrenzen, auf die ein Agent zugreifen kann. Diese Funktion wird im Abschnitt Mit MFT-Agenten-Sandboxes arbeiten beschrieben.

**Zugehörige Konzepte**

„MFT für erstmalige Verwendung konfigurieren” auf Seite 821
Einige Konfigurationstasks für Managed File Transfer-Agenten und Warteschlangenmanager müssen einmal vor ihrer ersten Verwendung ausgeführt werden.

## MFT für erstmalige Verwendung konfigurieren

Einige Konfigurationstasks für Managed File Transfer-Agenten und Warteschlangenmanager müssen einmal vor ihrer ersten Verwendung ausgeführt werden.

**Zugehörige Konzepte**

„Verbindung zu IBM MQ herstellen” auf Seite 822
Die gesamte Netzkommunikation mit IBM MQ-Warteschlangenmanagern (dazu gehört auch die Kommunikation von Managed File Transfer) erfolgt über IBM MQ-Kanäle. Ein IBM MQ-Kanal stellt ein Ende einer Netzverbindung dar. Kanäle werden entweder als Nachrichtenkanal oder als MQI-Kanal klassifiziert.

„Multi-Instanz-Warteschlangenmanager für die Arbeit mit MFT konfigurieren” auf Seite 828
IBM WebSphere MQ 7.0.1 und höher unterstützt die Erstellung von Multi-Instanz-Warteschlangenmanagern. Ein WS-Manager mit mehreren Instanzen wird automatisch auf einem Standby-Server erneut gestartet. Managed File Transfer unterstützt Verbindungen zu Multi-Instanz-Agenten-, Multi-Instanz-Koordinations- und Multi-Instanz-Befehlswarteschlangenmanagern.

**Zugehörige Tasks**

„MFT-Netzwarteschlangenmanager konfigurieren” auf Seite 823
Sind in Ihrem Managed File Transfer-Netz mehrere IBM MQ-Warteschlangenmanager vorhanden, muss zwischen diesen IBM MQ-Warteschlangenmanagern eine Remotekommunikation möglich sein.

„MFT-Agentenwarteschlangenmanager konfigurieren” auf Seite 832
Führen Sie nach der Installation das Script *agent_name*`_create.mqsc` im Verzeichnis *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/agents/`*agent_name* aus, um die erforderliche Konfiguration für den Agentenwarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell ausführen möchten, führen Sie diese Schritte auf dem Agenten-WS-Manager aus.

„Koordinationswarteschlangenmanager für MFT konfigurieren” auf Seite 824
Führen Sie nach der Ausführung des Befehls **fteSetupCoordination** das Script *coordination_qmgr_name*`.mqsc` im Verzeichnis *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name* aus, um die erforderliche Konfiguration für den Koordinationswarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell durchführen möchten, gehen Sie für den Koordinations-Warteschlangenmanager wie im Folgenden beschrieben vor.

„Dataset für MFT-Agenten- oder -Protokollfunktionsbefehle erstellen” auf Seite 783
Aus dem Managed File Transfer-Befehlsvorlagen-Dataset können Sie für eine bestimmte Koordination ein PDSE-Dataset mit Befehlen für einen bestimmten Managed File Transfer Agent oder Managed File Transfer Logger erstellen.

„Vorhandenes Dataset für MFT-Agenten- oder -Protokollfunktionsbefehle unter z/OS aktualisieren” auf Seite 796
Sie können ein aus dem Managed File Transfer-BefehlsvorlagenDataset erstelltes PDSE-Bibliotheks-Dataset für Managed File Transfer-Befehle aktualisieren.

**Zugehörige Verweise**

Einstellungen von MFT-Agentenwarteschlangen

MFT-Systemwarteschlangen und der Systemabschnitt

„MFT-Protokollnachrichten speichern" auf Seite 830

Managed File Transfer sendet Informationen zum Fortschritt der Dateiübertragung und Protokollinformationen an den Koordinationswarteschlangenmanager. Der Koordinationswarteschlangenmanager veröffentlicht diese Informationen in allen übereinstimmenden Subskriptionen für das Thema SYSTEM.FTE. Wenn keine Subskriptionen vorhanden sind, werden diese Informationen nicht beibehalten.

## Verbindung zu IBM MQ herstellen

Die gesamte Netzkommunikation mit IBM MQ-Warteschlangenmanagern (dazu gehört auch die Kommunikation von Managed File Transfer) erfolgt über IBM MQ-Kanäle. Ein IBM MQ-Kanal stellt ein Ende einer Netzverbindung dar. Kanäle werden entweder als Nachrichtenkanal oder als MQI-Kanal klassifiziert.

### Managed File Transfer und Kanäle

Managed File Transfer verbindet mithilfe von MQI-Kanälen Agenten im Clientmodus mit den entsprechenden Agentenwarteschlangenmanagern und Befehlsanwendungen (z. B. **`fteCreateTransfer`**) mit den entsprechenden Befehls- und Koordinationswarteschlangenmanagern. In der Standardkonfiguration werden diese Verbindungen über einen SVRCONN-Kanal namens SYSTEM.DEF.SVRCONN hergestellt, der standardmäßig auf allen Warteschlangenmanagern vorhanden ist. Aufgrund dieser Standardeinstellungen müssen für eine Managed File Transfer-Basisinstallation keine MQI-Kanäle geändert werden.

Es gibt sechs Typen von Nachrichtenkanalendpunkten. In diesem Abschnitt werden aber nur Sender-/Empfängerpaare behandelt. Informationen zu anderen Kanalkombinationen finden Sie unter Verteilte Warteschlangenkomponenten.

### Erforderliche Nachrichtenpfade

IBM MQ-Nachrichten können nur über Kanäle übertragen werden, daher müssen Sie sicherstellen, dass für alle von Managed File Transfer benötigten Nachrichtenpfade Kanäle vorhanden sind. Es muss sich dabei nicht um direkte Pfade handeln. Nachrichten können bei Bedarf auch über zwischengeschaltete Warteschlangenmanager übertragen werden. In diesem Abschnitt wird allerdings nur die direkte Punkt-zu-Punkt-Kommunikation behandelt. Weitere Informationen zu diesen Optionen finden Sie unter Vorgehensweise beim Abrufen des fernen Warteschlangenmanagers.

Von Managed File Transfer werden folgende Kommunikationspfade verwendet:

**Agent zu Agent**
: Für alle Agenten, zwischen denen Dateien übertragen werden, ist eine bidirektionale Kommunikation zwischen den Warteschlangenmanagern der Agenten erforderlich. Da dieser Pfad die Massendaten überträgt, sollte er je nach Anforderungen so kurz, schnell oder günstig wie möglich sein.

**Agent zu Koordinationswarteschlangenmanager**
: Die Protokollnachrichten der an einer Übertragung beteiligten Agenten müssen den Koordinationswarteschlangenmanager erreichen können.

**Befehlswarteschlangenmanager zu Agent**
: Jeder Warteschlangenmanager, zu dem Befehlsanwendungen oder IBM MQ Explorer (über den Befehlswarteschlangenmanager) eine Verbindung herstellen, muss Nachrichten an die Warteschlangenmanager der Agenten senden können, die durch diese Befehlsanwendungen gesteuert werden. Damit die Befehle Rückmeldungen anzeigen können, sollten Sie eine bidirektionale Verbindung einrichten.

Weitere Informationen finden Sie im Abschnitt *IBM MQ-Installation überprüfen* für die Plattform bzw. Plattformen, die in Ihrem Unternehmen eingesetzt werden.

**Zugehörige Konzepte**

„Multi-Instanz-Warteschlangenmanager für die Arbeit mit MFT konfigurieren" auf Seite 828

IBM WebSphere MQ 7.0.1 und höher unterstützt die Erstellung von Multi-Instanz-Warteschlangenmanagern. Ein WS-Manager mit mehreren Instanzen wird automatisch auf einem Standby-Server erneut

gestartet. Managed File Transfer unterstützt Verbindungen zu Multi-Instanz-Agenten-, Multi-Instanz-Koordinations- und Multi-Instanz-Befehlswarteschlangenmanagern.

**Zugehörige Tasks**

„MFT-Netzwarteschlangenmanager konfigurieren" auf Seite 823
Sind in Ihrem Managed File Transfer-Netz mehrere IBM MQ-Warteschlangenmanager vorhanden, muss zwischen diesen IBM MQ-Warteschlangenmanagern eine Remotekommunikation möglich sein.

„Koordinationswarteschlangenmanager für MFT konfigurieren" auf Seite 824
Führen Sie nach der Ausführung des Befehls **fteSetupCoordination** das Script *coordination_qmgr_name*.mqsc im Verzeichnis *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* aus, um die erforderliche Konfiguration für den Koordinationswarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell durchführen möchten, gehen Sie für den Koordinations-Warteschlangenmanager wie im Folgenden beschrieben vor.

# MFT-Netzwarteschlangenmanager konfigurieren

Sind in Ihrem Managed File Transfer-Netz mehrere IBM MQ-Warteschlangenmanager vorhanden, muss zwischen diesen IBM MQ-Warteschlangenmanagern eine Remotekommunikation möglich sein.

## Informationen zu diesem Vorgang

Es gibt zwei Möglichkeiten, die WS-Manager so zu konfigurieren, dass sie miteinander kommunizieren können:

- Durch Einrichtung eines Clusters aus IBM MQ-Warteschlangenmanagern.

  Informationen zu IBM MQ-Warteschlangenmanagerclustern und zu deren Konfiguration finden Sie unter „WS-Manager-Cluster konfigurieren" auf Seite 312.
- Durch die Einrichtung von Kanälen zwischen den WS-Managern, die wie folgt beschrieben werden:

### Kanäle zwischen Warteschlangenmanagern einrichten

Richten Sie die folgenden Nachrichtenkanäle zwischen Ihren Warteschlangenmanagern ein:

- Vom Agenten-WS-Manager zum Koordinationswarteschlangenmanager
- Vom Befehlswarteschlangenmanager zum Agentenwarteschlangenmanager.
- Vom Agenten-WS-Manager zum Befehlswarteschlangenmanager (zum Aktivieren von Feedback-Nachrichten, die von den Befehlen angezeigt werden sollen).
- Vom Befehlswarteschlangenmanager zum Koordinationswarteschlangenmanager
- Vom Agentenwarteschlangenmanager zu einem anderen Agentenwarteschlangenmanager im Managed File Transfer-Netz

Eine Einführung in die Konfiguration dieser Kommunikation finden Sie im Abschnitt Verwaltung remote angebundener IBM MQ-Objekte mithilfe von MQSC.

Einige empfohlene Beispielschritte sind:

## Vorgehensweise

1. Erstellen Sie eine Übertragungswarteschlange auf dem IBM MQ-Warteschlangenmanager, der denselben Namen aufweist wie der Koordinationswarteschlangenmanager.

   Sie können den folgenden MQSC-Befehl verwenden:

   ```
   DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
   ```
2. Erstellen Sie im IBM MQ-Warteschlangenmanager einen Senderkanal zum Koordinationswarteschlangenmanager von Managed File Transfer.

Der Name der Übertragungswarteschlange, die im vorherigen Schritt erstellt wurde, ist ein erforderlicher Parameter für diesen Kanal.

Für Agenten unter Managed File Transfer for IBM MQwerden Nachrichten in einem leeren Format veröffentlicht.

Sie können den folgenden MQSC-Befehl verwenden:

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')
 XMITQ(coordination-qmgr-name) CONVERT(NO)
```

**Anmerkung:** Setzen Sie CONVERT (NO) nur auf, wenn dies erforderlich ist.

3. Erstellen Sie im Koordinationswarteschlangenmanager von Managed File Transfer einen Empfängerkanal zum IBM MQ-Warteschlangenmanager. Geben Sie diesem Empfängerkanal denselben Namen wie dem Senderkanal auf dem IBM MQ-Warteschlangenmanager.

   Sie können den folgenden MQSC-Befehl verwenden:

   ```
   DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
   ```

### Nächste Schritte

Führen Sie als Nächstes die Konfigurationsschritte für Ihren Koordinationswarteschlangenmanager aus: „Koordinationswarteschlangenmanager für MFT konfigurieren” auf Seite 824.

**Zugehörige Konzepte**

„Verbindung zu IBM MQ herstellen” auf Seite 822
Die gesamte Netzkommunikation mit IBM MQ-Warteschlangenmanagern (dazu gehört auch die Kommunikation von Managed File Transfer) erfolgt über IBM MQ-Kanäle. Ein IBM MQ-Kanal stellt ein Ende einer Netzverbindung dar. Kanäle werden entweder als Nachrichtenkanal oder als MQI-Kanal klassifiziert.

„Multi-Instanz-Warteschlangenmanager für die Arbeit mit MFT konfigurieren” auf Seite 828
IBM WebSphere MQ 7.0.1 und höher unterstützt die Erstellung von Multi-Instanz-Warteschlangenmanagern. Ein WS-Manager mit mehreren Instanzen wird automatisch auf einem Standby-Server erneut gestartet. Managed File Transfer unterstützt Verbindungen zu Multi-Instanz-Agenten-, Multi-Instanz-Koordinations- und Multi-Instanz-Befehlswarteschlangenmanagern.

**Zugehörige Tasks**

„Koordinationswarteschlangenmanager für MFT konfigurieren” auf Seite 824
Führen Sie nach der Ausführung des Befehls **fteSetupCoordination** das Script *coordination_qmgr_name*.mqsc im Verzeichnis *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* aus, um die erforderliche Konfiguration für den Koordinationswarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell durchführen möchten, gehen Sie für den Koordinations-Warteschlangenmanager wie im Folgenden beschrieben vor.

## Koordinationswarteschlangenmanager für MFT konfigurieren

Führen Sie nach der Ausführung des Befehls **fteSetupCoordination** das Script *coordination_qmgr_name*.mqsc im Verzeichnis *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* aus, um die erforderliche Konfiguration für den Koordinationswarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell durchführen möchten, gehen Sie für den Koordinations-Warteschlangenmanager wie im Folgenden beschrieben vor.

### Informationen zu diesem Vorgang

### Vorgehensweise

1. Erstellen Sie eine lokale Warteschlange mit dem Namen SYSTEM.FTE.
2. Fügen Sie die Warteschlange SYSTEM.FTE der Namensliste SYSTEM.QPUBSUB.QUEUE.NAMELIST hinzu.

3. Erstellen Sie ein Thema mit dem Namen SYSTEM.FTE und der Themazeichenfolge SYSTEM.FTE.
4. Vergewissern Sie sich, dass die Attribute des Themas SYSTEM.FTE für die nicht persistente Nachrichtenübermittlung (NPMSGDLV) und persistente Nachrichtenübermittlung (PMSGDLV) auf den Wert ALLAVAIL gesetzt sind.
5. Vergewissern Sie sich, dass das Attribut PSMODE (Publish/Subscribe-Modus) des Warteschlangenmanagers auf ENABLED gesetzt ist.

### Nächste Schritte

Wenn Sie den Befehl `strmqm -c` auf einem Warteschlangenmanager ausführen, der als Koordinations-Warteschlangenmanager konfiguriert wurde, löscht der Befehl die in Schritt 2 vorgenommene Änderung ( SYSTEM.FTE -Warteschlange an das SYSTEM.QPUBSUB.QUEUE.NAMELIST -Namensliste). Dies liegt daran, dass `strmqm -c` die IBM MQ-Standardobjekte erneut erstellt und die Managed File Transfer-Änderungen zurücknimmt. Falls Sie den Warteschlangenmanager mit `strmqm -c` gestartet haben, müssen Sie deshalb einen der folgenden Schritte ausführen:

- Führen Sie das Script *coordination_qmgr_name*`.mqsc` auf dem Warteschlangenmanager erneut aus.
- Wiederholen Sie Schritt 2.

**Zugehörige Konzepte**

„Verbindung zu IBM MQ herstellen” auf Seite 822
Die gesamte Netzkommunikation mit IBM MQ-Warteschlangenmanagern (dazu gehört auch die Kommunikation von Managed File Transfer) erfolgt über IBM MQ-Kanäle. Ein IBM MQ-Kanal stellt ein Ende einer Netzverbindung dar. Kanäle werden entweder als Nachrichtenkanal oder als MQI-Kanal klassifiziert.

„Multi-Instanz-Warteschlangenmanager für die Arbeit mit MFT konfigurieren” auf Seite 828
IBM WebSphere MQ 7.0.1 und höher unterstützt die Erstellung von Multi-Instanz-Warteschlangenmanagern. Ein WS-Manager mit mehreren Instanzen wird automatisch auf einem Standby-Server erneut gestartet. Managed File Transfer unterstützt Verbindungen zu Multi-Instanz-Agenten-, Multi-Instanz-Koordinations- und Multi-Instanz-Befehlswarteschlangenmanagern.

**Zugehörige Tasks**

„MFT-Netzwarteschlangenmanager konfigurieren” auf Seite 823
Sind in Ihrem Managed File Transfer-Netz mehrere IBM MQ-Warteschlangenmanager vorhanden, muss zwischen diesen IBM MQ-Warteschlangenmanagern eine Remotekommunikation möglich sein.

**Zugehörige Verweise**

fteSetupCoordination

## IBM MQ File Transfer-Struktur erstellen

Sie können eine Managed File Transfer-Struktur mit einem einzelnen Agenten konfigurieren, der mit einem Warteschlangenmanager auf demselben System verbunden ist.

### Informationen zu diesem Vorgang

Die MFT-Konfiguration wird in einer Dateistruktur im Datenpfad von IBM MQ auf dem System, auf dem sich der Agent befindet, gespeichert.

Für die folgende Beispielkonfiguration wurden ein MFT in IBM MQ 8.0-Warteschlangenmanager mit dem Namen SAMPLECOORD (mit inaktivierter Sicherheit) und ein einzelner MFT-Agent mit dem Namen SAMPLEAGENT verwendet:

```
+--- config
             +--- SAMPLECOORD
                            +--- command.properties
                            +--- coordination.properties
                            +--- SAMPLECOORD.mqsc
                            +--- agents
                                          +--- SAMPLEAGENT
                                                        +--- agent.properties
                                                        +--- SAMPLEAGENT_create.mqsc
```

```
                                                          +--- SAMPLEAGENT_delete.mqsc
 +---logs
             +--- SAMPLECOORD
                             +--- agents
                                           +--- SAMPLEAGENT
                                                          +--- logs
```

In diesem Beispiel wird davon ausgegangen, dass die Sicherheit des Warteschlangenmanagers inaktiviert wurde. Die folgenden Befehle, die in **runmqsc**ausgeführt werden, inaktivieren die Sicherheit, nachdem der Warteschlangenmanager erneut gestartet wurde:

```
runmqsc queue manager
 alter qmgr CONNAUTH(NONE);
 alter qmgr CHLAUTH(DISABLED);
 end;
```

Für die Konfiguration mit aktivierter Sicherheit in MFT in IBM MQ 8.0 oder höher erfordert **CONNAUTH** alle MFT -Befehle, die eine Verbindung mit einem Warteschlangenmanager herstellen, um Benutzer-ID-und Kennwortberechtigungsnachweise bereitzustellen. Sie können die zusätzlichen Parameter **-mquserid** und **-mqpassword** für jeden Befehl anwenden oder eine MQMFTCredentials.xml -Datei definieren. In der folgenden Beispielberechtigungsnachweisdatei wird die Benutzer-ID fteuserdefiniert, für die das Kennwort MyPassword verwendet werden soll, wenn eine Verbindung zum Warteschlangenmanager SAMPLECOORDhergestellt wird:

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">
    <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPELCOORD"/>
</tns:mqmftCredentials>
```

Weitere Informationen finden Sie unter MFT und IBM MQ-Verbindungsauthentifizierung.

**Anmerkungen:**

- Zur Lokalisierung des Konfigurationsverzeichnisses von MFT geben Sie den Befehl **fteDisplayVersion -v** ein.
- Für z/OS -Benutzer kann die Datei MQMFTCredential.xml als Member in einer partitionierten Datei mit einem variablen Satzformat (RECFM = V) oder einem nicht definierten Satzformat (RECFM = U) lokalisiert werden.
- Fügen Sie für die Konfiguration mit aktivierter Sicherheit den folgenden Parameter zu den folgenden Schritten hinzu, um die Berechtigungsnachweise dem relevanten Warteschlangenmanager zuzuordnen: -F *full_credential_file_path*.
- Das Klartextkennwort in der MQMFTCredential.xml kann mit dem folgenden Befehl verschleiert werden:

  ```
  fteObfuscate -f full_file_path_to_MQMFTCredentials.xml
  ```

## Vorgehensweise

1. Erstellen Sie einen Koordinations-WS-Manager.

   Ein Koordinations-WS-Manager ist ein einzelner Warteschlangenmanager, mit dem alle Übertragungs- und Statusinformationen von seinen Agenten empfangen werden. Führen Sie den folgenden Befehl aus:

   ```
   fteSetupCoordination -coordinationQMgr coordination_qmgr_name
   ```

   Dadurch wird die grundlegende Konfiguration der obersten Ebene erstellt und eine IBM MQ-Scriptdatei für den Aufruf von *coordination_qmgr_name*.mqsc erstellt.

Die Konfiguration muss dann mit folgendem IBM MQ-Befehl in den Warteschlangenmanager geladen werden:

```
runmqsc queue manager name < coordination_qmgr_name.mqsc
```

**Anmerkung:** Für die TCP-Clientverbindung zu einem WS-Manager können Sie Folgendes verwenden:

```
fteSetupCoordination -coordinationQMgr coordination_qmgr_name
-coordinationQMgrHost coordination_qmgr_host -coordinationQMgrPort coordination_qmgr_port
-coordinationQMgrChannel coordination_qmgr_channel
```

Für den erstellten *coordination_qmgr_name*.mqscmüssen Sie den Befehl **runmqsc** auf derselben Maschine ausführen, auf der der Koordinationswarteschlangenmanager ausgeführt wird.

2. Erstellen Sie den Befehlswarteschlangenmanager.

   Ein Befehlswarteschlangenmanager ist ein einzelner Warteschlangenmanager, der so vorkonfiguriert wurde, dass die IBM MQ -Infrastruktur MFT -Anforderungen an den entsprechenden Agenten weiterleiten kann. Führen Sie den folgenden Befehl aus:

   ```
   fteSetupCommands -connectionQMgr Command QM Name -p Coordination QM Name
   ```

   Dadurch wird eine command.properties-Datei im Koordinationsverzeichnis erstellt. Beachten Sie, dass der -p optional ist und nicht erforderlich ist, wenn die Befehle für die Standardkoordinierung konfiguriert werden.

   **Anmerkung:** Für die TCP-Clientverbindung zu einem WS-Manager können Sie Folgendes verwenden:

   ```
   fteSetupCommands -p coordination_qmgr_name -commandQMgr connection_qmgr_name
   -commandQMgrHost connection_qmgr_host -commandQMgrPort connection_qmgr_port
   -commandQMgrChannel connection_qmgr_channel
   ```

3. Erstellen Sie den Agenten.

   Ein Agent ist eine Anwendung, die Dateien senden und empfangen kann. Führen Sie den folgenden Befehl aus:

   ```
   fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
   ```

   Dadurch wird die Agentenkonfiguration unter der Koordination erstellt und eine IBM MQ -Scriptdatei erstellt, um *agent_name*.mqsc im Konfigurationsverzeichnis des Agenten aufzurufen.

   Führen Sie den folgenden IBM MQ-Befehl aus, um die IBM MQ-Scriptdatei in den Warteschlangenmanager zu laden:

   ```
   runmqsc agent_qmgr_name < agent_name_create.mqsc file
   ```

   **Anmerkung:** Für die TCP-Clientverbindung zu einem WS-Manager können Sie Folgendes verwenden:

   ```
   fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
   -agentQMgrHost agent_qmgr_host -agentQMgrPort agent_qmgr_port -agentQMgrChannel
   agent_qmgr_channel
   ```

4. Starten Sie den Agenten.

   Führen Sie den folgenden Befehl aus:

   ```
   fteStartAgent -p coordination_qmgr_name agentName
   ```

   Der Agent wird im Hintergrund gestartet, und die Eingabeaufforderung wird zurückgegeben. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent aktiv ist:

```
fteListAgents -p coordination_qmgr_name
```

Hier wird der Status der Agenten angezeigt. Wenn der Agent erfolgreich ausgeführt wird, wird er wie im Status READY aufgelistet.

### Ergebnisse

Sie verfügen nun über eine grundlegende, einsatzbereite MFT-Infrastruktur, innerhalb der Sie mit dem Befehl **fteCreateTransfer** eine Übertragung anfordern können. Wenn IBM MQ Explorer zur Verfügung steht, können Sie Übertragungen alternativ auch mit den Plug-ins von MFT erstellen und überwachen.

Weitere Agenten können der Konfiguration hinzugefügt werden, indem Sie den Schritt 3: Create the agent (Agent erstellen) wiederholen. Wenn die TCP-Clientverbindung verwendet wird, können diese sich auf verschiedenen Maschinen befinden. Auf verschiedenen Maschinen müssen die Befehle **fteSetupCoordination** und **fteSetupCommands** für jede Maschine wiederholt werden. Die mqsc-Scripts müssen jedoch nicht ausgeführt werden.

Komplexere Konfigurationen können separate WS-Manager für die Koordination und jeden Agenten haben. In diesen Fällen müssen die verschiedenen WS-Manager miteinander verbunden werden.

**Zugehörige Konzepte**

Vorgehensweise, wenn Ihr MFT-Agent vom Befehl **fteListAgents** nicht aufgelistet wird

**Zugehörige Verweise**

fteSetupCoordination

fteSetupCommands: MFT-Datei 'command.properties' erstellen

fteCreateAgent

**fteObfuscate**: Verschlüsselung sensibler Daten

MFT-Berechtigungsnachweisdateiformat

Datei MFT `agent.properties`

## Multi-Instanz-Warteschlangenmanager für die Arbeit mit MFT konfigurieren

IBM WebSphere MQ 7.0.1 und höher unterstützt die Erstellung von Multi-Instanz-Warteschlangenmanagern. Ein WS-Manager mit mehreren Instanzen wird automatisch auf einem Standby-Server erneut gestartet. Managed File Transfer unterstützt Verbindungen zu Multi-Instanz-Agenten-, Multi-Instanz-Koordinations- und Multi-Instanz-Befehlswarteschlangenmanagern.

### Konfigurieren eines Multi-Instanz-Warteschlangenmanagers

**Wichtig:** Informationen zum Konfigurieren eines IBM MQ-Multi-Instanz-Warteschlangenmanagers finden Sie unter „Warteschlangenmanager mit mehreren Instanzen” auf Seite 546. Stellen Sie sicher, dass Sie diese Informationen gelesen haben, bevor Sie versuchen, einen Multi-Instanz-Warteschlangenmanager für die Arbeit mit Managed File Transferzu konfigurieren.

### Warteschlangenmanager mit mehreren Instanzen als Agentenwarteschlangenmanager verwenden

Um einen Agenten für die Verbindung zu der aktiven Instanz und der Standby-Instanz Ihres Multi-Instanz-Warteschlangenmanagers zu aktivieren, fügen Sie die Eigenschaft `agentQMgrStandby` zur Datei `agent.properties` des Agenten hinzu. Die Eigenschaft `agentQMgrStandby` definiert den Hostnamen und die Portnummer, die für Clientverbindungen für die Standby-Warteschlangenmanagerinstanz verwendet werden. Der Wert der Eigenschaft muss im MQ-CONNAME-Format angegeben werden, also *host_name*(*port_number*).

Die Eigenschaft `agentQMgr` gibt den Namen des Multi-Instanz-Warteschlangenmanagers an. Die Eigenschaft `agentQMgrHost` gibt den Hostnamen für die aktive WS-Manager-Instanz an, und die Eigenschaft `agentQMgrPort` gibt die Portnummer für die aktive WS-Manager-Instanz an. Der Agent muss sich im

Clientmodus sowohl mit der aktiven als auch mit der Standby-Instanz des Warteschlangenmanagers mit mehreren Instanzen verbinden.

Weitere Informationen finden Sie unter Die MFT `agent.properties`-Datei.

Dieses Beispiel zeigt den Inhalt der `agent.properties`-Datei für AGENT1, die eine Verbindung zu einem Multi-Instanz-Warteschlangenmanager mit dem Namen QM_JUPITER herstellt. Die aktive Instanz von QM_JUPITER befindet sich auf dem System host1 und verwendet die Anschlussnummer 1414 für Clientverbindungen. Die Standby-Instanz von QM_JUPITER befindet sich auf dem System host2 und verwendet die Portnummer 1414 für Clientverbindungen.

```
agentName=AGENT1
agentDesc=
agentQMgr=QM_JUPITER
agentQMgrPort=1414
agentQMgrHost=host1
agentQMgrChannel=SYSTEM.DEF.SVRCONN
agentQMgrStandby=host2(1414)
```

## Warteschlangenmanager mit mehreren Instanzen als Koordinationswarteschlangenmanager verwenden

Um Verbindungen zur aktiven Instanz und der Standby-Instanz Ihres Koordinationswarteschlangenmanagers mit mehreren Instanzen zu aktivieren, fügen Sie die Eigenschaft `coordinationQMgrStandby` allen Dateien des Typs `coordination.properties` in Ihrer Managed File Transfer-Topologie hinzu.

Weitere Informationen finden Sie im Abschnitt Die MFT-Datei 'coordination.properties'.

Dieses Beispiel zeigt den Inhalt einer `coordination.properties`-Datei, die die Verbindungsdetails zu einem Multi-Instanz-Koordinationswarteschlangenmanager mit dem Namen QM_SATURN angibt. Die aktive Instanz von QM_SATURN befindet sich auf dem System 'coordination_host1' und verwendet die Portnummer 1420 für Clientverbindungen. Die Standby-Instanz von QM_SATURN befindet sich auf dem System coordination_host2 und verwendet die Portnummer 1420 für Clientverbindungen.

```
coordinationQMgr=QM_SATURN
coordinationQMgrHost=coordination_host1
coordinationQMgrPort=1420
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
coordinationQMgrStandby=coordination_host2(1420)
```

Die eigenständige Protokollfunktion von Managed File Transfer muss mit ihrem Warteschlangenmanager immer im Bindungsmodus verbunden sein. Wenn Sie die eigenständige Protokollfunktion mit einem Multi-Instanz-Koordinations-WS-Manager verwenden, verbinden Sie die eigenständige Protokollfunktion im Bindungsmodus mit einem anderen Warteschlangenmanager. Die hierfür erforderlichen Schritte werden im Abschnitt „Alternative Konfigurationen für eine eigenständige MFT-Protokollfunktion” auf Seite 856 beschrieben. Sie müssen die Kanäle zwischen dem Warteschlangenmanager des eigenständigen Protokollmanagers und dem Koordinations-WS-Manager mit dem Hostnamen und der Portnummer der beiden Instanzen des Koordinations-WS-Managers für mehrere Instanzen definieren. Informationen zur Vorgehensweise finden Sie in „Warteschlangenmanager mit mehreren Instanzen” auf Seite 546.

Das Managed File Transfer-Plug-in für IBM MQ Explorer stellt eine Verbindung zum Koordinationswarteschlangenmanager im Clientmodus her. Wenn die aktive Instanz des Multi-Instanz-Koordinations-WS-Managers fehlschlägt, wird die Standby-Instanz des Koordinations-WS-Managers aktiv und das Plug-in wird erneut verbunden.

Die Managed File Transfer-Befehle **`fteList*`** und **`fteShowAgentDetails`** stellen eine direkte Verbindung zum Koordinationswarteschlangenmanager her. Wenn die aktive Instanz der Multi-Instanz-Koordination nicht verfügbar ist, versuchen diese Befehle, eine Verbindung zur Standby-Instanz des Koordinations-WS-Managers herzustellen.

### Verwenden eines Warteschlangenmanagers mit mehreren Instanzen als Befehlswarteschlangenmanager

Um Verbindungen sowohl zur aktiven als auch zur Standby-Instanz Ihres Multi-Instanz-Befehlswarteschlangenmanagers zu ermöglichen, fügen Sie die Eigenschaft `connectionQMgrStandby` allen `command.properties` -Dateien in Ihrer Managed File Transfer -Topologie hinzu.

Weitere Informationen finden Sie im Abschnitt Die MFT-Datei 'command.properties'.

Dieses Beispiel zeigt den Inhalt einer `command.properties`-Datei, die die Verbindungsdetails zu einem Multi-Instanz-Befehlswarteschlangenmanager mit dem Namen QM_MARS angibt. Die aktive Instanz von QM_MARS befindet sich auf dem System command_host1 und verwendet die Portnummer 1424 für Clientverbindungen. Die Standby-Instanz von QM_MARS befindet sich auf dem System command_host2 und verwendet die Portnummer 1424 für Clientverbindungen.

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

**Zugehörige Konzepte**

„Verbindung zu IBM MQ herstellen” auf Seite 822
Die gesamte Netzkommunikation mit IBM MQ-Warteschlangenmanagern (dazu gehört auch die Kommunikation von Managed File Transfer) erfolgt über IBM MQ-Kanäle. Ein IBM MQ-Kanal stellt ein Ende einer Netzverbindung dar. Kanäle werden entweder als Nachrichtenkanal oder als MQI-Kanal klassifiziert.

**Zugehörige Tasks**

„MFT-Netzwarteschlangenmanager konfigurieren” auf Seite 823
Sind in Ihrem Managed File Transfer-Netz mehrere IBM MQ-Warteschlangenmanager vorhanden, muss zwischen diesen IBM MQ-Warteschlangenmanagern eine Remotekommunikation möglich sein.

„Koordinationswarteschlangenmanager für MFT konfigurieren” auf Seite 824
Führen Sie nach der Ausführung des Befehls **fteSetupCoordination** das Script *coordination_qmgr_name*`.mqsc` im Verzeichnis *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name* aus, um die erforderliche Konfiguration für den Koordinationswarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell durchführen möchten, gehen Sie für den Koordinations-Warteschlangenmanager wie im Folgenden beschrieben vor.

## MFT-Protokollnachrichten speichern

Managed File Transfer sendet Informationen zum Fortschritt der Dateiübertragung und Protokollinformationen an den Koordinationswarteschlangenmanager. Der Koordinationswarteschlangenmanager veröffentlicht diese Informationen in allen übereinstimmenden Subskriptionen für das Thema SYSTEM.FTE. Wenn keine Subskriptionen vorhanden sind, werden diese Informationen nicht beibehalten.

### Optionen, um das Beibehalten von Informationen sicherzustellen

Wenn der Fortschritt der Übertragung oder die Protokollinformationen für Ihr Unternehmen von Bedeutung sind, müssen Sie einen der folgenden Schritte ausführen, um sicherzustellen, dass die Informationen beibehalten werden:

- Verwenden Sie die Managed File Transfer -Datenbankprotokollfunktion, um auf dem SYSTEM.FTE/Log in eine Oracle -oder Db2 -Datenbank.
- Definieren Sie eine Subskription für SYSTEM.FTE , das Veröffentlichungen in einer IBM MQ -Warteschlange speichert. Definieren Sie diese Subskription, bevor Sie Dateiübertragungen übertragen, um sicherzustellen, dass alle Fortschritt-und Protokollnachrichten in der Warteschlange gespeichert werden.
- Schreiben Sie eine Anwendung, von der die Schnittstelle für Nachrichtenwarteschlangen (MQI) oder der IBM MQ JMS für die Erstellung einer permanenten Subskription und die Verarbeitung der Veröffentlichungen genutzt wird, die an die Subskription übergeben werden. Diese Anwendung muss in Betrieb

sein, bevor Dateien übertragen werden, um sicherzustellen, dass die Anwendung alle Fortschritt-und Protokollnachrichten empfängt.

Jeder dieser Ansätze wird in den folgenden Abschnitten ausführlicher beschrieben.

Wenn es darum geht, Protokollinformationen längerfristig zu speichern, sollten Sie sich nicht auf das IBM MQ Explorer-Plug-in verlassen.

## Managed File Transfer-Datenbankprotokollfunktion für das Beibehalten von Protokollnachrichten verwenden

Die Datenbankprotokollfunktion ist eine optionale Komponente von Managed File Transfer, mit der die Protokolldaten zu Analyse- und Prüfungszwecken in eine Datenbank kopiert werden können. Bei dieser Funktion handelt es sich um eine eigenständige Java-Anwendung, die auf einem System installiert wird, das als Host für den Koordinationswarteschlangenmanager und die Datenbank dient. Weitere Informationen zur Datenbankprotokollfunktion finden Sie im Abschnitt „MFT-Protokollfunktion konfigurieren” auf Seite 843.

## Fortschritt und Protokollnachrichten mit dem IBM MQ Explorer-Plug-in beibehalten

Beim ersten Start einer Instanz des IBM MQ Explorer-Plug-ins erstellt die Instanz im Koordinationswarteschlangenmanager eine permanente Subskription. Diese permanente Subskription wird verwendet, um die Informationen zu erfassen, die in den Ansichten **Übertragungsprotokoll** und **Aktueller Übertragungsfortschritt** angezeigt werden.

Der Name der permanenten Subskription hat das Präfix, das anzeigt, dass die Subskription vom IBM MQ Explorer MFT -Plug-in erstellt wurde, den Hostnamen und den Namen des Benutzers. Beispiel: `MQExplorer_MFT_Plugin_HOST_TJWatson`.

Dieses Präfix wird für den Fall hinzugefügt, dass ein Administrator eine permanente Subskription löschen möchte, die nicht mehr aktiv von einer Instanz des IBM MQ Explorer-Plug-ins genutzt wird.

Die Verwendung einer permanenten Subskription auf dem Koordinations-WS-Manager kann dazu führen, dass Nachrichten in den Warteschlangen SYSTEM.MANAGED.DURABLE erstellt werden. Wenn Sie über ein Managed File Transfer-Netz mit großem Volumen verfügen und/oder das IBM MQ Explorer-Plug-in nur selten verwenden, kann das lokale Dateisystem durch diese Nachrichtendaten überlastet werden.

Um dies zu verhindern, geben Sie an, dass das IBM MQ Explorer -Plug-in eine nicht permanente Subskription für den Koordinationswarteschlangenmanager verwenden soll. Führen Sie hierzu die folgenden Schritte in IBM MQ Explorer aus:

1. Wählen Sie **Fenster** > **Benutzervorgaben** > **MQ Explorer** > **Managed File Transfer** aus.
2. Wählen Sie in der Liste **Subskriptionstyp des Übertragungsprotokolls** den Eintrag NON_DURABLE aus.

## Veröffentlichungen in einer IBM MQ-Warteschlange speichern

Wenn Protokoll- oder Fortschrittsnachrichten in einer IBM MQ-Warteschlange gespeichert werden sollen, müssen Sie eine Subskription in dem Koordinationswarteschlangenmanager konfigurieren, der die Nachrichten an diese Warteschlange weiterleitet. Wenn Sie beispielsweise alle Protokollnachrichten an eine Warteschlange mit dem Namen LOG.QUEUE weiterleiten möchten, übergeben Sie den folgenden MQSC-Befehl:

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE)WSCHEMA(TOPIC)
```

Sobald die Protokollnachrichten an eine IBM MQ-Warteschlange weitergeleitet wurden, verbleiben sie in der Warteschlange, bis sie von einer IBM MQ-Anwendung verarbeitet werden, welche die Warteschlange nutzt.

### Schreiben von Anwendungen, die eine permanente Subskription für das Thema SYSTEM.FTE verwalten

Sie können Anwendungen schreiben, die ihre eigenen permanenten Subskriptionen für SYSTEM.FTE unter Verwendung einer der von IBM MQunterstützten Anwendungsprogrammierschnittstellen. Diese Anwendungen können IBM MQ-Warteschlangen- oder Protokollnachrichten empfangen und auf deren Basis je nach Geschäftsanforderung entsprechend agieren.

Weitere Informationen zu den verfügbaren Anwendungsprogrammierschnittstellen finden Sie im Abschnitt Anwendungen entwickeln .

## MFT-Agentenwarteschlangenmanager konfigurieren

Führen Sie nach der Installation das Script *agent_name*_create.mqsc im Verzeichnis *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name* aus, um die erforderliche Konfiguration für den Agentenwarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell ausführen möchten, führen Sie diese Schritte auf dem Agenten-WS-Manager aus.

### Vorgehensweise

1. Erstellen Sie die Agentenoperationswarteschlangen.

   Diese Warteschlangen sind benannt:

   - SYSTEM.FTE.COMMAND.*Agentenname*
   - SYSTEM.FTE.DATA.*Agentenname*
   - SYSTEM.FTE.EVENT.*Agentenname*
   - SYSTEM.FTE.REPLY.*Agentenname*
   - SYSTEM.FTE.STATE.*Agentenname*

   Informationen zu den Warteschlangenparametern und zur Verwendung der Warteschlangen finden Sie im Abschnitt MFT -Agentenwarteschlangeneinstellungen.
2. Erstellen Sie die Agentenberechtigungswarteschlangen.

   Diese Warteschlangen sind benannt:

   - SYSTEM.FTE.AUTHADM1.*Agentenname*
   - SYSTEM.FTE.AUTHAGT1.*Agentenname*
   - SYSTEM.FTE.AUTHMON1.*Agentenname*
   - SYSTEM.FTE.AUTHOPS1.*Agentenname*
   - SYSTEM.FTE.AUTHSCH1.*Agentenname*
   - SYSTEM.FTE.AUTHTRN1.*Agentenname*

   Informationen zu den Warteschlangenparametern und zur Verwendung der Warteschlangen finden Sie im Abschnitt MFT -Agentenwarteschlangeneinstellungen.

### Nächste Schritte

Informationen zum Erstellen und Konfigurieren eines Protokollbridgeagenten finden Sie unter fteCreateBridgeAgent (einen MFT-Protokollbridgeagenten erstellen und konfigurieren) und Konfigurieren einer Protokollbridge für einen FTPS-Server.

**Zugehörige Konzepte**

„Verbindung zu IBM MQ herstellen” auf Seite 822
Die gesamte Netzkommunikation mit IBM MQ-Warteschlangenmanagern (dazu gehört auch die Kommunikation von Managed File Transfer) erfolgt über IBM MQ-Kanäle. Ein IBM MQ-Kanal stellt ein Ende einer Netzverbindung dar. Kanäle werden entweder als Nachrichtenkanal oder als MQI-Kanal klassifiziert.

„Multi-Instanz-Warteschlangenmanager für die Arbeit mit MFT konfigurieren” auf Seite 828

IBM WebSphere MQ 7.0.1 und höher unterstützt die Erstellung von Multi-Instanz-Warteschlangenmanagern. Ein WS-Manager mit mehreren Instanzen wird automatisch auf einem Standby-Server erneut gestartet. Managed File Transfer unterstützt Verbindungen zu Multi-Instanz-Agenten-, Multi-Instanz-Koordinations- und Multi-Instanz-Befehlswarteschlangenmanagern.

**Zugehörige Tasks**

„MFT-Netzwarteschlangenmanager konfigurieren" auf Seite 823
Sind in Ihrem Managed File Transfer-Netz mehrere IBM MQ-Warteschlangenmanager vorhanden, muss zwischen diesen IBM MQ-Warteschlangenmanagern eine Remotekommunikation möglich sein.

„Koordinationswarteschlangenmanager für MFT konfigurieren" auf Seite 824
Führen Sie nach der Ausführung des Befehls **fteSetupCoordination** das Script *coordination_qmgr_name*.mqsc im Verzeichnis *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* aus, um die erforderliche Konfiguration für den Koordinationswarteschlangenmanager auszuführen. Wenn Sie diese Konfiguration jedoch manuell durchführen möchten, gehen Sie für den Koordinations-Warteschlangenmanager wie im Folgenden beschrieben vor.

**Zugehörige Verweise**

Einstellungen von MFT-Agentenwarteschlangen
fteSetupCoordination

## MFT-Agenten für mehrere Kanäle in einem Cluster konfigurieren

Wenn Sie die IBM MQ -Mehrkanalunterstützung in einer Clusterkonfiguration verwenden wollen, setzen Sie zuerst die Eigenschaft **agentMultipleChannelsEnabled** auf `true` und führen Sie dann die Schritte in diesem Abschnitt aus.

### Informationen zu diesem Vorgang

In einem Cluster wird die Unterstützung für mehrere Kanäle nur über die IBM MQ-Definitionen im Warteschlangenmanager des Zielagenten aktiviert.

Zusätzlich zu den standardmäßigen IBM MQ-Konfigurationsschritten für einen Managed File Transfer-Agenten (siehe „MFT für erstmalige Verwendung konfigurieren" auf Seite 821) müssen Sie noch die hier beschriebenen Schritte ausführen.

Die nachfolgenden Konfigurationsbeispiele veranschaulichen die Verwendung der **runmqsc**-Befehle.

### Vorgehensweise

1. Definieren Sie für jeden Kanal, der verwendet werden soll, einen Clusterempfängerkanal. Beispiel bei Verwendung von zwei Kanälen:

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
   ```

   Dabei gilt:

   - *ZIEL_WS_MANAGER* steht für den Namen des Warteschlangenmanagers des Zielagenten.
   - *MFTCLUSTER* steht für den Namen des IBM MQ-Clusters.

   Es wird empfohlen, die Namenskonvention *MFTCLUSTER.ZIEL_WS_MANAGER*_n für Kanäle zu verwenden; diese Konvention ist jedoch nicht verbindlich.
2. Definieren Sie für jeden Kanal einen Warteschlangenmanager-Aliasnamen. For example:

   ```
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
   DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
   ```

   Sie müssen das SYSTEM.FTE.*DESTQMGRNAME_n* Namenskonvention für WS-Manager-Aliasnamen , da der sendende Agent nach WS-Manager-Aliasnamen dieses Formats sucht. Für *n* müssen jeweils

fortlaufende Nummern angegeben werden, die bei 1 beginnen. Die Definitionen müssen clusterweit erfolgen, damit sie auf dem Warteschlangenmanager des Quellenagenten verfügbar sind.

Damit Quellen- und Zielagent die Nummer der Warteschlangenmanager-Aliasnamen korrekt ermitteln kann, darf für den Warteschlangenmanager **keine** Standardübertragungswarteschlange (XMITQ) definiert werden.

**Zugehörige Tasks**

„MFT-Agenten für mehrere Kanäle konfigurieren: clusterunabhängig” auf Seite 834
Soll die IBM MQ-Unterstützung für mehrere Kanäle in einer Konfiguration aktiviert werden, bei der es sich nicht um einen Cluster handelt, müssen Sie zunächst die Eigenschaft 'agentMultipleChannelsEnabled' auf `true` setzen und anschließend die in diesem Abschnitt beschriebenen Schritte ausführen.

**Zugehörige Verweise**

Die MFT `agent.properties`-Datei

## MFT-Agenten für mehrere Kanäle konfigurieren: clusterunabhängig

Soll die IBM MQ-Unterstützung für mehrere Kanäle in einer Konfiguration aktiviert werden, bei der es sich nicht um einen Cluster handelt, müssen Sie zunächst die Eigenschaft 'agentMultipleChannelsEnabled' auf `true` setzen und anschließend die in diesem Abschnitt beschriebenen Schritte ausführen.

### Informationen zu diesem Vorgang

In einer Konfiguration, bei der es sich nicht um einen Cluster handelt, wird die Unterstützung mehrerer Kanäle über die IBM MQ-Definitionen im Warteschlangenmanager auf dem Quellen- und auf dem Zielagenten aktiviert.

Zusätzlich zu den standardmäßigen IBM MQ-Konfigurationsschritten für einen Managed File Transfer-Agenten (siehe „MFT für erstmalige Verwendung konfigurieren” auf Seite 821) müssen Sie noch die hier beschriebenen Schritte ausführen.

Bei den folgenden Schritten wird davon ausgegangen, dass die Kommunikation zwischen den Quellen- und Zielwarteschlangenmanagern über Sender-/Empfängerkanäle erfolgt.

Die nachfolgenden Konfigurationsbeispiele veranschaulichen die Verwendung der **runmqsc**-Befehle.

### Vorgehensweise

1. Definieren Sie im Warteschlangenmanager des Zielagenten für jeden Kanal, der verwendet werden soll, einen Empfängerkanal. Beispiel bei Verwendung von zwei Kanälen:

   ```
   DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
   DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
   ```

   Dabei ist DESTQMGRNAME der Name des Warteschlangenmanagers des Zielagenten.

   Für Kanalnamen empfiehlt sich die Verwendung des Formats TO.DESTMGRNAME_n, seine Verwendung ist jedoch nicht zwingend. Die Namen der Empfängerkanäle müssen den jeweiligen Senderkanälen des Warteschlangenmanagers des Quellenagenten entsprechen.
2. Definieren Sie im Warteschlangenmanager des Quellenagenten für jeden Kanal, der verwendet werden soll, eine Übertragungswarteschlange. Beispiel bei Verwendung von zwei Kanälen:

   ```
   DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
   DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
   ```

   Für Übertragungswarteschlangen empfiehlt sich die Verwendung des Formats DESTMGRNAME_n, seine Verwendung ist jedoch nicht zwingend. Auf die von Ihnen definierten Übertragungswarteschlangen wird in den folgenden Schritten von den Senderkanaldefinitionen und den Warteschlangenmanager-Aliasnamensdefinitionen verwiesen.

3. Definieren Sie im Warteschlangenmanager des Quellenagenten für jeden Kanal, der verwendet werden soll, einen Senderkanal. Beispiel bei Verwendung von zwei Kanälen:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_1)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_2)
```

   Für Kanäle empfiehlt sich die Verwendung des Formats TO.DESTMGRNAME_n, seine Verwendung ist jedoch nicht zwingend. Die Namen der Senderkanäle müssen den jeweiligen Empfängerkanälen im Warteschlangenmanager des Zielagenten entsprechen.
4. Definieren Sie im Warteschlangenmanager des Quellenagenten für jeden Kanal einen entsprechenden Warteschlangenmanager-Aliasnamen. For example:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

   Für Warteschlangenmanager-Aliasnamen muss SYSTEM.FTE.DESTQMGRNAME_n als Namensformat verwendet werden, da der sendende Agent nach Warteschlangenmanager-Aliasnamen dieses Formats sucht. Für *n* müssen jeweils fortlaufende Nummern angegeben werden, die bei 1 beginnen.

   Damit der Agent die Nummer der Warteschlangenmanager-Aliasnamen korrekt ermitteln kann, darf für den Warteschlangenmanager **keine** Standardübertragungswarteschlange (XMITQ) definiert werden.

**Zugehörige Tasks**

„MFT-Agenten für mehrere Kanäle in einem Cluster konfigurieren” auf Seite 833
Wenn Sie die IBM MQ -Mehrkanalunterstützung in einer Clusterkonfiguration verwenden wollen, setzen Sie zuerst die Eigenschaft **agentMultipleChannelsEnabled** auf `true` und führen Sie dann die Schritte in diesem Abschnitt aus.

**Zugehörige Verweise**

Die MFT `agent.properties`-Datei

## Windows MFT-Agenten mit MSCS konfigurieren

Die Konfiguration von Managed File Transfer Agent Microsoft Cluster Service (MSCS) wird unterstützt, wenn die Plattform von MFT unterstützt wird und eine der Versionen von Windowsausgeführt wird.

### Informationen zu diesem Vorgang

In dieser Task werden zwei Szenarios beschrieben, mit denen Sie eine Funktionsübernahme eines MFT-Agenten erreichen:

- Szenario 1: Agent als MSCS-Ressource konfigurieren.
- Szenario 2: Agentenwarteschlangenmanager und Agent als MSCS-Ressourcen konfigurieren.

### Prozedur

Szenario 1: Agent als MSCS-Ressource konfigurieren

- Gehen Sie folgendermaßen vor, um den Agenten als eine MSCS-Ressource zu konfigurieren:
  a) Installieren Sie Managed File Transfer lokal auf jedem System im Cluster.

     Weitere Informationen finden Sie unter Managed File Transfer installieren.
  b) Erstellen Sie den Agenten auf dem primären System im Cluster.

     Der Agent sollte so konfiguriert sein, dass er eine Verbindung zum Agentenwarteschlangenmanager mit dem CLIENT-Transport herstellen kann. Stellen Sie sicher, dass Sie alle Objekte im Warteschlangenmanager für diesen Agenten erstellen. Informationen zur Vorgehensweise finden Sie in Agenten einrichten.

c) Ändern Sie den Agenten so, dass er als Windows-Service ausgeführt wird, und konfigurieren Sie ihn so, dass er bei einem Neustart von Windows nicht automatisch gestartet wird, indem Sie das Feld **Initialisierungstyp** für den Agentenservice im Tool mit den Windows-Services auf `Manuell` setzen.

Weitere Informationen finden Sie im Abschnitt MFT-Agent als Windows-Dienst starten.

d) Wiederholen Sie Schritt „2" auf Seite 835 und Schritt „3" auf Seite 836 von Szenario 1 im sekundären System.

Auf diese Weise wird sichergestellt, dass die Dateistruktur für Protokolle, Eigenschaften usw. im anderen System im Cluster vorhanden ist. Beachten Sie, dass die Warteschlangenmanagerobjekte nicht wie in Schritt „2" auf Seite 835 erstellt werden müssen.

e) Fügen Sie im primären System den Agenten in der MSCS-Steuerung als 'Generic Service' (Allgemeiner Service) hinzu.

Gehen Sie dazu wie folgt vor:

a. Klicken Sie mit der rechten Maustaste auf den Cluster und wählen Sie **Role -> Add Resource -> 'Generic Service'** (Rolle > Ressourcen hinzufügen > Allgemeiner Service) aus.

b. Wählen Sie in der Liste der Windows-Services den Agentenservice aus und schließen Sie den Konfigurationsassistenten durch Klicken auf **Weiter** ab.

Der Agentenservice ist jetzt als MSCS-Ressource hinzugefügt. Bei einer Funktionsübernahme wird der Agentenservice auf dem anderen System gestartet.

Szenario 2: Agentenwarteschlangenmanager und Agent als MSCS-Ressourcen konfigurieren

- Führen Sie die folgenden Schritte aus, um den Agentenwarteschlangenmanager und den Agenten als MSCS-Ressource zu konfigurieren:

a) Konfigurieren Sie den Agentenwarteschlangenmanager so, dass er als MSCS-Ressourcen ausgeführt wird.

Informationen zur Vorgehensweise finden Sie in „WS-Manager unter MSCS-Steuerung einschalten" auf Seite 533.

b) Erstellen Sie den Agenten auf dem primären System im Cluster.

Der Agent sollte so konfiguriert sein, dass er eine Verbindung zum Agentenwarteschlangenmanager mit dem BINDINGS-Transport herstellen kann. Stellen Sie sicher, dass Sie alle Objekte im Warteschlangenmanager für diesen Agenten erstellen. Informationen zur Vorgehensweise finden Sie in Agenten einrichten.

c) Ändern Sie den Agenten so, dass er als Windows-Service ausgeführt wird, und konfigurieren Sie ihn so, dass er bei einem Neustart von Windows nicht automatisch gestartet wird, indem Sie das Feld **Initialisierungstyp** für den Agentenservice im Tool mit den Windows-Services auf `Manuell` setzen.

Weitere Informationen finden Sie im Abschnitt MFT-Agent als Windows-Dienst starten.

d) Stellen Sie sicher, dass der Agentenwarteschlangenmanager (unter MSCS-Steuerung) auf dem sekundären System ausgeführt wird.

Der Agent, der auf diesem System erstellt wurde, stellt eine Verbindung zum Warteschlangenmanager über den BINDINGS-Transport her und muss daher verfügbar sein, wenn der Agent erstellt wird.

e) Wiederholen Sie Schritt „2" auf Seite 836 und Schritt „3" auf Seite 836 von Szenario 2 im sekundären System.

Auf diese Weise wird sichergestellt, dass die Dateistruktur für Protokolle, Eigenschaften usw. im anderen System im Cluster vorhanden ist. Beachten Sie, dass die Warteschlangenmanagerobjekte nicht wie in Schritt „2" auf Seite 836 erstellt werden müssen.

f) Fügen Sie den Agenten in der MSCS-Steuerung als 'Generic Service' (Allgemeiner Service) hinzu.

Gehen Sie dazu wie folgt vor:

a. Klicken Sie mit der rechten Maustaste auf den Cluster und wählen Sie **Role -> Add Resource -> 'Generic Service'** (Rolle > Ressourcen hinzufügen > Allgemeiner Service) aus.

b. Wählen Sie in der Liste der Windows-Services den Agentenservice aus und schließen Sie den Konfigurationsassistenten durch Klicken auf **Weiter** ab.

g) Ändern Sie die Ressourceneigenschaften des Agentenservice, um die Warteschlangenmanagerressource in der Liste mit den Abhängigkeiten hinzuzufügen.

   Dadurch wird sichergestellt, dass die Warteschlangenmanagerressource vor dem Agenten gestartet wird.

h) Ändern Sie den Status der Warteschlangenmanagerressource in offline und den Status der Agentenressource in online. Stellen Sie sicher, dass die Warteschlangenmanagerressource und der Agent gestartet wurden.

   Bei einer Funktionsübernahme werden der Agentenservice und der Agentenwarteschlangenmanager auf dem sekundären System gestartet.

## Hochverfügbarkeitsagenten in Managed File Transfer

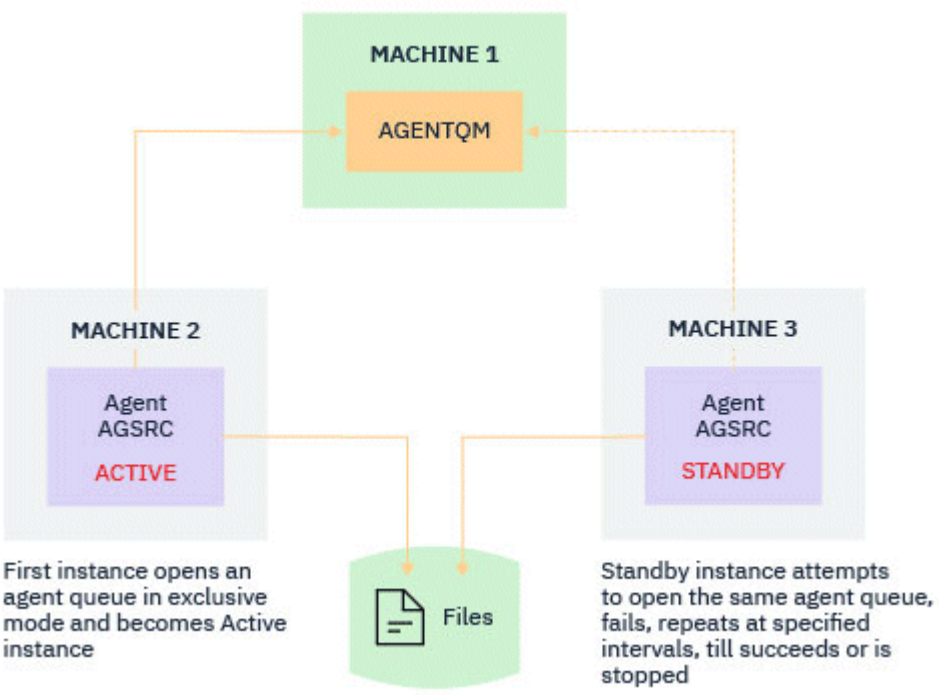
Sie können Standard- oder Bridgeagenten in MFT so konfigurieren, dass sie in einer Hochverfügbarkeitskonfiguration (HA-Konfiguration) ausgeführt werden. An einer HA-Konfiguration ist ein Paar von Agenteninstanzen mit identischen Konfigurationen beteiligt, bei dem die Instanzen auf verschiedenen Maschinen ausgeführt werden. Beide Instanzen sind so konfiguriert, dass sie eine Verbindung zum gleichen Agentenwarteschlangenmanager herstellen.

### Übersicht

Nur eine der beiden Instanzen, die *aktive Instanz*, verarbeitet Dateiübertragungen, während sich die andere Instanz, die *Standby-Instanz*, in einem partiell initialisierten Status befindet und keine Dateiübertragungen verarbeiten kann.

Wenn eine aktive Instanz fehlschlägt oder die Verbindung zum Warteschlangenmanager unterbrochen wird, beendet die Standby-Instanz die Initialisierung, wird aktiv und startet mit der Verarbeitung von Dateiübertragungen. Alle unvollständigen Übertragungen, die beim Fehlschlagen der aktiven Instanz in Bearbeitung waren, werden ab dem letzten bekannten Prüfpunkt fortgesetzt.

In der folgenden Abbildung wird eine allgemeine Konfiguration von aktiven Agenten und Standby-Agenten




gezeigt:

**Anmerkungen:**

1. Eine Instanz eines Agenten wird auf zwei verschiedenen Maschinen ausgeführt, wobei eine der Instanzen als *aktive Instanz* und die andere als *Standby-Instanz* verwendet wird.

2. Jede Instanz des Agenten wird auf einer anderen Maschine ausgeführt, wobei eine der Instanzen als aktive Instanz und die andere als Standby-Instanz verwendet wird.
3. Die gleiche Gruppe von Agentenwarteschlangen wird von beiden Instanzen des Agenten gemeinsam genutzt.
4. Beide Instanzen des Agenten benötigen Zugriff auf dasselbe gemeinsam genutzte Dateisystem, um verwaltete Übertragungen ausführen zu können.

Die Instanz des aktiven Standby-Agenten setzt eine Sperre für eine gemeinsam genutzte Ressource. Die Agenteninstanz, die eine Sperre in den gemeinsam genutzten Ressourcen setzt, wird zur aktiven Instanz, während die andere Instanz (die keine Sperre setzten kann) zur Standby-Instanz wird.

Die hier gemeinsam genutzte Ressource ist eine neue Warteschlange, SYSTEM.FTE.HA.<agent name>. Diese Warteschlange wird automatisch erstellt, wenn ein Agent für IBM MQ 9.1.4 oder höher konfiguriert wird.

## Funktionsweise des Prozesses

Zum Erstellen eines HA-Agenten erstellen Sie eine Agenten mit identischen Konfigurationsparametern auf zwei Maschinen, indem Sie den Befehl **fteCreateAgent** oder **fteCreateBridgeAgent** und den zusätzlichen Parameter **-x** zusammen mit der Agenteneigenschaft **highlyAvailable** ausführen, die in der Datei agent.properties auf true gesetzt ist.

**Anmerkungen:**

- Beide Konfigurationen müssen auf den gleichen Agentenwarteschlangenmanager verweisen.
- Die erforderlichen Agentenwarteschlangen müssen nur einmal im Agentenwarteschlangenmanager erstellt werden.

Weitere Informationen zum Parameter **-x** und zur Datei agent.properties finden Sie in der Beschreibung des Befehls **fteCreateAgent** . Weitere Informationen zur Agenteneigenschaft **highlyAvailable** finden Sie in der Beschreibung des Befehls.

**Anmerkung:** Bei Ausführung des Befehls **fteCreateAgent** oder **fteCreateBridgeAgent** wird eine MQSC-Datei mit den Scripts erstellt, die zum Erstellen von IBM MQ -Objekten im Agentenwarteschlangenmanager und in der SYSTEM.FTE.HA.*agent name* -Warteschlange erforderlich sind. Diese MQSC-Datei wird unabhängig davon erstellt, ob Sie den Parameter **-x** angeben.

Beim Erstellen einer hoch verfügbaren Agentenkonfiguration überprüft der Befehl **fteCreateAgent** oder **fteCreateBridgeAgent** das Vorhandensein einer Instanz desselben Agenten, die an anderer Stelle vorhanden ist, indem er das Topic SYSTEM.FTE/Agents/*agent name* subskribiert. Wenn eine Instanz des gleichen Agenten gefunden wird, wird mit einem der Befehle die erforderliche Konfiguration im Dateisystem erstellt, aber die Erstellung des Agenten wird nicht erneut veröffentlicht.

Beim Starten eines Agenten im HA-Modus werden die folgenden Aktionen ausgeführt:

1. Der Agent versucht, die SYSTEM.FTE.HA.*agent name*-Warteschlange in einem exklusiven GET-Modus zu öffnen.
2. Wenn der Agent die SYSTEM.FTE.HA.*agent name*-Warteschlange erfolgreich öffnet, wird dies zur *aktiven Instanz* eines Agenten, und ein weiterer Startprozess wird fortgesetzt.
3. Wenn der Versuch, die SYSTEM.FTE.HA.*agent name* -Warteschlange in einem exklusiven GET-Modus zu öffnen, mit dem Ursachencode MQRC_OBJECT_IN_USE fehlschlägt, bedeutet dies, dass bereits eine aktive Instanz des Agenten an anderer Stelle ausgeführt wird. Dadurch wird diese Instanz zur *Standby-Instanz* des Agenten.

   Die Standby-Instanz versucht, die Warteschlange SYSTEM.FTE.HA.*agent name* in angegebenen Intervallen zu öffnen. Zu diesem Zweck wird die zusätzliche Agenteneigenschaft **standbyPollInterval** in der Datei agent.properties bereitgestellt.

   Mit dem Standardwert versucht die Standby-Instanz, die SYSTEM.FTE.HA.*agent name*-Warteschlange alle fünf Sekunden zu öffnen. Dieser Vorgang wird wiederholt, bis die Instanz die Warte-

schlange SYSTEM.FTE.HA.*agent name* erfolgreich öffnet oder mit dem Befehl **fteStopAgent** gestoppt wird.

Ab IBM MQ 9.2.4 und IBM MQ 9.2.0 Fix Pack 5wird die Eigenschaft **standbyPollInterval** auch von allen Instanzen verwendet, um festzulegen, wie lange eine Instanz zwischen Verbindungswiederholungen wartet, wenn sie von ihrem Agentenwarteschlangenmanager getrennt wird.

## Mehrere Standby-Instanzen

Alle Standby-Instanzen versuchen, die SYSTEM.FTE.HA.*agent name*-Warteschlange in einem exklusiven GET-Modus zu verwenden, und die Instanz, die nach dem Fehlschlagen der aktiven Instanz erfolgreich ist, wird zum aktiven Exemplar.

Die aktive Instanz verwaltet Informationen aller bekannten Standby-Instanzen und veröffentlicht die Informationen als Teil der Veröffentlichungen zum Agentenstatus. Die Ausgabe des Befehls **fteShowAgentDetails**, der Antwort des Agenten GET REST API und des IBM MQ Explorer MFT-Plug-ins zeigen Informationen zu allen Standby-Instanzen an.

Weitere Informationen finden Sie in den Beispielausgaben des Befehls **fteShowAgentDetails** und der GET-Antwort der REST API für den Agenten.

Im Abschnitt Statusnachrichten für den MFT-Agenten finden Sie Beispiele für Statusinformationen zum Agenten im XML-Format.

## Versionsanforderung

Die aktiven Agenten und Standby-Agenten müssen in IBM MQ 9.1.4 oder höher verwendet werden.

**Achtung:**

- Versionen von IBM MQ vor IBM MQ 9.1.4 können nicht in einem Hochverfügbarkeitsmodus konfiguriert oder gestartet werden.
- Die aktive Instanz und die Standby-Instanz müssen mit der gleichen Version des Codes ausgeführt werden.

Die Version der aktiven und der Standby-Instanz werden geprüft, um sicherzustellen, dass beide Instanzen dieselbe Version haben. Für die Kommunikation zwischen den Instanzen wird eine temporäre dynamische Warteschlange verwendet. Der Name der temporären dynamischen Warteschlange ergibt sich aus zwei Agenteneigenschaften (**dynamicQueuePrefix** und **modelQueueName**), die in der Datei `agent.properties` definiert sind.

## *Erforderliche Informationen zu Hochverfügbarkeitsagenten in Managed File Transfer*

Es gibt verschiedene Arten von Informationen, die Sie über MFT-Standardagenten oder -Bridgeagenten, die in einer Hochverfügbarkeitskonfiguration aktiv sind, wissen müssen. Zu diesen Informationen gehören die unterschiedlichen Methoden zum Starten des Agenten, die Vorgehensweise zum Ermitteln der Instanz des Agenten in der Protokolldatei und die Statusinformationen zum Agenten.

## Agent starten

### Eine Instanz eines Agenten wird in einem Nicht-HA-Modus an anderer Stelle ausgeführt

Wenn versucht wird, eine weitere Instanz des Agenten zu starten, die nicht als HA-Agent konfiguriert ist, wird zunächst geprüft, ob eine Sperre für die Warteschlange SYSTEM.FTE.HA.*agent name* angefordert werden kann.

Da die andere Instanz im Nicht-HA-Modus gestartet wurde, wird die Sperre für die Warteschlange SYSTEM.FTE.HA.*agent name* von dieser Instanz angefordert. Der Agent setzt die Initialisierung fort, schlägt aber zu einem späteren Zeitpunkt fehl, weil die Befehlswarteschlange exklusiv von der anderen Instanz geöffnet ist.

In diesem Fall werden die Nachrichten, die im folgenden Beispiel angezeigt werden, in der `output0.log`-Datei des Agenten protokolliert und der Agent setzt den Versuch fort, die Befehlswarteschlange alle 30 Sekunden zu öffnen:

```
BFGMQ1045I: Agent's system queue 'SYSTEM.FTE.COMMAND.SRC' is configured as either NOSHARE or
DEFSOPT (GEMEINSAM genutzt).

BFGAG0035W: The agent received MQI reason code 2042 when trying to open queue
'SYSTEM.FTE.COMMAND.SRC' on the queue manager 'MFTHAQM' with connection name 'localhost(1414)'
and channel 'MFT_HA_CHN'.  The agent will try the operation again every 30 seconds.
```

#### Eine Instanz eines Agenten wird in einem HA-Modus an anderer Stelle ausgeführt

Wenn versucht wird, eine weitere Instanz des Agenten zu starten, die nicht als HA-Agent konfiguriert ist, wird zunächst geprüft, ob eine Sperre für die Warteschlange SYSTEM.FTE.HA.*agent name* angefordert werden kann.

Da die andere Instanz als aktive Instanz ausgeführt wurde, schlägt der Versuch zum Anfordern einer Sperre fehl. Die Instanz kann nicht gestartet werden, und die folgende Fehlernachricht wird in der output0.log-Datei des Agenten protokolliert:

```
BFGAG0194E: Eine Instanz dieses Agenten ist bereits woanders aktiv.
Daher kann diese Instanz nicht fortgesetzt werden und wird beendet.
```

#### Windows Agenten als Windows-Service starten

In Windows können Sie einen Agenten als Windows-Service starten.

Während des Starts startet Windows den MFT -Agenten im normalen Modus oder im Hochverfügbarkeitsmodus. Wenn der Agent für die Ausführung im HA-Modus konfiguriert ist, wird der Service als aktive Instanz oder als Standby-Instanz ausgeführt, abhängig davon, welche Instanz zuerst die Sperre anfordert.

## Instanztyp eines Agenten in der Protokolldatei angeben

Informationsnachrichten werden in die output0.log-Datei des Agenten geschrieben, um den Typ der Instanz anzugeben. Wenn eine Agenteninstanz als aktive Instanz gestartet wird, wird die folgende Nachricht in die Datei geschrieben:

```
BFGAG0193I: The agent has successfully initialized as an active instance.
```

Wenn eine Agenteninstanz als Standby-Instanz gestartet wird, wird die folgende Nachricht geschrieben:

```
BFGAG0193I: The agent has successfully initialized as a standby instance.
```

## Aktualisierungen des Agentenstatus

Da zwei Instanzen des gleichen Agenten ausgeführt werden, benötigen Sie Informationen zu beiden Instanzen in der Veröffentlichung des Agentenstatus.

Beachten Sie, dass die aktive Instanz den Status beider Instanzen veröffentlicht.

#### Standby Instance

Bei der Veröffentlichung des Agentenstatus überprüft die aktive Instanz das Alter der Veröffentlichung für die Standby-Instanz.

Zu diesem Zweck gibt es in der agent.properties-Datei zwei weitere Eigenschaften:

- **standbyStatusExpiry** ist die Ablaufzeit für die Standby-Statusnachricht, die in die Befehlswarteschlange des Agenten eingereiht werden soll. Die Nachricht läuft ab, wenn die aktive Instanz eines Agenten diese Nachricht nicht in diesem Zeitraum verarbeitet.

  Der Standardwert für **standbyStatusExipry** ist 30 Sekunden. Die Nachricht ist auch eine Nachricht mit niedriger Priorität (9), um die Prioritätsverarbeitung von Übertragungsanforderungen über Standby-Statusnachrichten zu ermöglichen.
- **standbyStatusPublishInterval** legt die Häufigkeit fest, mit der die Standby-Instanz ihren Status veröffentlicht.

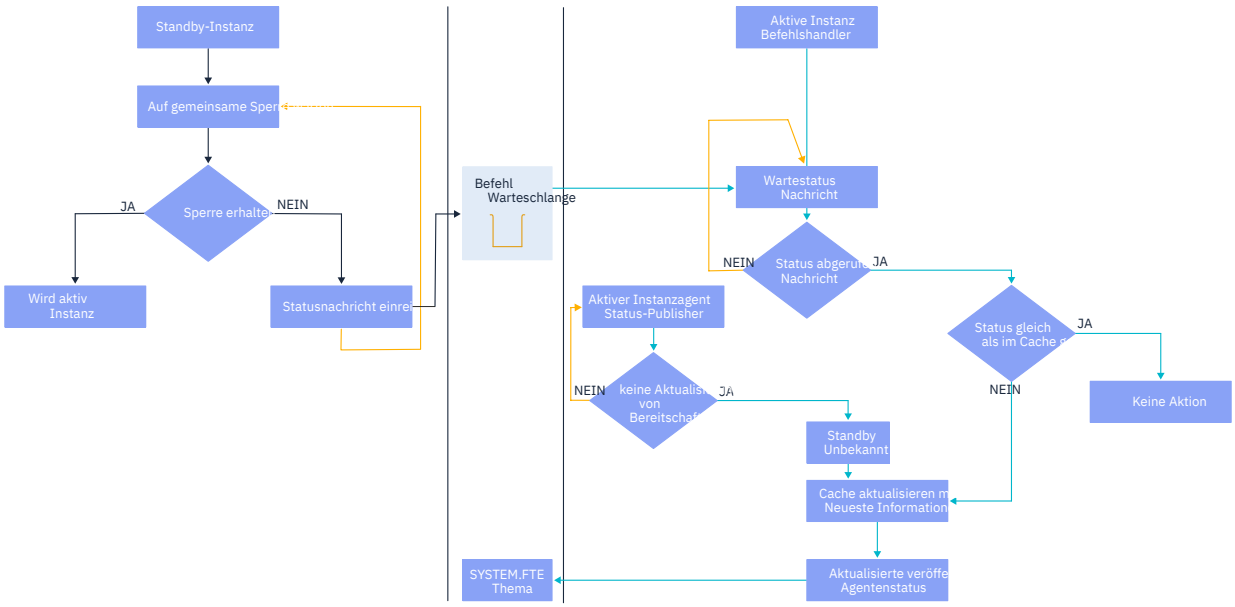
#### Aktive Instanz

Die aktive Instanz führt die folgenden Schritte aus, um Statusaktualisierungen von der Standby-Instanz zu verarbeiten:

1. Ruft die Nachricht aus der SYSTEM.FTE.COMMAND.<agent name>-Warteschlange ab und delegiert die Nachrichtenverarbeitung an einen Worker-Thread.
2. Der Arbeitsthread ruft die Inhalte aus dem Nachrichtenhauptteil ab, aktualisiert das Agentenstatusobjekt mit den Informationen der Standby-Instanz und informiert den Bereitsteller des Agentenstatus darüber, dass der Status veröffentlicht werden soll.
3. Der Bereitsteller der Agentenstatus veröffentlicht den Status.

   Beachten Sie, dass hier Optimierungen ausgeführt werden, um die Informationen zum Standby-Status zwischenspeichern zu können. Wenn eine Anforderung gestellt wird, überprüft der Bereitsteller des Agentenstatus den neuen Status mit dem zwischengespeicherten Status und veröffentlicht nur, wenn es einen Unterschied gibt.

Im folgenden Diagramm wird der Ablauf der aktiven Instanzen oder der Standby-Instanzen nach der Veröffentlichung des Status eines Agenten gezeigt:




## *Instanzen, Funktionsübernahme und Verwaltung in Hochverfügbarkeitsagenten löschen*

Hoch verfügbare Managed File Transfer-Instanzen können gelöscht werden, können auf verschiedene Arten fehlschlagen und müssen möglicherweise gewartet werden.

### Status der Standby-Instanz löschen

Es kann Situationen geben, in denen die aktive Instanz mit Übertragungen beschäftigt ist und die Statusnachrichten einer Standby-Instanz nicht verarbeiten kann, oder in denen die Standby-Instanz fehlgeschlagen ist oder aus einem beliebigen Grund keine Statusnachrichten veröffentlicht.

In solchen Szenarios wartet der aktive Agent, der das Vorhandensein einer Standby-Instanz erkannt hat, auf den Wert, der durch die Eigenschaft **standbyStatusDiscardTime** in der Datei agent.properties angegeben ist, bevor die Standby-Instanz aus der Liste entfernt wird. Der Standardwert für diese Eigenschaft beträgt 600 Sekunden und ist damit doppelt so groß wie die Eigenschaft **standbyStatusPublishInterval**.

## Normale Funktionsübernahme einer Instanz

Für die Ausführung einer normale Funktionsübernahme verwenden Sie den Befehl **fteStopAgent** mit der Option **-i**.

Dadurch wird sichergestellt, dass die aktive Instanz sofort gestoppt wird. Wenn Sie einen Agenten ohne die Option **-i** stoppen, wird der Agent weiter ausgeführt, bis alle laufenden Übertragungen durch die aktive Instanz abgeschlossen wurden. Dadurch kann es einige Zeit dauern, bis die Funktionsübernahme ausgeführt werden kann.

Alle unvollständigen Übertragungen werden vom letzten bekannten Prüfpunkt fortgesetzt.

## Funktionsübernahme einer Instanz in anderen Situationen

Wenn eine aktive Instanz auf eine Weise beendet wird, die nicht normal ist, oder wenn die gesamte Maschine ausfällt, wird die Verbindung zur Agentenwarteschlange unterbrochen und der Warteschlangenmanager schließt alle offenen Warteschlangen, einschließlich der SYSTEM.FTE.HA.<agent name> -Warteschlange, und Verbindungen.

Deshalb fordert die Standby-Instanz den exklusiven GET-Modus an und schließt den Rest der Agenteninitialisierung ab.

Auch hier werden alle unvollständigen Übertragungen von den letzten bekannten Prüfpunkten fortgesetzt.

## Verbindung zu einem Warteschlangenmanager wird unterbrochen

### Clientmodus

Ein Agentenprozess besteht aus mehreren Threads. Mit Ausnahme von Standardthreads, wo ein Thread beispielsweise den Agentenstatus in regelmäßigen Intervallen veröffentlicht, wird jede Übertragungsanforderung von einer Gruppe von Threads verarbeitet, die nach Abschluss einer Übertragung beendet werden.

Viele dieser Threads stellen eine Verbindung zum Agentenwarteschlangenmanager her und führen das Einreihen und Abrufen von Nachrichten aus. Diese Verbindungen können aufgrund eines Netzproblems oder beim Fehlschlagen eines Warteschlangenmanagers unterbrochen werden. Wenn ein Thread ein Problem aufgrund einer unterbrochenen Verbindung erkennt, informiert er den Hauptthread, damit die Wiederherstellung eingeleitet werden kann, und wird beendet.

Der Hauptthread startet anschließend einen weiteren Thread, der darauf wartet, dass eine Verbindung zum Warteschlangenmanager hergestellt wird. Sobald die Verbindung wiederhergestellt ist, wird versucht, den exklusiven GET-Modus für den Agenten anzufordern. Ist dieser Vorgang erfolgreich, setzt der Agent den Versuch der Wiederherstellung fort und wird zur aktiven Instanz. Wenn der Versuch, den exklusiven GET-Modus anzufordern, fehlschlägt, wird die Instanz zur Standby-Instanz.

### Bindungsmodus

Wenn eine Verbindung im Bindungsmodus hergestellt ist, wird der Agentenprozess beendet, falls die Verbindung zu einem Agenten unterbrochen wird. Der Prozesscontroller verarbeitet den Neustart des Agenten. Wenn ein Agent erneut gestartet wird, versucht er, den exklusiven GET-Modus für sich anzufordern.

Ist er erfolgreich, wird der Agent die aktive Instanz; andernfalls wird er zur Standby-Instanz.

## Aktualisierungen der Wartungsstufe anwenden

Die Schritte zum Anwenden der Wartung für Hochverfügbarkeitsagenten sind weitgehend mit denen identisch, die für Multi-Instanz-Warteschlangenmanager dokumentiert sind. Weitere Informationen finden Sie unter Wartungsstufenaktualisierungen für Multi-Instanz-Warteschlangenmanager unter Windows oder Wartungsstufenaktualisierungen für Multi-Instanz-Warteschlangenmanager unter AIXanwendenoder Wartungsstufenaktualisierungen für Multi-Instanz-Warteschlangenmanager unter Linuxanwenden.

Bevor Sie die Wartung anwenden, müssen Sie den Agenten stoppen, der auf der Maschine ausgeführt wird, auf der die Wartungsstufe angewendet werden soll. Wenn Sie eine aktive Instanz aktualisieren, müssen Sie zur Gewährleistung der Kontinuität von Übertragungen eine Funktionsübernahme der aktiven Instanz in eine Standby-Instanz vornehmen.

Nach Abschluss der Aktualisierung müssen Sie die Agenteninstanz starten, eine Funktionsübernahme der aktuell aktiven Instanz in die aktualisierte Instanz vornehmen und anschließend die Standby-Instanz aktualisieren.

### Agenten von einer früheren Version des Produkts migrieren

Agenten, die von IBM MQ-Versionen vor IBM MQ 9.1.4 migriert wurden, werden nicht als Hochverfügbarkeitsagenten ausgeführt. Sie können Sie als Hochverfügbarkeitsagenten ausführen, indem Sie den Anweisungen im Abschnitt Managed File Transfer-Agenten von einer früheren Version migrieren folgen.

## MFT-Protokollfunktion konfigurieren

Bei der Übertragung von Dateien veröffentlicht Managed File Transfer im Koordinationswarteschlangenmanager Informationen zu den einzelnen Aktionen. Die Datenbankprotokollfunktion ist eine optionale Komponente von Managed File Transfer, mit der Sie diese Informationen zu Analyse- und Prüfzwecken in eine Datenbank kopieren können.

Es gibt drei Versionen des Loggers:

- ALW Eigenständige Dateiprotokollfunktion
- Eigenständige Datenbankprotokollfunktion
- Java Platform, Enterprise Edition-Protokollfunktion (Java EE)

### Protokollfunktionen unter IBM i

IBM i

Managed File Transfer-Protokollfunktionen werden auf der IBM i-Plattform nicht unterstützt.

### Eigenständige Dateiprotokollfunktion

ALW

Die eigenständige Dateiprotokollfunktion ist ein Java-Prozess, der entweder auf dem System mit dem Koordinationswarteschlangenmanager ausgeführt wird oder auf einem System mit einem Warteschlangenmanager, der Verbindung zum Koordinationswarteschlangenmanager hat. Dabei verwendet die eigenständige Dateiprotokollfunktion IBM MQ-Bindungen zur Verbindung mit dem zugehörigen Warteschlangenmanager. Die eigenständige Protokollfunktion wird mit dem Befehl **`fteCreateLogger`** erstellt.

Windows Die eigenständige Dateiprotokollfunktion kann als Windows-Dienst ausgeführt werden; so wird sichergestellt, dass die Dateiprotokollfunktion auch nach Ihrer Abmeldung von der Windows-Sitzung weiterhin aktiv ist; außerdem kann sie so konfiguriert werden, dass sie bei einem Neustart des Systems automatisch gestartet wird. Weitere Informationen finden Sie unter „Eigenständige MFT-Dateiprotokollfunktion installieren” auf Seite 844.

Die eigenständige Dateiprotokollfunktion wird auf den folgenden Plattformen nicht unterstützt:

- z/OS z/OS
- IBM i IBM i

### Eigenständige Datenbankprotokollfunktion

Die eigenständige Datenbankprotokollfunktion ist eine Java-Anwendung, die Sie auf einem System installieren, auf dem sich ein Warteschlangenmanager und eine Datenbank befinden. Die eigenständige Daten-

bankprotokollfunktion wird häufig auf demselben System wie der Koordinationswarteschlangenmanager installiert. Sie kann jedoch auch auf demselben System wie jeder Warteschlangenmanager installiert werden, der über eine Verbindung zum Koordinations-WS-Manager verfügt. Die eigenständige Datenbankprotokollfunktion stellt die Verbindung zum zugeordneten Warteschlangenmanager über IBM MQ-Bindungen her; die Verbindung zur einer Db2- oder Oracle-Datenbank erfolgt über einen JDBC-Treiber des Typs 2 oder 4. Diese Verbindungstypen sind erforderlich, da die Protokollfunktion der eigenständigen Datenbank die XA-Unterstützung des WS-Managers verwendet, um eine globale Transaktion sowohl über den Warteschlangenmanager als auch über die Datenbank zu koordinieren und die Daten zu schützen.

Windows Bei Verwendung eines Windows-Systems können Sie die eigenständigen Protokollfunktionen auch als Windows-Dienste ausführen; damit wird sichergestellt, dass die Protokollfunktionen auch nach Ihrer Abmeldung von der Windows-Sitzung weiterhin aktiv sind. Weitere Informationen finden Sie unter „Eigenständige MFT-Datenbankprotokollfunktion installieren” auf Seite 852 für eine eigenständige Datenbankprotokollfunktion.

### Java EE-Datenbankprotokollfunktion

Die Java EE-Datenbankprotokollfunktion wird als EAR-Datei bereitgestellt, die auf einem Anwendungsserver installiert wird. Wenn eine Java EE-Anwendungsserverumgebung vorhanden ist, ist diese Protokollfunktion unter Umständen komfortabler als die eigenständige Datenbankprotokollfunktion, da die Java EE-Datenbankprotokollfunktion zusammen mit den anderen Unternehmensanwendungen verwaltet werden kann. Sie können die Java EE-Datenbankprotokollfunktion auch getrennt von den Systemen, auf denen sich der IBM MQ-Server und die zugehörige Datenbank befinden, auf einem anderen System installieren. Die Java EE-Datenbankprotokollfunktion wird für Db2- und Oracle-Datenbanken unterstützt. Die Java EE-Datenbankprotokollfunktion unterstützt auch Oracle Real Application Clusters, wenn sie unter WebSphere Application Server 7.0 installiert ist.

Anweisungen zum Konfigurieren einer Protokollfunktion finden Sie in den folgenden Abschnitten:

- „Eigenständige MFT-Dateiprotokollfunktion installieren” auf Seite 844
- „Eigenständige MFT-Datenbankprotokollfunktion installieren” auf Seite 852
- „Java EE-Datenbankprotokollfunktion für MFT installieren” auf Seite 857

**Zugehörige Tasks**

„MFT mit einer fernen Datenbank verwenden” auf Seite 854
Mithilfe der Managed File Transfer-Protokollfunktion können Sie mit einer Datenbank auf einem fernen System kommunizieren.

**Zugehörige Verweise**

Fehlerbehandlung und Zurückweisung von Nachrichten der MFT -Protokollfunktion
Konfigurationseigenschaften der MFT-Protokollfunktion

## ALW Eigenständige MFT-Dateiprotokollfunktion installieren

Die eigenständige Dateiprotokollfunktion ist ein Java -Prozess, der im IBM MQ -Bindungsmodus oder -Clientmodus eine Verbindung zu einem Koordinationswarteschlangenmanager herstellen muss. Verwenden Sie zum Definieren einer eigenständigen Dateiprotokollfunktion den Befehl **`fteCreateLogger`** und führen Sie die in diesem Abschnitt beschriebenen Schritte aus.

### Informationen zu diesem Vorgang

Weitere Informationen zur eigenständigen Dateiprotokollfunktion finden Sie im Abschnitt „MFT-Protokollfunktion konfigurieren” auf Seite 843. Die Schritte in diesem Thema konfigurieren eine Protokollfunktion, um eine Verbindung zu einem Koordinations-WS-Manager herzustellen. Beschreibungen alternativer Protokollfunktionskonfigurationen finden Sie im Abschnitt „Alternative Konfigurationen für eine eigenständige MFT-Protokollfunktion” auf Seite 856

Die eigenständige Dateiprotokollfunktion wird auf den folgenden Plattformen nicht unterstützt:

- z/OS z/OS

- IBM i IBM i

## Vorgehensweise

1. Stellen Sie sicher, dass die Komponente Managed File Transfer Logger installiert ist. Weitere Informationen finden Sie unter Produktoptionen für Managed File Transfer .
2. Führen Sie den Befehl **fteCreateLogger** unter Angabe des Koordinationswarteschlangenmanagers aus und setzen Sie den Parameter -loggerType auf FILE , um Ihre eigenständige Dateiprotokollfunktion zu erstellen. Weitere Informationen finden Sie in fteCreateLogger .
3. Optional: Wenn Sie ein angepasstes Format verwenden möchten, können Sie die vom Befehl **fteCreateLogger** erstellte XML-Datei ändern. Die Protokollformatdefinition befindet sich in der Datei FileLoggerFormat.xml . Weitere Informationen finden Sie unter „Format der eigenständigen MFT-Dateiprotokollfunktion” auf Seite 846.
4. Führen Sie die mit dem Befehl **fteCreateLogger** bereitgestellten MQSC-Befehle für Ihren Koordinationswarteschlangenmanager aus, um die Warteschlangen der Protokollfunktion zu erstellen.
5. Identifizieren Sie einen Benutzer, um den Protokollfunktionsprozess auszuführen und Berechtigungen für diesen Benutzer zu konfigurieren. Weitere Informationen finden Sie unter „Benutzerzugriff für eine eigenständige MFT-Dateiprotokollfunktion konfigurieren” auf Seite 851.
6. Optional: Sie können die eigenständige Dateiprotokollfunktion weiter konfigurieren, indem Sie die Datei logger.properties bearbeiten, die bei Ausführung des Befehls **fteCreateLogger** erstellt wurde. Bei dieser Datei handelt es sich um eine Java-Eigenschaftendatei, die Schlüssel/Wert-Paare enthält. Die Datei logger.properties befindet sich im Verzeichnis *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name*. Weitere Informationen zu den verfügbaren Eigenschaften und deren Auswirkungen finden Sie unter MFT-Konfigurationseigenschaften der Protokollfunktion.
7. Windows
   Optional: Bei Verwendung von Windows können Sie die eigenständige Dateiprotokollfunktion auch als Windows-Dienst ausführen. Führen Sie den Befehl **fteModifyLogger** mit dem Parameter **-s** aus. Weitere Informationen finden Sie in fteModifyLogger .
8. Starten Sie die eigenständige Dateiprotokollfunktion mit dem Befehl **fteStartLogger** . Weitere Informationen finden Sie in fteStartLogger .

   Wenn Sie den vorherigen Schritt ausgeführt und den **fteModifyLogger** -Befehl mit dem Parameter **-s** unter Windowsverwendet haben, wird die eigenständige Dateiprotokollfunktion als Windows -Dienst gestartet.
9. Überprüfen Sie die Protokollfunktionsausgabe. Die eigenständige Dateiprotokollfunktion generiert zwei Typen von Ausgabe-, Dateiübertragungs-und Logger-Diagnosedaten. Die Prüfdaten für die Dateiübertragung finden Sie in *MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/loggers/*logger_name*/logs. Die Diagnosedaten der Protokollfunktion befinden sich in *MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/loggers/*logger_name*
10. Sie können die Protokollfunktion mit dem Befehl **fteStopLogger** stoppen. Weitere Informationen finden Sie in fteStopLogger .

## Ergebnisse

### Zugehörige Tasks

„Benutzerzugriff für eine eigenständige MFT-Dateiprotokollfunktion konfigurieren” auf Seite 851
In einer Testumgebung können Sie neue erforderliche Berechtigungen zu einem normalen Benutzerkonto hinzufügen. In einer Produktionsumgebung ist zu empfehlen, einen neuen Benutzer mit den Berechtigungen zu erstellen, die für die Durchführung des Jobs mindestens erforderlich sind.

### Zugehörige Verweise

Konfigurationseigenschaften der MFT-Protokollfunktion
fteStartLogger
fteCreateLogger

fteModifyLogger

fteStopLogger

„Format der eigenständigen MFT-Dateiprotokollfunktion” auf Seite 846
Das Format der Nachrichteninformationen, die von der Dateiprotokollfunktion geschrieben werden, kann in der `FileLoggerFormat.xml`-Datei definiert werden.

Berechtigungen für die MFT-Protokollfunktion

## ALW *Format der eigenständigen MFT-Dateiprotokollfunktion*

Das Format der Nachrichteninformationen, die von der Dateiprotokollfunktion geschrieben werden, kann in der `FileLoggerFormat.xml`-Datei definiert werden.

Das Konfigurationsverzeichnis für die Protokollfunktion befindet sich in *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/loggers/`*logger_name*. Wenn Sie eine neue Dateiprotokollfunktion erstellen, wird eine Version dieser Datei erstellt, die eine Standardgruppe von Definitionen enthält, die von der Dateiprotokollfunktion verwendet wird. Weitere Informationen zu der Standardprotokollformatdefinition finden Sie unter MFT-Standardprotokollformat für die Standalone-Dateiprotokollfunktion.

Wenn Sie Ihr eigenes benutzerdefiniertes Protokollformat angeben möchten, bearbeiten Sie die Datei `FileLoggerFormat.xml`.

### Eine benutzerdefinierte Protokollformatdefinition

Eine Protokollformatdefinition besteht aus einer Gruppe von Nachrichtentypen mit jedem Nachrichtentyp mit einer Formatdefinition. Eine Formatdefinition für einen Nachrichtentyp besteht aus einer Gruppe von Einfügungen, die im XPATH-Format bereitgestellt werden, und einem Trennzeichen, das zum Trennen der einzelnen Einfügungen verwendet wird. Die Reihenfolge der Einfügungen bestimmt die Reihenfolge, in der der Inhalt in die Zeilen gestellt wird, die für die Ausgabe in die Protokolldateien generiert werden. Dies ist z. B. die Definition für den Nachrichtentyp `callStarted` :

```
<callStarted>
  <format>
    <inserts>
      <insert type="user"   width="19" ignoreNull="false">/transaction/action/
       @time</insert>
      <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6"  ignoreNull="false">type</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/
       @agent</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@QMgr</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/job/name</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/
       call/command/@type</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/
      call/command/@name</insert>
      <insert type="system" width="0"  ignoreNull="true">callArguments</insert>
    </inserts>
    <separator></separator>
  </format>
</callStarted>
```

In diesem Format wird eine Zeile in der Protokolldatei wie folgt erzeugt:

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
  AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

Die in der Formatdefinition enthaltenen Einfügungen befinden sich in der Reihenfolge, in der die Informationen in der Zeile in der Protokolldatei angezeigt werden. Weitere Informationen zum XML-Schema, das das Format für die `FileLoggerFormat.xml`-Datei definiert, finden Sie unter XSD-Format für eigenständige Dateiprotokollfunktion.

## Nachrichtentypen

Die FTE-Agenten schreiben eine Reihe unterschiedlicher Nachrichtentypen in das Unterthema SYSTEM.FTE/Log . Weitere Informationen finden Sie unter SYSTEM.FTE Thema. Die Protokolldateidefinition kann Formatdefinitionen für diese Typen von Nachrichten enthalten:

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

Das Format der Nachrichten kann variieren. Die meisten Nachrichtentypen schreiben in der Protokolldatei eine einzelne Zeile für jede Protokollnachricht, die aus dem Unterabschnitt SYSTEM.FTE/Log verarbeitet wird. Dies führt zu dem einfachen Fall, in dem sich die in der Protokollformatdefinition angegebenen XPATH-Adressen auf das Stammelement der Nachricht beziehen. Hierbei handelt es sich um die Nachrichtentypen, die diese Methode zum Schreiben der Ausgabe verwenden:

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

Die andere Methode, die zum Schreiben einer Protokollnachricht verwendet wird, verwendet mehrere Zeilen, um die Elemente in einem Übertragungssatz innerhalb einer Protokollnachricht darzustellen. In diesem Fall wird das bereitgestellte Format auf jedes Element in der Übertragungsgruppe in der Protokollnachricht angewendet. Wenn Sie Informationen enthalten möchten, die für jedes Element in der Übertragungsgruppe spezifisch sind, ist der bereitgestellte XPATH erforderlich, um das Element als XPATH-Stammverzeichnis verwenden zu können. Hierbei handelt es sich um die Nachrichtentypen, die diese Methode zum Schreiben der Ausgabe verwenden:

```
scheduleSubmitTransferSet
transferProgress
```

Für jedes Element in der Übertragungsgruppe wird eine Zeile der Ausgabe geschrieben. Informationen, die für alle Elemente in einem Übertragungsset festgelegt werden sollen, können weiterhin XPATH-Adressen relativ zum Stammverzeichnis der Protokollnachricht verwenden. Im folgenden vereinfachten `transferProgress` -Formatdefinitionsbeispiel ist es die Zeitmarke und die Übertragungs-ID, die festgelegt wurden. Alle Informationen, die relativ zu einem Element als Root sind, werden für jede Zeile,

die geschrieben wird, unterschiedlich sein. In diesem Beispiel werden die Quellen-und Zieldateiinformationen für die einzelnen Elemente geschrieben.

```
<transferProgress>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/
           @time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="true">status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file |
           source/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file/@size |
           source/queue/@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">source/@type</insert>
          <insert type="user"   width="6"  ignoreNull="true">source/@disposition</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file |
            destination/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file/@size |
            destination/queue/@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">destination/@type</insert>
          <insert type="user"   width="9"  ignoreNull="true">destination/@exist</insert>
          <insert type="user"   width="0"  ignoreNull="true">status/supplement</insert>
        </inserts>
        <separator></separator>
      </format>
    </transferProgress>
```

Dadurch wird ein Protokolldateieintrag mit einer oder mehreren Zeilen in diesem Format erstellt:

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
  ;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
  ;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

## Format einfügen

Beim Definieren eines Formats für einen Nachrichtentyp stehen zwei Typen von Einfügetypen zur Verfügung: `Benutzer` und `System` . Der Typ einer Einfügung wird im Attribut `type` des Einfügeelements definiert. Beide Typen von Einfügungen können auch über die Attribute **`width`** und **`ignoreNull`** des Einfügeelements angepasst werden. For example:

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

In diesem Beispiel nimmt die Einfügung die Informationen in der Protokollnachricht bei `/transaction/@ID` und trims an oder füllt sie auf 48 Zeichen auf, bevor sie in das Protokoll geschrieben wird. Wenn der Inhalt von `/transaction/@ID` null ist, schreibt er die Zeichenfolge null, nachdem er ihn auf 48 Zeichen aufgefüllt hat, da das Attribut `ignoreNull` auf `false` gesetzt ist. Wenn `ignoreNull` auf `true` gesetzt ist, wird stattdessen die leere Zeichenfolge, die auf 48 Zeichen aufgefüllt ist, geschrieben. Wenn Sie width="0" festlegen, bedeutet dies, dass die Spaltenbreite nicht getrimmt wird. Es bedeutet nicht, dass die Breite auf 0 getrimmt wird. Das Attribut `ignoreNull` kann auf diese Weise verwendet werden, um im Protokoll zu erkennen, wann ein Nullwert gefunden wird, wenn es nicht erwartet wurde. Dies kann beim Debugging einer neuen Protokolldateidefinition nützlich sein.

## Benutzerdefinierte Einfügungen

Eine Benutzereinfügung enthält eine XPATH-Adresse für die Informationen, die in diese Einfügung geschrieben werden sollen. Diese Adresse bezieht sich auf ein Teil der Informationen, die in der FTE-Protokollnachricht gefunden wurden. Weitere Informationen zu Protokollnachrichtenformaten finden Sie unter:

- Nachrichtenformate für Dateiübertragungsprotokolls
- Nachrichtenformate für geplante Dateiübertragungsprotokolls
- Protokollnachrichtenformat für MFT-Überwachungsprotokoll

## Vom System definierte Einfügungen

Systemdefinierte Einfügungen enthalten ein Schlüsselwort, das auf ein Teil der Informationen verweist, die entweder nicht in der Protokollnachricht gefunden werden können oder nicht leicht in der XPATH-Sprache definiert werden können.

Folgende Systemeinfügungen werden unterstützt:

- `type` -Schreibt den Typ der Protokollnachricht in einem kurzen Format.
- `callArguments` -Schreibt die Gruppe von Argumenten, die einem verwalteten Aufruf bereitgestellt werden, in einem durch Leerzeichen getrennten Format.
- `transferMetaData` -Schreibt die Gruppe von Metadateneinträgen, die für eine Übertragung definiert sind, in einem durch Kommas getrennten *key* = *value* -Format.

In der folgenden Tabelle ist der Wert des Typs "type" für systemdefinierte Einfügungen für jeden Nachrichtentyp aufgeführt.

*Tabelle 50. Zusammenfassung der unterstützten Nachrichtentypen und ihrer Systemeinfügungen vom Typ "Typ".*

| **Nachrichtentyp** | **Wert des Systemeinfügetyps "Typ"** |
|---|---|
| callCompleted | [ CCOM] |
| callStarted | [ CSTR] |
| monitorAction | [ MACT] |
| monitorCreate | [ MCRT] |
| monitorFired | [ MFIR] |
| notAuthorized | [ AUTH] |
| scheduleDelete | [ SDEL] |
| scheduleExpire | [ SEXP] |
| scheduleSkipped | [ SSKP] |
| scheduleSubmitInfo | [ SSIN] |
| scheduleSubmitTransfer | [ SSTR] |
| scheduleSubmitTransferSet | [ SSTS] |
| transferStarted | [ TSTR] |
| transferCancelled | [ TCAN] |
| transferComplete | [ TCOM] |
| transferDelete | [ TDEL] |
| transferProgress | [ TPRO] |

**Zugehörige Verweise**

Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion

XSD-Format (eigenständiges Dateiprotokollfunktionsformat)
Thema 'SYSTEM.FTE'
Nachrichtenformate für Dateiübertragungsprotokolls
Nachrichtenformate für geplante Dateiübertragungsprotokolls
Protokollnachrichtenformat für MFT-Überwachungsprotokoll

ALW *Nachrichtentypen aus der eigenständigen MFT-Dateiprotokollfunktion ausschließen*
Soll ein bestimmter Nachrichtentyp nicht in der Ausgabe der Dateiprotokollfunktion enthalten sein, können Sie leere Nachrichtentypelemente verwenden.

**Beispiel**

Die folgende Formatdefinition unterbindet beispielsweise die Ausgabe von `transferProgress`-Nachrichten durch die Dateiprotokollfunktion.

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
 xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
  </messageTypes>
</logFormatDefinition>
```

ALW *Benutzerdefinierte Formate für die eigenständige MFT-Dateiprotokollfunktion definieren*
Es besteht die Möglichkeit, in einer Protokollformatdefinition eine begrenzte Anzahl an benutzerdefinierten Nachrichtentypen zu definieren, um den Konfigurationsaufwand bei der Anpassung des Protokolldateiformats zu begrenzen.

## Informationen zu diesem Vorgang

Ist ein `messageTypes`-Element nicht in der Datei `FileLoggerFormat.xml` enthalten, wird für diesen Nachrichtentyp das Standardformat verwendet. Sie müssen nur die Formate angeben, die vom Standardformat abweichen.

**Beispiel**

In diesem Beispiel wird das Standardformat für den Nachrichtentyp `transferStarted` durch die Formatdefinition mit dieser reduzierten Version ersetzt, bei der nur der Benutzer ausgegeben wird, der die Übertragung gestartet hat. Für alle anderen Nachrichtentypen wird das Standardformat verwendet, da sie nicht in dieser Protokollformatdefinition enthalten sind:

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
 xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/
           @time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/
           userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

**Zugehörige Verweise**
Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion

XSD-Format (eigenständiges Dateiprotokollfunktionsformat)

ALW *Anzahl doppelter Nachrichten in der eigenständigen MFT-Dateiprotokollfunktion reduzieren*
Doppelte Protokollnachrichten können im Protokoll der eigenständigen Dateiprotokollfunktion auftreten. Durch die Verwendung der `logger.properties`-Datei können Sie die eigenständige Dateiprotokollfunktion optimieren und die Anzahl der Duplikate reduzieren.

## Doppelte Nachrichten im Protokoll der Dateiprotokollfunktion

Im Falle eines Fehlers wird möglicherweise eine Protokollnachricht in das Protokoll der eigenständigen Dateiprotokollfunktion geschrieben, ohne dass die Protokollnachricht vom SYSTEM.FTE/Log# Topic wird in IBM MQfestgeschrieben. Wenn dies der Fall ist, ruft die eigenständige Dateiprotokollfunktion erneut dieselbe Nachricht ein und schreibt sie erneut in die Protokolldatei. Planen Sie die Möglichkeit, diese Duplikate zu bearbeiten, wenn Sie die Protokolldateien entweder manuell oder automatisch bearbeiten. Um die Erkennung von Duplikaten zu unterstützen, gibt die eigenständige Dateiprotokollfunktion die folgende Nachricht in die Protokolldatei aus, wenn sie gestartet wird:

```
BFGDB0054I: The file logger has successfully started
```

Duplikate werden immer um die Startzeit der eigenständigen Dateiprotokollfunktion ausgeführt, da dies der Zeitpunkt ist, an dem die letzte Nachricht gelesen wurde, bevor die vorherige Instanz fehlgeschlagen ist. Wenn Sie wissen, wann die neue Instanz gestartet wurde, können Sie feststellen, ob Duplikate zu erwarten sind, und ob sie bearbeitet werden müssen oder nicht.

## Reduzieren der Anzahl der Duplikate

Die eigenständige Dateiprotokollfunktion gruppiertgemeinsam Protokollnachrichten, die sie in Transaktionen verarbeitet, um die Leistung zu verbessern. Bei dieser Stapelgröße handelt es sich um die maximale Anzahl doppelter Nachrichten, die im Falle eines Fehlers angezeigt werden. Um die Anzahl der Duplikate zu reduzieren, können Sie die folgende Eigenschaft in der `logger.properties`-Datei optimieren:

```
wmqfte.max.transaction.messages
```

Wenn Sie beispielsweise diese Einstellung auf 1 setzen, wird die maximale Anzahl der duplizierten Nachrichten auf 1 reduziert. Beachten Sie, dass die Änderung dieses Werts Auswirkungen auf die Leistung Ihrer eigenständigen Dateiprotokollfunktion hat, so dass gründliche Tests erforderlich sind, um sicherzustellen, dass dies Ihr System nicht beeinträchtigt.

Die Datei `logger.properties` befindet sich im Verzeichnis *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/loggers/`*logger_name*. Weitere Informationen zu den verfügbaren Eigenschaften und deren Auswirkungen finden Sie unter MFT-Konfigurationseigenschaften der Protokollfunktion.

ALW ***Benutzerzugriff für eine eigenständige MFT-Dateiprotokollfunktion konfigurieren***
In einer Testumgebung können Sie neue erforderliche Berechtigungen zu einem normalen Benutzerkonto hinzufügen. In einer Produktionsumgebung ist zu empfehlen, einen neuen Benutzer mit den Berechtigungen zu erstellen, die für die Durchführung des Jobs mindestens erforderlich sind.

## Informationen zu diesem Vorgang

Die eigenständige Dateiprotokollfunktion und IBM MQ müssen auf demselben System installiert werden. Konfigurieren Sie die Berechtigungen des Benutzers wie folgt:

## Vorgehensweise

1. Stellen Sie sicher, dass der Benutzer über Lese- und (bei Bedarf) Ausführungsberechtigung für die Dateien verfügt, die als Teil von Managed File Transfer installiert werden.

2. Stellen Sie sicher, dass der Benutzer über die Berechtigung zum Erstellen und Schreiben in einer Datei im Verzeichnis `logs` verfügt, die sich im Konfigurationsverzeichnis befindet. Dieses Verzeichnis wird für alle Ereignisprotokolle verwendet sowie bei Bedarf für Diagnosetracedateien und FFDC-Dateien (First-Failure Data Capture; Datenerfassung bei erstmaligem Fehlervorkommen).
3. Stellen Sie sicher, dass der Benutzer in einer eigenen Gruppe enthalten ist und nicht zu Gruppen mit weit reichenden Berechtigungen im Koordinations-Warteschlangenmanager gehört. Der Benutzer sollte nicht der Gruppe 'mqm' angehören. Auf einigen Plattformen erhält die Mitarbeitergruppe automatisch ebenfalls Zugriff auf den Warteschlangenmanager; der Benutzer der eigenständigen Dateiprotokollfunktion sollte nicht zu dieser Gruppe gehören. In IBM MQ Explorer können Sie die Berechtigungssätze für den Warteschlangenmanager selbst sowie für die Objekte auf dem Warteschlangenmanager anzeigen. Klicken Sie mit der rechten Maustaste auf das Objekt, und wählen Sie **Objektberechtigungen > Berechtigungsdatensätze verwalten** aus. In der Befehlszeile können Sie die Befehle dspmqaut (Anzeigeberechtigung) oder dmpmqaut (Speicherauszugsberechtigung) verwenden.
4. Fügen Sie im Fenster **Manage Authority Records** (Berechtigungssätze verwalten) von IBM MQ Explorer oder mit dem Befehl setmqaut (Berechtigung erteilen oder entziehen) Berechtigungen für die eigene Gruppe des Benutzers hinzu (unter AIX sind IBM MQ-Berechtigungen nur Gruppen zugeordnet, keinen einzelnen Benutzern). Folgende Berechtigungen sind erforderlich:
   - Verbindungsberechtigung (Connect) und Abfrageberechtigung (Inquire) auf dem Warteschlangenmanager (für die IBM MQ Java-Bibliotheken ist die Abfrageberechtigung erforderlich).
   - Subskriptionsberechtigung (Subscribe) für das Thema SYSTEM.FTE
   - PUT-Berechtigung für die Warteschlange SYSTEM.FTE.LOG.RJCT.*Name_der_Protokollfunktion*.
   - GET-Berechtigung für die Warteschlange SYSTEM.FTE.LOG.CMD.*Name_der_Protokollfunktion*.

   Die oben angegebenen Zurückweisungs- und Befehlswarteschlangennamen sind die Standardnamen. Wenn Sie bei der Konfiguration der Warteschlangen für die eigenständige Dateiprotokollfunktion andere Namen für die Warteschlangen angegeben haben, müssen Sie die Berechtigungen diesen Namen zuordnen.

## Eigenständige MFT-Datenbankprotokollfunktion installieren

Führen Sie die folgenden Schritte aus, um die eigenständige Datenbankprotokollfunktion zu installieren und zu konfigurieren.

### Informationen zu diesem Vorgang

**Wichtig:** Managed File Transfer-Protokollfunktionen werden auf der IBM i-Plattform nicht unterstützt.

Weitere Informationen zur eigenständigen Datenbankprotokollfunktion finden Sie im Abschnitt „MFT-Protokollfunktion konfigurieren” auf Seite 843.

**Anmerkung:** Für das gleiche Schema einer Datenbank kann jeweils nur eine Datenbankprotokollfunktion (eigenständige oder Java EE) ausgeführt werden. Der Versuch, dies zu tun, würde zu Überschneidungen führen, wenn versucht wird, Daten der Übertragungsprotokolls in die Datenbank zu schreiben.

### Vorgehensweise

1. Installieren Sie Ihre Datenbanksoftware mit Hilfe der Dokumentation für Ihre Datenbank.

   Wenn die JDBC-Unterstützung eine optionale Komponente für Ihre Datenbank ist, müssen Sie diese Komponente installieren.
2. Führen Sie den Befehl **`fteCreateLogger`** aus und setzen Sie den Parameter **`-loggerType`** auf DATABASE, um die eigenständige Datenbankprotokollfunktion zu erstellen. Weitere Informationen finden Sie in fteCreateLogger .

   Der Standardschemaname lautet FTELOG. Wenn Sie einen anderen Schemanamen als FTELOG verwenden, müssen Sie die bereitgestellte SQL-Datei für Ihre Datenbank, `ftelog_tables_db2.sql` oder `ftelog_tables_oracle.sql`, bearbeiten, um diesen Schemanamen widerzuspiegeln, bevor Sie mit dem nächsten Schritt fortfahren. Weitere Informationen finden Sie unter 'wmqfte.database.schema' in MFT-Konfigurationseigenschaften der Protokollfunktion.

3. Erstellen Sie die erforderlichen Datenbanktabellen mit den Tools Ihrer Datenbank.

   Multi Unter Multiplatforms enthalten die Dateien `ftelog_tables_db2.sql` und `ftelog_tables_oracle.sql` SQL-Befehle, die zum Erstellen der Tabellen ausgeführt werden können.

   z/OS Unter z/OS ist es von der von Ihnen verwendeten Version von Db2 for z/OS abhängig, welche Datei Sie ausführen müssen:

   - Führen Sie für Db2 for z/OS 9.0 und frühere Versionen die Datei `ftelog_tables_zos.sql` aus, um die Tabellen zu erstellen. Diese Datei erstellt die Tabellen mit einem Datentyp INTEGER für Felder, die die Größe der übertragenen Dateien und die Tabellen-ID, die jeder Übertragung zugeordnet sind, angeben.
   - Führen Sie für Db2 for z/OS 9.1 und höher die Datei `ftelog_tables_zos_bigint.sql` aus, um die Tabellen zu erstellen. Diese Datei erstellt die Tabellen mit einem BIGINT-Datentyp für Felder, die die Größe der übertragenen Dateien und die Tabellen-ID, die jeder Übertragung zugeordnet sind, angeben.
4. Führen Sie die mit dem Befehl **fteCreateLogger** bereitgestellten MQSC-Befehle für Ihren Befehlswarteschlangenmanager der Protokollfunktion aus, um die Warteschlangen der Protokollfunktion zu erstellen. Die eigenständige Datenbankprotokollfunktion verwendet zwei Warteschlangen auf dem Koordinationswarteschlangenmanager. Die erste Warteschlange ist eine Befehlswarteschlange, in die Nachrichten zur Steuerung der Operation der eigenständigen Datenbankprotokollfunktion gestellt werden. Der Standardname dieser Befehlswarteschlange lautet SYSTEM.FTE.LOG.CMD. *logger_name*. Die zweite Warteschlange ist eine Zurückweisungswarteschlange. Da die eigenständige Datenbankprotokollfunktion keine Protokollnachrichten verwirft, wenn die Protokollfunktion eine Nachricht feststellt, die sie nicht verarbeiten kann, wird die Nachricht in die Zurückweisungswarteschlange für die Untersuchung und die mögliche erneute Verarbeitung versetzt. Es wird nicht empfohlen, die Warteschlange für nicht zustellbare Nachrichten des WS-Managers zu diesem Zweck zu verwenden, da zurückgewiesene Nachrichten keinen DLH-Header haben und weil zurückgewiesene Nachrichten nicht mit Nachrichten kombiniert werden sollten, die aus anderen Gründen in die Warteschlange für nicht zustellbare Nachrichten gestellt werden. Der Standardname für die Zurückweisungswarteschlange ist SYSTEM.FTE.LOG.RJCT. *logger_name*. Diese beiden Warteschlangen werden in den MQSC-Scriptdateien definiert, die vom **fteCreateLogger** -Befehl generiert wurden.
5. Benutzer auswählen und Berechtigungen konfigurieren
6. Optional: Über die Datei `logger.properties`, die bei der Ausführung des Befehls **fteCreateLogger** in Schritt „2" auf Seite 852 erstellt wurde, können Sie weitere Konfigurationsschritte für die eigenständige Datenbankprotokollfunktion vornehmen. Bei dieser Datei handelt es sich um eine Java-Eigenschaftendatei, die Schlüssel/Wert-Paare enthält. Die Datei `logger.properties` befindet sich im Verzeichnis *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/loggers/`*logger_name*. Weitere Informationen zu den verfügbaren Eigenschaften und deren Auswirkungen finden Sie unter MFT-Konfigurationseigenschaften der Protokollfunktion.
7. Windows
   Optional: Bei Verwendung von Windows können Sie die eigenständige Datenbankprotokollfunktion auch als Windows-Dienst ausführen. Führen Sie den Befehl **fteModifyLogger** mit dem Parameter **-s** aus. Weitere Informationen finden Sie in fteModifyLogger .
8. Optional: Wenn Sie eine Oracle-Datenbank verwenden oder remote auf eine Db2-Datenbank zugreifen, müssen Sie Benutzername und Kennwort für die Authentifizierung der Protokollfunktion beim Datenbankserver eingeben. Dieser Benutzername und das zugehörige Kennwort werden in einer Berechtigungsnachweisdatei angegeben, die dem Format entspricht, das durch das `MQMFTCredentials.xsd`-Schema definiert ist. Weitere Informationen finden Sie im Abschnitt MFT-Dateiformat für Berechtigungsnachweise . Nachdem Sie die Berechtigungsnachweisdatei erstellt haben, müssen Sie die Position der Berechtigungsnachweisdatei in der `logger.properties` -Datei mit der Eigenschaft `wmqfte.database.credentials.file` angeben.
9. Starten Sie die eigenständige Datenbankprotokollfunktion mit dem Befehl **fteStartLogger** . Standardmäßig wird die eigenständige Datenbankprotokollfunktion im Hintergrund ausgeführt, und die eigenständige Datenbankprotokollfunktion stellt die Ausgabe in eine Datei im Verzeichnis `logs` ein. Wenn Sie die eigenständige Datenbankprotokollfunktion im Vordergrund ausführen und eine Ausgabe

in der Konsole sowie in der Protokolldatei erzeugen wollen, fügen Sie den Parameter **-F** zum Befehl **fteStartLogger** hinzu.

Wenn Sie den vorherigen Schritt ausgeführt haben und den Befehl **fteModifyLogger** mit dem Parameter **-s** unter Windowsverwendet haben, wird die eigenständige Datenbankprotokollfunktion als Windows -Dienst gestartet.

**Zugehörige Tasks**

„Benutzerzugriff für eine eigenständige MFT-Datenbankprotokollfunktion konfigurieren" auf Seite 855
In einer Testumgebung können Sie neue erforderliche Berechtigungen zu einem normalen Benutzerkonto hinzufügen. In einer Produktionsumgebung ist zu empfehlen, einen neuen Benutzer mit den Berechtigungen zu erstellen, die für die Durchführung des Jobs mindestens erforderlich sind.

**Zugehörige Verweise**

Konfigurationseigenschaften der MFT-Protokollfunktion
fteStartLogger
fteModifyLogger
Berechtigungen für die MFT-Protokollfunktion

### *MFT mit einer fernen Datenbank verwenden*

Mithilfe der Managed File Transfer-Protokollfunktion können Sie mit einer Datenbank auf einem fernen System kommunizieren.

## Informationen zu diesem Vorgang

Ist die Datenbank auf einem anderen System als Managed File Transfer installiert, müssen Sie wie im Folgenden beschrieben vorgehen. Sofern nicht anders angegeben, gelten die Schritte sowohl für Db2 als auch für Oracle.

## Vorgehensweise

1. Installieren Sie einen Datenbankclient auf dem System, auf dem sich auch Managed File Transfer befindet.
2. Fügen Sie der Konfiguration Ihres lokalen Datenbankclients den fernen Datenbankserver hinzu. Diese Konfigurationsaktualisierung ist für den korrekten Zugriff von Managed File Transfer und IBM MQ auf die Datenbank erforderlich.
3. Geben Sie die neuen Eigenschaften in der Datei `logger.properties` an, um über die Berechtigungsnachweisdatei **wmfte.database.credentials.file** eine Verbindung zur Datenbank herzustellen.

   **Anmerkung:** Deprecated In früheren Versionen von Managed File Transfer wurden die Eigenschaften **wmqfte.oracle.user** oder **wmqfte.database.user** und **wmqfte.oracle.password** oder **wmqfte.database.password** verwendet. Diese Eigenschaften werden jedoch nicht weiter unterstützt. Verwenden Sie stattdessen **wmfte.database.credentials.file**.
4. **Nur Oracle:** Wenn Sie eine remote Verbindung zur Datenbank zulassen möchten, ändern Sie die XAResourceManager-Zeilengruppe in der `qm.ini`-Datei des Koordinations-WS-Managers wie folgt (stellen Sie sicher, dass Sie den Datenbanknamen, den Benutzernamen und das Benutzerkennwort so ändern, dass sie Ihren eigenen Informationen entsprechen):

   `Oracle_XA+Acc=P/ftelog/qgw783jhT+SesTm=35+DB=FTEAUDIT1`**`+SqlNet=FTEAUDIT1`**`+threads=false` ist die Änderung fett hervorgehoben.
5. **Nur Oracle:** Geben Sie einen Host und einen Port in der Datei `logger.properties` mit den Eigenschaften **wmqfte.oracle.host** und **wmqfte.oracle.port** an. Die Arbeit mit einem lokalen Datenbankclient ist bereits mit den Standardwerten für Host und Port möglich. Wenn Sie also zuvor mit einer lokalen Datenbank gearbeitet haben, wurden für diese Eigenschaften möglicherweise noch keine Werte festgelegt.

**Zugehörige Verweise**

Konfigurationseigenschaften der MFT-Protokollfunktion

### *Benutzerzugriff für eine eigenständige MFT-Datenbankprotokollfunktion konfigurieren*

In einer Testumgebung können Sie neue erforderliche Berechtigungen zu einem normalen Benutzerkonto hinzufügen. In einer Produktionsumgebung ist zu empfehlen, einen neuen Benutzer mit den Berechtigungen zu erstellen, die für die Durchführung des Jobs mindestens erforderlich sind.

## Informationen zu diesem Vorgang

Anzahl und Typ der zur Ausführung der eigenständigen Datenbankprotokollfunktion benötigten Benutzerkonten hängt von der Anzahl der Systeme ab, die eingesetzt werden. Sie können die eigenständige Datenbankprotokollfunktion, IBM MQ und die Datenbank auf einem oder auf zwei Systemen verteilt installieren. Die eigenständige Datenbankprotokollfunktion muss sich auf demselben System befinden wie IBM MQ. Die Komponenten können in den folgenden Topologien installiert werden:

**Eigenständige Datenbankprotokollfunktion, IBM MQ und die Datenbank auf demselben System**
Sie können einen einzelnen Betriebssystembenutzer für die Nutzung aller drei Komponenten definieren. Dies ist eine passende Konfiguration für die eigenständige Datenbankprotokollfunktion. Die eigenständige Protokollfunktion verwendet den Bindungsmodus für die Verbindung mit IBM MQ und eine native Verbindung für den Zugriff auf die Datenbank.

**Eigenständige Datenbankprotokollfunktion und IBM MQ auf einem System, die Datenbank auf einem separaten System**
Für diese Konfiguration werden zwei Benutzer erstellt, einmal ein Betriebssystembenutzer auf dem System, auf dem die eigenständige Datenbankprotokollfunktion aktiv ist, einmal ein Betriebssystembenutzer mit Remotezugriff auf die Datenbank auf dem Datenbankserver. Dies ist eine passende Konfiguration für die eigenständige Datenbankprotokollfunktion mit einer fernen Datenbank. Die eigenständige Protokollfunktion verwendet den Bindungsmodus für die Verbindung mit IBM MQ und eine Clientverbindung für den Zugriff auf die Datenbank.

Als Beispiel wird bei den restlichen Anweisungen davon ausgegangen, dass es sich bei dem Benutzer um `ftelog` handelt, Sie können jedoch einen anderen Benutzernamen verwenden. Konfigurieren Sie die Berechtigungen des Benutzers wie folgt:

## Vorgehensweise

1. Stellen Sie sicher, dass der Benutzer über die Berechtigung verfügt, die Dateien, die als Teil der Managed File Transfer Remote Tools and Documentation-Installation installiert werden, zu lesen und bei Bedarf auch auszuführen.
2. Stellen Sie sicher, dass der Benutzer über die Berechtigung zum Erstellen und Schreiben in einer Datei im Verzeichnis `logs` (im Konfigurationsverzeichnis) verfügt. Dieses Verzeichnis wird für ein Ereignisprotokoll und gegebenenfalls für Diagnosetrace- und FFDC-Dateien verwendet.
3. Stellen Sie sicher, dass der Benutzer über eine eigene Gruppe verfügt und kein Mitglied von Gruppen mit umfassenden Berechtigungen auf dem Koordinationswarteschlangenmanager ist. Der Benutzer sollte nicht der Gruppe 'mqm' angehören. Auf einigen Plattformen erhält die Mitarbeitergruppe automatisch ebenfalls Warteschlangenmanagerzugriff; die eigenständige Datenbankprotokollfunktion sollte nicht zur Mitarbeitergruppe gehören. In IBM MQ Explorer können Sie die Berechtigungssätze für den Warteschlangenmanager selbst sowie für die Objekte auf dem Warteschlangenmanager anzeigen. Klicken Sie mit der rechten Maustaste auf das Objekt, und wählen Sie **Objektberechtigungen > Berechtigungsdatensätze verwalten** aus. In der Befehlszeile können Sie die Befehle dspmqaut (Anzeigeberechtigung) oder dmpmqaut (Speicherauszugsberechtigung) verwenden.
4. Fügen Sie im Fenster **Manage Authority Records** (Berechtigungssätze verwalten) von IBM MQ Explorer oder mit dem Befehl setmqaut (Berechtigung erteilen oder entziehen) Berechtigungen für die eigene Gruppe des Benutzers hinzu (unter AIX sind IBM MQ-Berechtigungen nur Gruppen zugeordnet, keinen einzelnen Benutzern). Folgende Berechtigungen sind erforderlich:
   - Verbindungsberechtigung (Connect) und Abfrageberechtigung (Inquire) auf dem Warteschlangenmanager (für die IBM MQ Java-Bibliotheken ist die Abfrageberechtigung erforderlich).
   - Subskriptionsberechtigung (Subscribe) für das Thema SYSTEM.FTE
   - PUT-Berechtigung für die Warteschlange SYSTEM.FTE.LOG.RJCT.*Name_der_Protokollfunktion*.

- GET-Berechtigung für die Warteschlange SYSTEM.FTE.LOG.CMD.*Name_der_Protokollfunktion*.

Die oben angegebenen Zurückweisungs- und Befehlswarteschlangennamen sind die Standardnamen. Wenn Sie bei der Konfiguration der Warteschlangen für die eigenständige Datenbankprotokollfunktion andere Namen für die Warteschlangen angegeben haben, müssen Sie die Berechtigungen diesen Namen zuordnen.

5. Führen Sie die Benutzerkonfiguration aus, die für die von Ihnen verwendete Datenbank bestimmt ist.
   - Bei Verwendung einer Db2-Datenbank müssen Sie die folgenden Schritte ausführen:

     Es gibt verschiedene Mechanismen für die Verwaltung von Datenbankbenutzern mit Db2. Diese Anweisungen gelten für das Standardschema, das auf Betriebssystembenutzern basiert.

     – Stellen Sie sicher, dass der Benutzer `ftelog` zu keiner Db2-Verwaltungsgruppe (beispielsweise 'db2iadm1', 'db2fadm1' oder 'dasadm1') gehört.
     – Erteilen Sie dem Benutzer die Berechtigung, eine Verbindung zur Datenbank herzustellen, und die Berechtigung, in den unter Schritt 2: Erforderliche Datenbanktabellen erstellen erstellten Tabellen auszuwählen sowie Einfügungen und Aktualisierungen vorzunehmen.
   - Bei einer Oracle-Datenbank müssen Sie die folgenden Schritte ausführen:
     – Stellen Sie sicher, dass der Benutzer `ftelog` nicht zu einer Oracle-Verwaltungsgruppe (beispielsweise 'ora_dba' (unter Windows) oder 'dba' (unter AIX and Linux) gehört.
     – Erteilen Sie dem Benutzer die Berechtigung, eine Verbindung zur Datenbank herzustellen, und die Berechtigung, in den unter Schritt 2: Erforderliche Datenbanktabellen erstellen erstellten Tabellen auszuwählen sowie Einfügungen und Aktualisierungen vorzunehmen.

## Alternative Konfigurationen für eine eigenständige MFT-Protokollfunktion

In der Regel befindet sich die eigenständige Managed File Transfer-Protokollfunktion unabhängig vom Typ (Datei oder Datenbank) auf demselben System wie der Koordinationswarteschlangenmanager und ist im IBM MQ-Bindungsmodus mit diesem verbunden. Sie kann jedoch auch auf demselben System wie jeder WS-Manager installiert werden, der über eine Verbindung zum Koordinations-WS-Manager verfügt. Der eigenständige Logger empfängt Nachrichten mit einer Subskription, die die eigenständige Protokollfunktion automatisch erstellt. Dies ist die Konfiguration, die in den Installationsanweisungen beschrieben ist.

Wenn Sie jedoch über site-spezifische Überlegungen verfügen, können Sie eine eigenständige Protokollfunktion so konfigurieren, dass Nachrichten auf zwei andere Arten empfangen werden, die durch die Eigenschaft wmqfte.message.source.type gesteuert werden. Diese Eigenschaft wird in den Konfigurationseigenschaften der MFT-Protokollfunktion beschrieben.

### Verwaltungssubskription

Standardmäßig erstellt eine eigenständige Protokollfunktion ihre eigene Subskription für das Thema SYSTEM.FTE/Log/#, wobei die Standardoptionen für permanente Subskriptionen und eine verwaltete Subskription verwendet werden (d. a. der Warteschlangenmanager steuert die Sicherungswarteschlange, die zum Speichern der Nachrichten verwendet wird, bevor sie an die Anwendung übergeben werden). Wenn andere Optionen in der Subskription oder in der Warteschlange erforderlich sind, können Sie stattdessen selbst eine Subskription erstellen, die erforderlichen Optionen festlegen und die eigenständige Protokollfunktion so konfigurieren, dass sie stattdessen diese Subskription verwendet. Denken Sie daran, die Berechtigung für die eigenständige Protokollfunktion hinzuzufügen, um die von Ihnen erstellten Subskription zu verwenden.

Ein Beispiel für die Verwendung dieser Konfiguration ist das Partitionieren des Protokollspeicherbereichs mithilfe von zwei Platzhaltersubskriptionen zum Senden von Protokollen von Agenten, deren Name mit FINANCE beginnt, in eine Datenbank und Protokolle von Agenten, die mit ACCOUNTING beginnen, in eine andere Datenbank. Für diesen Konfigurationstyp sind zwei eigenständige Logger-Instanzen erforderlich, die jeweils eine eigene `logger.properties`-Datei enthalten, die sich auf das erforderliche Abonnement und die eigene Befehlswarteschlange und die eigene Zurückweisungswarteschlange bezieht.

Um Protokollnachrichten nur von Agenten zu erfassen, deren Namen mit ACCOUNTING beginnen, erstellen Sie ein Subskriptionsobjekt auf Ihrem Koordinationswarteschlangenmanager mit einer Themenzei-

chenfolge von SYSTEM.FTE/Log/ACCOUNTING*. Setzen Sie den Wert für **Platzhalterzeichen** auf **Platzhalterzeichen auf Zeichenebene** . Sie müssen auch Einträge zur `logger.properties`-Datei für Ihre Protokollfunktion hinzufügen. Wenn Sie beispielsweise ein Abonnementobjekt mit dem Namen ACCOUNTING.LOGS mit diesen Einstellungen erstellen, fügen Sie die folgenden Einträge zur Datei `logger.properties` hinzu:

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

Die eigenständige Protokollfunktion verarbeitet Protokollnachrichten, die mit der Themenzeichenfolge SYSTEM.FTE/Log/ beginnen. Sie können eine restriktivere Themenzeichenfolge angeben, aber Sie können keine weniger restriktive Zeichenfolge angeben. Wenn Sie eine weniger restriktive Zeichenfolge in Fehler angeben, werden alle Veröffentlichungen, die sich auf eine andere Themenzeichenfolge als SYSTEM.FTE/Log/ beziehen, in die Zurückweisungswarteschlange und die eigenständige Protokollfunktion die Fehlernachricht BFGDB0002E erstellt. Diese Fehlernachricht weist darauf hin, dass ein Problem mit der Konfiguration der eigenständigen Protokollfunktion aufgetreten ist.

### Warteschlange

In der typischen Topologie wird die eigenständige Protokollfunktion auf demselben System ausgeführt wie der Koordinations-WS-Manager. Wenn dies nicht möglich ist, können Sie eine Subskription auf dem Koordinations-WS-Manager erstellen, indem Sie eine Warteschlange in einem anderen Warteschlangenmanager als Subskriptionsziel verwenden (entweder mit Hilfe einer Definition einer fernen Warteschlange oder mithilfe der Eigenschaft DESTQMGR der Subskription). Die Protokollfunktion kann dann auf dem System ausgeführt werden, auf dem sich der zweite WS-Manager befindet, und die Nachrichten aus der Warteschlange lesen. Um die Transaktionsintegrität zu gewährleisten, muss die eigenständige Protokollfunktion immer eine Verbindung zu ihrem Warteschlangenmanager im Bindungsmodus herstellen. Sie müssen die Zurückweisungswarteschlange und die Befehlswarteschlange auf demselben Warteschlangenmanager definieren, zu dem die eigenständige Protokollfunktion eine Verbindung herstellt. Die Warteschlangenmanager müssen IBM WebSphere MQ 7.5 oder höher aufweisen.

Wenn Sie beispielsweise Protokollnachrichten erfassen möchten, die von einem Abonnement in die Warteschlange USER.QUEUE gestellt werden, fügen Sie diese Einträge zur Datei `logger.properties` hinzu:

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

## Java EE-Datenbankprotokollfunktion für MFT installieren

Dieser Abschnitt enthält die Anweisungen zum Installieren und Konfigurieren der JEE-Datenbankprotokollfunktion zur Verwendung mit Managed File Transfer.

### Informationen zu diesem Vorgang

Weitere Informationen zur Java EE-Datenbankprotokollfunktion finden Sie im Abschnitt „MFT-Protokollfunktion konfigurieren” auf Seite 843.

**Anmerkung:** Sie können eine Java EE-Datenbankprotokollfunktion nicht gleichzeitig mit einer eigenständigen Protokollfunktion ausführen, es sei denn, diese beiden Protokollfunktionen verwenden separate Instanzen der Datenbank.

### Vorgehensweise

1. Vor der Installation der Java EE-Datenbankprotokollfunktion müssen Sie Ihre Umgebung vorbereiten. Folgen Sie den Anweisungen im Abschnitt „Vorbereiten der Installation der Java EE-Datenbankprotokollfunktion für MFT” auf Seite 858.
2. Installieren Sie die Java EE -Datenbankprotokollfunktion in einem Java Platform, Enterprise Edition (Java EE) -oder Jakarta EE -konformen Anwendungsserver.

Anweisungen finden Sie im Abschnitt „Java EE-Datenbankprotokollfunktion für MFT mit WebSphere Application Server traditional 9.0 installieren” auf Seite 861

**Zugehörige Tasks**

„Vorbereiten der Installation der Java EE-Datenbankprotokollfunktion für MFT” auf Seite 858
Führen Sie die folgenden Anweisungen aus, um Ihre Managed File Transfer-Umgebung vorzubereiten, bevor Sie die Java EE-Datenbankprotokollfunktion installieren.

„Java EE-Datenbankprotokollfunktion für MFT mit WebSphere Application Server traditional 9.0 installieren” auf Seite 861
Führen Sie die folgenden Anweisungen aus, um die Java Platform, Enterprise Edition (Java EE)-Datenbankprotokollfunktion für Managed File Transfer mit WebSphere Application Server traditional 9.0 zu installieren und zu konfigurieren.

„Benutzerzugriff für die Java EE-Datenbankprotokollfunktion für MFT konfigurieren” auf Seite 866
Beim Konfigurieren der Datenbankprotokollfunktion für Java Platform, Enterprise Edition (Java EE) für Managed File Transfer benötigen Sie Benutzerkonten für den Zugriff auf IBM MQ, Ihr Datenbank- und Ihr Betriebssystem. Die Anzahl der erforderlichen Betriebssystembenutzer hängt von der Anzahl der Systeme ab, die Sie zum Hosten dieser Komponenten verwenden.

„Migration von der eigenständigen Datenbankprotokollfunktion in die Java EE-Datenbankprotokollfunktion für MFT” auf Seite 868
Sie können von der eigenständigen Datenbankprotokollfunktion auf die Java EE-Datenbankprotokollfunktion migrieren. Sie müssen die eigenständige Datenbankprotokollfunktion stoppen und die JEE-Datenbankprotokollfunktion installieren. Damit Protokolleinträge nicht verloren gehen oder dupliziert werden, müssen Sie die Veröffentlichung von Nachrichten im SYSTEM.FTE vor dem Stoppen der eigenständigen Datenbankprotokollfunktion und nach der Installation der Java EE -Datenbankprotokollfunktion einen Neustart durchführen. Sichern Sie Ihre Datenbank vor der Migration.

**Zugehörige Verweise**

Berechtigungen für die MFT-Protokollfunktion

## *Vorbereiten der Installation der Java EE-Datenbankprotokollfunktion für MFT*

Führen Sie die folgenden Anweisungen aus, um Ihre Managed File Transfer-Umgebung vorzubereiten, bevor Sie die Java EE-Datenbankprotokollfunktion installieren.

### Informationen zu diesem Vorgang

Weitere Informationen zur Java EE-Datenbankprotokollfunktion finden Sie im Abschnitt „MFT-Protokollfunktion konfigurieren” auf Seite 843.

### Vorgehensweise

1. Installieren Sie Ihre Datenbanksoftware mit Hilfe der Dokumentation für Ihre Datenbank.

   Wenn die JDBC-Unterstützung eine optionale Komponente für Ihre Datenbank ist, müssen Sie diese Komponente installieren.
2. Erstellen Sie eine Datenbank mit den Tools, die von Ihrer Datenbank bereitgestellt werden. Die Datenbank muss eine Tabellenbereichs-und Pufferpoolseitengröße von mindestens 8 KB aufweisen.

   Der Standardschemaname lautet FTELOG. Wenn Sie einen anderen Schemanamen als FTELOG verwenden, müssen Sie die bereitgestellte SQL-Datei für Ihre Datenbank, `ftelog_tables_db2.sql` oder `ftelog_tables_oracle.sql`, bearbeiten, um diese zu reflektieren, bevor Sie mit dem nächsten Schritt fortfahren.

   **Anmerkung:** Die Dateien `ftelog_tables_db2.sql` und `ftelog_tables_oracle.sql` befinden sich im Dateipfad `<MQ-installation-path>/mqft/sql`.
3. Erstellen Sie die erforderlichen Datenbanktabellen mit den Tools Ihrer Datenbank.

   Multi Unter Multiplatforms enthalten die Dateien `ftelog_tables_db2.sql` und `ftelog_tables_oracle.sql` SQL-Befehle, die zum Erstellen der Tabellen ausgeführt werden können.

z/OS Unter z/OS ist es von der von Ihnen verwendeten Version von Db2 for z/OS abhängig, welche Datei Sie ausführen müssen:

- Führen Sie für Db2 for z/OS 9.0 und frühere Versionen die Datei `ftelog_tables_zos.sql` aus, um die Tabellen zu erstellen. Diese Datei erstellt die Tabellen mit einem Datentyp INTEGER für Felder, die die Größe der übertragenen Dateien und die Tabellen-ID, die jeder Übertragung zugeordnet sind, angeben.
- Führen Sie für Db2 for z/OS 9.1 und höher die Datei `ftelog_tables_zos_bigint.sql` aus, um die Tabellen zu erstellen. Diese Datei erstellt die Tabellen mit einem BIGINT-Datentyp für Felder, die die Größe der übertragenen Dateien und die Tabellen-ID, die jeder Übertragung zugeordnet sind, angeben.

4. Wenn Sie den Schemanamen aus FTELOG geändert haben, müssen Sie den Schemanamen in der EAR-Datei ändern. Weitere Informationen finden Sie unter „Schemanamen in der Java EE-Datenbankprotokollfunktion für MFT ändern" auf Seite 859.
5. Erstellen Sie in IBM MQ eine Ablehnungswarteschlange.

   Da die Protokollfunktion keine Protokollnachrichten verwirft, wenn die Protokollfunktion eine Nachricht feststellt, die sie nicht verarbeiten kann, wird die Nachricht in die Zurückweisungswarteschlange für die Prüfung und die mögliche erneute Verarbeitung versetzt. Verwenden Sie die Warteschlange für nicht zustellbare Nachrichten des WS-Managers nicht für diesen Zweck, da zurückgewiesene Nachrichten keinen DLH-Header haben und weil zurückgewiesene Nachrichten nicht mit Nachrichten kombiniert werden dürfen, die aus anderen Gründen in die Warteschlange für nicht zustellbare Nachrichten gestellt werden. Der Befehl **`fteCreateLogger`** erstellt eine Zurückweisungswarteschlange. Der Standardname für diese Zurückweisungswarteschlange lautet SYSTEM.FTE.LOG.RJCT. *logger_name*
6. Führen Sie die Anweisungen im Abschnitt „Benutzerzugriff für die Java EE-Datenbankprotokollfunktion für MFT konfigurieren" auf Seite 866 aus.

## Nächste Schritte

Installieren Sie die Java EE -Datenbankprotokollfunktion in einem Java EE -oder Jakarta EE -konformen Anwendungsserver. Folgen Sie den Anweisungen im Abschnitt „Java EE-Datenbankprotokollfunktion für MFT mit WebSphere Application Server traditional 9.0 installieren" auf Seite 861

## *Schemanamen in der Java EE-Datenbankprotokollfunktion für MFT ändern*

Die Datenbankprotokollfunktion von Java Platform, Enterprise Edition (Java EE) kann eine Datenbank verwenden, die über einen nicht standardmäßigen Schemanamen verfügt. Sie müssen den Schemanamen in der EAR-Datei der Java EE-Datenbankprotokollfunktion ändern.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um den Namen des Schemas zu ändern, das von der Java EE-Datenbankprotokollfunktion verwendet wird:

## Vorgehensweise

1. Extrahieren Sie die JPA-JAR-Datei mit folgendem Befehl aus der EAR-Datei:

   ```
   jar -xvf ear_file lib/jpa_file
   ```

   Dabei gilt:

   - Die *EAR-Datei* ist `com.ibm.wmqfte.databaselogger.jee.oracle.ear` oder `com.ibm.wmqfte.databaselogger.jee.ear`, je nachdem, ob Sie Db2 or Oracle verwenden.
   - *jpa_file* ist `com.ibm.wmqfte.web.jpa.oracle.jar` oder `com.ibm.wmqfte.web.jpa.jar` je nachdem, ob Sie Db2 or Oracle verwenden.

2. Extrahieren Sie die `persistence.xml`-Datei aus der JPA-JAR-Datei, indem Sie den folgenden Befehl verwenden:

```
jar -xvf lib/jpa_file META_INF/persistence.xml
```

Dabei gilt:

- *jpa_file* ist `com.ibm.wmqfte.web.jpa.oracle.jar` oder `com.ibm.wmqfte.web.jpa.jar` je nachdem, ob Sie Db2 or Oracle verwenden.

3. Bearbeiten Sie die `persistence.xml`-Datei, um die folgende Zeile zu ändern:

```
<property name="openjpa.jdbc.Schema" value="schema_name" />
```

Dabei gilt Folgendes:

- *Schemaname* ist der Schemaname, den Sie verwenden möchten.

4. Aktualisieren Sie die JPA-JAR-Datei mit der geänderten Datei `persistence.xml`, indem Sie den folgenden Befehl verwenden:

```
jar -uvf lib/jpa_file META_INF/persistence.xml
```

Dabei gilt:

- *jpa_file* ist `com.ibm.wmqfte.web.jpa.oracle.jar` oder `com.ibm.wmqfte.web.jpa.jar` je nachdem, ob Sie Db2 or Oracle verwenden.

5. Aktualisieren Sie die EAR-Datei mit der geänderten JPA-JAR-Datei. Führen Sie dazu folgenden Befehl aus:

```
jar -uvf ear_file lib/jpa_file
```

Dabei gilt:

- Die *EAR-Datei* ist `com.ibm.wmqfte.databaselogger.jee.oracle.ear` oder `com.ibm.wmqfte.databaselogger.jee.ear`, je nachdem, ob Sie Db2 or Oracle verwenden.
- *jpa_file* ist `com.ibm.wmqfte.web.jpa.oracle.jar` oder `com.ibm.wmqfte.web.jpa.jar` je nachdem, ob Sie Db2 or Oracle verwenden.

## Nächste Schritte

Verwenden Sie die geänderte EAR-Datei, um die Java EE-Datenbankprotokollfunktion zu installieren.

**Zugehörige Tasks**

„Java EE-Datenbankprotokollfunktion für MFT mit WebSphere Application Server traditional 9.0 installieren” auf Seite 861

Führen Sie die folgenden Anweisungen aus, um die Java Platform, Enterprise Edition (Java EE)-Datenbankprotokollfunktion für Managed File Transfer mit WebSphere Application Server traditional 9.0 zu installieren und zu konfigurieren.

## *Pfad der nativen Bibliothek in WebSphere Application Server traditional 9.0 festlegen*

Wenn Sie die Java Platform, Enterprise Edition (Java EE)-Datenbank-Protokollanwendung auf WebSphere Application Server traditional 9.0 bereitstellen, und Bindings-Mode-Verbindungen zwischen der Anwendung und IBM MQ verwenden möchten, müssen Sie den IBM MQ-Messaging-Provider mit dem Ort der nativen Bibliotheken IBM MQ im System konfigurieren.

## Informationen zu diesem Vorgang

Wird der Pfad mit den nativen Bibliotheken nicht auf dem Anwendungsserver gesetzt, erhalten Sie unter Umständen im Ausgabeprotokoll auf dem WebSphere Application Server traditional 9.0-System die folgende Fehlernachricht:

```
A connection could not be made to WebSphere MQ for the following reason:
  CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnd' was not found. [3=mqjbnd]
```

Führen Sie in der Administrationskonsole von WebSphere Application Server traditional 9.0 die folgenden Schritte aus:

## Vorgehensweise

1. Erweitern Sie im Navigationsfenster die Einträge **Ressourcen** > **JMS** > **JMS-Provider**.
2. Wählen Sie den IBM MQ-Messaging-Provider im korrekten Bereich für die Verbindungsfactory oder die Aktivierungsspezifikation aus, mit der die Verbindung im Bindungsmodus erstellt wird.

   **Anmerkung:** Angaben zu nativen Pfaden im `Server`-Bereich wird der Vorzug vor Angaben zu nativen Pfaden höherer Bereiche gegeben, und Angaben zu nativen Pfadinformationen im `Node`-Bereich wird der Vorzug vor Angaben zu nativen Pfaden im `Cell`-Bereich gegeben.
3. Geben Sie im Feld **Native library path** (Pfad der nativen Bibliotheken) im Abschnitt mit den allgemeinen Eigenschaften den vollständigen Namen des Verzeichnisses an, das die nativen IBM MQ-Bibliotheken enthält.

   Geben Sie etwa auf Linux ein: `/opt/mqm/java/lib`. Geben Sie den Namen nur eines Verzeichnisses ein.
4. Klicken Sie auf **OK**.

   Sobald der Pfad festgelegt ist, sollten Sie die Änderungen an der Hauptkonfiguration speichern, damit diese wirksam werden.
5. Starten Sie den Anwendungsserver erneut, um die Konfiguration zu aktualisieren.
6. Erforderlich: Starten Sie den Anwendungsserver anschließend ein zweites Mal, um die Bibliotheken zu laden.

**Zugehörige Tasks**

„Java EE-Datenbankprotokollfunktion für MFT mit WebSphere Application Server traditional 9.0 installieren” auf Seite 861
Führen Sie die folgenden Anweisungen aus, um die Java Platform, Enterprise Edition (Java EE)-Datenbankprotokollfunktion für Managed File Transfer mit WebSphere Application Server traditional 9.0 zu installieren und zu konfigurieren.

# *Java EE-Datenbankprotokollfunktion für MFT mit WebSphere Application Server traditional 9.0 installieren*

Führen Sie die folgenden Anweisungen aus, um die Java Platform, Enterprise Edition (Java EE)-Datenbankprotokollfunktion für Managed File Transfer mit WebSphere Application Server traditional 9.0 zu installieren und zu konfigurieren.

## Vorbereitende Schritte

Befolgen Sie vor der Installation der Anwendung der JEE-Datenbankprotokollfunktion die Anweisungen in den Abschnitten „Vorbereiten der Installation der Java EE-Datenbankprotokollfunktion für MFT” auf Seite 858 und „Pfad der nativen Bibliothek in WebSphere Application Server traditional 9.0 festlegen” auf Seite 860.

## Informationen zu diesem Vorgang

Weitere Informationen zur Java EE-Datenbankprotokollfunktion finden Sie unter „MFT-Protokollfunktion konfigurieren” auf Seite 843.

## Vorgehensweise

1. Konfigurieren Sie den XA-JDBC-Provider:
   a) Wählen Sie in der Navigation der WebSphere Application Server traditional 9.0 -Administrationskonsole **Ressourcen** > **JDBC** > **JDBC Provider** aus.
   b) Erstellen Sie einen JDBC-Provider mit dem Konsolenassistenten, indem Sie auf **Neu** klicken.
   c) Wählen Sie in Schritt 1 des Assistenten in der Liste **Datenbanktyp** die Datenbank aus, die Sie verwenden, und geben Sie in der Liste **Providertyp** den zugehörigen Providertyp an. Wählen Sie in der Liste **Implementierungstyp** die Option **XA-Datenquelle** aus. Klicken Sie auf **Weiter (Next)**.

      
      
      Sie können einen Verweis auf `db2jcc_license_cisuz.jar` entfernen und `db2jcc.jar` auf `db2jcc4.jar` ändern, d. h. die Version der JAR-Datei, die mit der neuesten Version von Db2 geliefert wird, oder Ihre lokale Version.
   d) Stellen Sie in Schritt 2 des Assistenten sicher, dass die Verzeichnisposition der erforderlichen JAR-Dateien der Datenbank ordnungsgemäß festgelegt ist. Klicken Sie auf **Weiter**.
   e) Klicken Sie auf der Übersichtsseite auf **Fertig stellen** , um den JDBC-Provider zu erstellen.
2. Erstellen Sie Authentifizierungsaliasnamen. Erstellen Sie einen Alias für die Datenquelle und einen anderen für IBM MQ:
   a) Wählen Sie **Sicherheit** > **Globale Sicherheit** in der Navigation der WebSphere Application Server traditional 9.0 -Administrationskonsole aus.
   b) Erweitern Sie unter der Überschrift **Authentifizierung** den Eintrag **Java Authentication and Authorization Service** .
   c) Klicken Sie auf **J2C-Authentifizierungsdaten** . Die Authentifizierungsaliasseite wird geöffnet.
   d) Erstellen Sie einen Authentifizierungsalias für Ihre Datenquelle:
      i) Klicken Sie auf **Neu**.
      ii) Geben Sie die Details für **Alias** , **Benutzer-ID** , **Kennwort** und **Beschreibung** ein. Die Details, die in den Feldern **Benutzer-ID** und **Kennwort** eingegeben werden, müssen mit den Details übereinstimmen, die Sie bei der Erstellung des Datenbankbenutzers eingegeben haben. Weitere Informationen finden Sie unter „Benutzerzugriff für die Java EE-Datenbankprotokollfunktion für MFT konfigurieren” auf Seite 866.
      iii) Klicken Sie auf **OK**.
   e) Erstellen Sie ein Authentifizierungsalias für IBM MQ:
      i) Klicken Sie auf **Neu**.
      ii) Geben Sie die Details für **Alias** , **Benutzer-ID** , **Kennwort** und **Beschreibung** ein. Die Details, die Sie in den Feldern für **Benutzer-ID** und **Kennwort** eingeben, müssen mit der Benutzer- und Kennworteinstellung Ihrer IBM MQ-Installation übereinstimmen.
      iii) Klicken Sie auf **OK**.
3. Erstellen Sie eine Datenquelle:
   a) Wählen Sie **Ressourcen** > **JDBC** > **Datenquellen** in der Navigation der WebSphere Application Server traditional 9.0 -Administrationskonsole aus.
   b) Wählen Sie die Dropdown-Liste **Geltungsbereich** aus, und ändern Sie den Geltungsbereich in den entsprechenden Wert. Beispiel: `Node=yourNode, Server=yourServer`.
   c) Erstellen Sie eine Datenquelle mit dem Konsolenassistenten, indem Sie auf **Neu** klicken.
   d) Geben Sie in Schritt 1 des Assistenten im Feld **Datenquellenname** den Wert `wmqfte-database` ein und geben Sie im Feld **JNDI-Name** den Wert `jdbc/wmqfte-database` ein. Klicken Sie auf **Weiter (Next)**.
   e) Verwenden Sie in Schritt 2 des Assistenten die Dropdown-Liste **Vorhandenen JDBC-Provider auswählen** , um den JDBC-Provider auszuwählen, der in den vorherigen Schritten erstellt wurde. Klicken Sie auf **Weiter**.
   f) **Db2:** Geben Sie in Schritt 3 des Assistenten im Feld **Treibertyp** 4ein.

g) **Db2:** Geben Sie in den Feldern **Database name** (Datenbankname), **Server name** (Servername) und **Port number** (Portnummer) die entsprechenden Informationen ein und klicken Sie auf **Next** (Weiter).

**Oracle:** Geben Sie die Verbindungs-URL in das Feld **URL** ein und wählen Sie den richtigen Datenspeicher-Helper im Feld **Name der Helper-Klasse für Datenspeicher** aus.

**Oracle RAC:** Wenn eine Verbindung zu einem Oracle Real Application Cluster hergestellt wird, muss die Verbindungs-URL die Hostinformationen enthalten, die erforderlich sind, um eine Verbindung zu allen verfügbaren Instanzen der Datenbank herzustellen.

h) Wählen Sie in Schritt 4 des Assistenten den Namen des Datenquellenauthentifizierungsalias aus, den Sie in Schritt 2d in der Liste **Authentifizierungsalias für XA-Wiederherstellung** definiert haben. Wählen Sie denselben Namen aus den Listen **Aliasname der komponentengesteuerten Authentifizierung** und **Aliasname für containergesteuerte Authentifizierung** aus.

i) Klicken Sie auf der Zusammenfassungsseite auf **Fertig stellen** , um die Datenquelle zu erstellen.

4. Optional: Überprüfen Sie die Konfiguration der Datenquelle:
   a) Wählen Sie **Ressourcen** > **JDBC** > **Datenquellen** in der Navigation der WebSphere Application Server traditional 9.0 -Administrationskonsole aus.
   b) Klicken Sie auf die Schaltfläche **Verbindung testen** .
5. Erstellen Sie ein Thema.
   a) Klicken Sie in der Navigation der WebSphere Application Server traditional 9.0-Administrationskonsole nacheinander auf **Ressourcen** > **JMS** > **Themen**.
   b) Wählen Sie die Dropdown-Liste **Geltungsbereich** aus, und ändern Sie den Geltungsbereich in den entsprechenden Wert. Beispiel: `Node=yourNode, Server=yourServer`.
   c) Klicken Sie auf **Neu**.
   d) Klicken Sie auf **IBM MQ-Messaging-Provider** .
   e) Wählen Sie in der Anzeige **Verwaltung** der Eigenschaftenseite für das Thema eindeutige Werte für die Felder **Name** und **JNDI-Name** aus, auf die Sie später in der Konfiguration verweisen werden.
   f) Geben Sie in der Anzeige **IBM MQ Thema** `SYSTEM.FTE/Log/#` in das Feld **Themenname** ein.
6. Erstellen Sie eine Aktivierungsspezifikation:
   a) Klicken Sie in der Navigation der WebSphere Application Server traditional 9.0-Administrationskonsole nacheinander auf **Ressourcen** > **JMS** > **Aktivierungsspezifikation**.
   b) Wählen Sie die Dropdown-Liste **Geltungsbereich** aus, und ändern Sie den Geltungsbereich in den entsprechenden Wert. Beispiel: `Node=yourNode, Server=yourServer`.
   c) Klicken Sie auf **Neu**.
   d) Klicken Sie auf **IBM MQ-Messaging-Provider** .
   e) Wählen Sie in Schritt 1 des Assistenten eindeutige Werte für die Felder **Name** und **JNDI-Name** aus, die Sie später in der Konfiguration erneut referenzieren werden.
   f) Geben Sie in Schritt 1.1 den JNDI-Namen für das Thema ein, das Sie in Schritt 5 im Feld **Destination-JNDI-Name** konfiguriert haben.
   g) Wählen Sie in der Liste **Zieltyp** die Option **Thema** aus.
   h) Wählen Sie in Schritt 1.2 des Assistenten **Permanent Subscription** aus. Geben Sie `SYSTEM.FTE.DATABASELOGGER.AUTO` in das Feld **Subskriptionsname** ein.
   i) Wählen Sie in Schritt 2 des Assistenten die Option **Geben Sie alle erforderlichen Informationen in diesen Assistenten ein** aus.
   j) Geben Sie in Schritt 2.1 den Namen Ihres Warteschlangenmanagers im Feld **Queue manager or queue sharing group name** (Name des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange) ein.
   k) Wählen Sie in Schritt 2.2 die von Ihnen ausgewählte Transportmethode aus der Liste **Transport** aus. Wenn Sie **Bindings** auswählen, sind keine weiteren Informationen erforderlich. Wenn Sie

**Client** oder **Bindings then client** auswählen, geben Sie die Details für **Hostname** , **Port** und **Serververbindungskanal** ein.

l) Optional: Klicken Sie auf **Verbindung testen** , um die Bestätigung des Warteschlangenmanagers zu bestätigen. Sie können jedoch den Empfang von `NOT_AUTHORIZED` erwarten, bis Sie den Authentifizierungsalias in Schritt 6nreferenziert haben.

m) Klicken Sie auf **Speichern**.

n) Klicken Sie auf den Namen der von Ihnen erstellten Aktivierungsspezifikation. Blättern Sie im Abschnitt **Allgemeine Eigenschaften** auf der Registerkarte **Konfiguration** abwärts zur Anzeige **Erweitert** und geben Sie im Feld **Client-ID** einen eindeutigen Namen für Ihre IBM MQ -Verbindung ein. Wenn Sie diesen Schritt nicht ausführen, wird Ihre Verbindung von IBM MQ mit dem Fehlercode `JMSCC0101` abgelehnt.

o) Wenn Sie **Client** als Transportmethode ausgewählt haben, blättern Sie in die Anzeige **Sicherheitseinstellungen** und wählen Sie den Authentifizierungsalias aus, den Sie in Schritt 8 in der Liste **Authentifizierungsalias** definiert haben.

p) Klicken Sie auf **Anwenden**.

q) Klicken Sie im Abschnitt **Weitere Eigenschaften** auf der Registerkarte **Konfiguration** auf **Erweiterte Eigenschaften** . Geben Sie im Abschnitt **Verbindungskonsument** der Anzeige **Erweiterte Eigenschaften** 1 in das Feld **Maximale Anzahl Serversitzungen** ein.

**Anmerkung:** Stellen Sie sicher, dass Sie diesen Schritt ausführen, bevor Sie fortfahren Wenn dies nicht der Fehler ist, kann die Protokollfunktion nicht ordnungsgemäß funktionieren.

r) Klicken Sie im Abschnitt **Weitere Eigenschaften** auf der Registerkarte **Konfiguration** auf **Erweiterte Eigenschaften** . Setzen Sie den Wert für **Endpunkt stoppen, wenn Nachrichtenzustellung fehlschlägt** auf mindestens 1.

Wenn der Wert der Eigenschaft **`_numberOfFailedAttemptsBeforeReject`** höher als 1 ist (weitere Informationen finden Sie in 9j), muss **Stop endpoint if message delivery fails** (Endpunkt bei Fehlschlagen der Nachrichtenzustellung stoppen) mindestens auf den gleichen Wert wie die Eigenschaft **`_numberOfFailedAttemptsBeforeReject`** gesetzt werden. Dadurch wird verhindert, dass der Endpunkt gestoppt wird, wenn eine Nachricht empfangen wird, die nicht verarbeitet werden kann (z. B. eine fehlerhafte Übertragungsprotokollnachricht). Weitere Informationen finden Sie in der MFT Fehlerbehandlung und Zurückweisung Protokollfunktion.

7. Erstellen Sie eine Warteschlangenverbindungsfactory.

a) Klicken Sie in der Navigation der WebSphere Application Server traditional 9.0-Administrationskonsole nacheinander auf **Ressourcen** > **JMS** > **Warteschlangenverbindungsfactorys**.

b) Wählen Sie die Dropdown-Liste **Geltungsbereich** aus, und ändern Sie den Geltungsbereich in den entsprechenden Wert. Beispiel: `Node=yourNode, Server=yourServer`.

c) Klicken Sie auf **Neu**.

d) Klicken Sie auf **IBM MQ-Messaging-Provider** .

e) Wählen Sie in Schritt 1 des Assistenten eindeutige Werte für die Felder **Name** und **JNDI-Name** aus, die Sie später in der Konfiguration erneut referenzieren werden.

f) Wählen Sie in Schritt 2 die Option **Geben Sie alle erforderlichen Informationen in diesen Assistenten ein** aus.

g) Geben Sie in Schritt 2.1 den Namen Ihres Warteschlangenmanagers im Feld **Queue manager or queue sharing group name** (Name des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange) ein.

h) Wählen Sie in Schritt 2.2 die von Ihnen ausgewählte Transportmethode aus der Liste **Transport** aus. Wenn Sie **Bindings** auswählen, sind keine weiteren Informationen erforderlich. Wenn Sie **Client** oder **Bindings then client** auswählen, geben Sie die Details für **Hostname** , **Port** und **Serververbindungskanal** ein.

i) Optional: Klicken Sie auf **Verbindung testen** , um die Bestätigung des Warteschlangenmanagers zu bestätigen. Sie können jedoch erwarten, dass `NOT_AUTHORIZED` empfangen wird, bis Sie den Authentifizierungsalias in Schritt 7hreferenziert haben.

j) Wenn Sie **Client** oder **Bindings then Client** als Transportmethode ausgewählt haben, klicken Sie auf den Namen der gerade erstellten Warteschlangenverbindungs-Factory. Blättern Sie in die Anzeige **Sicherheitseinstellungen** der Registerkarte **Konfiguration** , und wählen Sie den Authentifizierungsalias aus, den Sie in Schritt 2e in den Listen **Authentifizierungsalias für XA-Wiederherstellung** und **Aliasname für containergesteuerte Authentifizierung** definiert haben.

8. Erstellen Sie in WebSphere Application Server eine Ablehnungswarteschlange:
   a) Klicken Sie in der Navigation der WebSphere Application Server traditional 9.0-Administrationskonsole nacheinander auf **Ressourcen** > **JMS** > **Warteschlangen**.
   b) Wählen Sie die Dropdown-Liste **Geltungsbereich** aus, und ändern Sie den Geltungsbereich in den entsprechenden Wert. Beispiel: `Node=yourNode, Server=yourServer`.
   c) Klicken Sie auf **Neu**.
   d) Klicken Sie auf **IBM MQ-Messaging-Provider** .
   e) Wählen Sie eindeutige Werte für die Felder **Name** und **JNDI-Name** aus, auf die Sie später in der Konfiguration erneut verweisen werden.
   f) Geben Sie SYSTEM.FTE.LOG.RJCT.*logger_name* im Feld **Warteschlangenname** ein. Stellen Sie sicher, dass Sie diese Warteschlange in Ihrem Koordinationswarteschlangenmanager erstellt haben.
   g) Geben Sie den Namen Ihres WS-Managers in das Feld **Name des Warteschlangenmanagers** ein.
   h) Klicken Sie auf **OK**.
9. Installieren Sie die JEE-Datenbank-Logger-Anwendung:
   a) Wählen Sie in der WebSphere Application Server traditional 9.0-Administrationskonsole nacheinander **Anwendungen** > **Neue Anwendung**.
   b) Wählen Sie die Dropdown-Liste **Geltungsbereich** aus, und ändern Sie den Geltungsbereich in den entsprechenden Wert. Beispiel: `Node=yourNode, Server=yourServer`.
   c) Wählen Sie in der Optionsliste die Option **Neue Unternehmensanwendung** aus.
   d) Wählen Sie auf der **Vorbereiten der Anwendungsinstallation**-Seite die Datei `com.ibm.wmqfte.databaselogger.jee.ear` oder die Datei `com.ibm.wmqfte.databaselogger.jee.oracle.ear` aus dem Verzeichnis *MQ_INSTALLATION_PATH*`/mqft/web` der Installation von Managed File Transfer Service aus und klicken Sie auf **Weiter**.
   e) Wählen Sie in der folgenden Anzeige **Detailliert** aus, um alle Installationsoptionen und -parameter anzuzeigen, und klicken Sie auf **Weiter** .
   f) Klicken Sie auf **Weiter** durch die Schritte 1-4, um die Standardwerte zu übernehmen.
   g) Blättern Sie in Schritt 5 des Assistenten, **Listener für nachrichtengesteuerte Beans binden** , zum Abschnitt **Listener Bindings** (Listener-Bindungen). Klicken Sie auf **Aktivierungsspezifikation (Activation Specification)**.

      Geben Sie die erforderlichen Werte für die folgenden Felder ein:

      **JNDI-Name der Zielressource**
      Der JNDI-Name, den Sie bei der Erstellung einer Aktivierungsspezifikation in Schritt 6d angegeben haben.

      **JNDI-Name des Ziels**
      Der JNDI-Name, den Sie bei der Erstellung eines Themas in Schritt 5d angegeben haben.

      Klicken Sie auf **Weiter**.
   h) Geben Sie in Schritt 6 des Assistenten, **Ressourcenreferenzen zu Ressourcen zuordnen** , die Details in das Feld **JNDI-Name der Zielressource** ein. Dieser Name ist der JNDI-Name, den Sie in Schritt 7c für die Verbindungs-Factory für Zurückweisungswarteschlangen angegeben haben. Klicken Sie auf **Weiter (Next)**.
   i) Geben Sie in Schritt 7 des Assistenten **Ressourcenumgebungseinträge in Ressourcen zuordnen** die Details in das Feld **JNDI-Name der Zielressource** ein. Dieser Name ist der JNDI-Name der Zurückweisungswarteschlange, die Sie in Schritt 8d erstellt haben. Klicken Sie auf **Weiter (Next)**.

j) Akzeptieren Sie in Schritt 8 des Assistenten **Umgebungseinträge für EJB-Module zuordnen** den Standardwert 1. Klicken Sie auf **Weiter**.

**Oracle RAC:** Wenn Sie eine Verbindung zu einem Oracle Real Application Cluster herstellen, müssen Sie den Wert für die Eigenschaft "_numberOfFailedAttemptsBeforeReject" auf **mindestens** 2 setzen. Diese Eigenschaft bestimmt die Anzahl der Versuche der Protokollfunktion, eine Prüfnachricht zu verarbeiten, nachdem ein Fehler aufgetreten ist. Bei einer Datenbankübernahmefunktion ist wahrscheinlich mindestens ein Fehler aufgetreten. Um zu vermeiden, dass eine Nachricht unnötigerweise in die Zurückweisungswarteschlange verschoben wird, kann durch eine Erhöhung dieses Werts ein zweiter Versuch unternommen werden, was in der Regel zu einem Erfolg führt, da eine Verbindung zur neuen Datenbankinstanz hergestellt wird. Wenn Sie während des Tests feststellen, dass Nachrichten während des Failovers Ihrer Datenbankinstanz immer noch in die Zurückweisungswarteschlange verschoben werden, erhöhen Sie diesen Wert weiter: Die Ablaufsteuerung des Switch zwischen den Instanzen kann zu mehr als einem Fehler für dieselbe Nachricht führen. Es ist jedoch zu beachten, dass die Erhöhung dieses Werts alle Fehlerfälle (z. B. eine fehlerhafte Nachricht) und nicht nur die Datenbankübernahme betrifft, sodass der Wert mit Vorsicht erhöht wird, um unnötige Neuversuche zu vermeiden.

k) Klicken Sie in Schritt 9 des Assistenten, **Metadaten für Module** , auf **Weiter** .

l) Klicken Sie in Schritt 10 des Assistenten, **Zusammenfassung** , auf **Fertig stellen** .

10. Sie können die Anwendung über die Administrationskonsole von WebSphere Application Server traditional 9.0 starten:
    a) Wählen Sie in der Konsolennavigation nacheinander **Anwendungen** > **Anwendungstypen** > **WebSphere Enterprise-Anwendungen**.
    b) Wählen Sie das Kontrollkästchen für die Unternehmensanwendung **Logger** aus der Objektgruppentabelle aus und klicken Sie auf **Starten** .

### *Benutzerzugriff für die Java EE-Datenbankprotokollfunktion für MFT konfigurieren*

Beim Konfigurieren der Datenbankprotokollfunktion für Java Platform, Enterprise Edition (Java EE) für Managed File Transfer benötigen Sie Benutzerkonten für den Zugriff auf IBM MQ, Ihr Datenbank- und Ihr Betriebssystem. Die Anzahl der erforderlichen Betriebssystembenutzer hängt von der Anzahl der Systeme ab, die Sie zum Hosten dieser Komponenten verwenden.

## Informationen zu diesem Vorgang

Anzahl und Typ der zur Ausführung der Java EE-Datenbankprotokollfunktion benötigten Benutzerkonten hängt von der Anzahl der Systeme ab, die eingesetzt werden. Benutzerkonten sind erforderlich, um auf die folgenden drei Umgebungen zuzugreifen:

- Lokales Betriebssystem
- IBM MQ
- Datenbank

Sie können die JEE-Datenbankprotokollfunktion, IBM MQ und die Datenbank auf einem einzigen System oder auf zwei Systemen verteilt installieren. Die Komponenten können in den folgenden Beispieltopologien installiert werden:

**Java EE-Datenbankprotokollfunktion, IBM MQ und Datenbank alle auf demselben System**
Sie können einen einzelnen Betriebssystembenutzer für die Nutzung aller drei Komponenten definieren. Die Protokollfunktion verwendet den Bindungsmodus für die Verbindung mit IBM MQ und eine native Verbindung für den Zugriff auf die Datenbank.

**Java EE-Datenbankprotokollfunktion und IBM MQ auf einem System, die Datenbank auf einem separaten System**
Für diese Konfiguration werden zwei Benutzer erstellt, einmal ein Betriebssystembenutzer auf dem System, auf dem die Protokollfunktion aktiv ist, einmal ein Betriebssystembenutzer mit Remotezugriff auf die Datenbank auf dem Datenbankserver. Die Protokollfunktion verwendet den Bindungsmodus für die Verbindung mit IBM MQ und eine Clientverbindung für den Zugriff auf die Datenbank.

**Java EE-Datenbankprotokollfunktion auf einem System, IBM MQ auf einem anderen System, die Datenbank auf noch einem anderen System**

Für diese Konfiguration erstellen Sie drei Benutzer: einen Betriebssystembenutzer für den Start des Anwendungsservers, einen IBM MQ-Benutzer für den Zugriff auf die verwendeten Warteschlangen und Themen und einen Datenbankserverbenutzer für den Zugriff auf die Datenbanktabellen sowie deren Bearbeitung. Die Protokollfunktion verwendet den Clientmodus für den Zugriff auf IBM MQ und eine Clientverbindung für den Zugriff auf die Datenbank.

Bei den übrigen Anweisungen wird beispielsweise davon ausgegangen, dass der Benutzer den Namen `ftelog` hat. Sie können jedoch einen beliebigen neuen oder vorhandenen Benutzernamen verwenden. Konfigurieren Sie die Berechtigungen des Benutzers wie folgt:

## Vorgehensweise

1. Stellen Sie sicher, dass der Betriebssystembenutzer über eine eigene Gruppe verfügt und kein Mitglied von Gruppen mit umfassenden Berechtigungen auf dem Koordinationswarteschlangenmanager ist. Der Benutzer sollte nicht der Gruppe 'mqm' angehören. Auf einigen Plattformen wird der Mitarbeitergruppe automatisch auch Warteschlangenmanagerzugriff erteilt; die Protokollfunktion sollte nicht zur Mitarbeitergruppe gehören. In IBM MQ Explorer können Sie die Berechtigungssätze für den Warteschlangenmanager selbst sowie für die Objekte auf dem Warteschlangenmanager anzeigen. Klicken Sie mit der rechten Maustaste auf das Objekt, und wählen Sie **Objektberechtigungen > Berechtigungsdatensätze verwalten** aus. In der Befehlszeile können Sie die Befehle dspmqaut (Anzeigeberechtigung) oder dmpmqaut (Speicherauszugsberechtigung) verwenden.
2. Verwenden Sie das Fenster **Manage Authority Records** (Berechtigungsdatensätze verwalten) im IBM MQ Explorer oder den Befehl setmqaut (Berechtigung erteilen oder entziehen), um Berechtigungen für die eigene Gruppe des IBM MQ-Benutzers hinzuzufügen (unter AIX sind IBM MQ-Berechtigungen nur Gruppen zugeordnet, nicht einzelnen Benutzer). Folgende Berechtigungen sind erforderlich:
   - Verbindungsberechtigung (CONNECT) und Abfrageberechtigung (INQUIRE) auf dem Warteschlangenmanager (für die IBM MQ Java-Bibliotheken ist die Abfrageberechtigung INQUIRE erforderlich).
   - Subskriptionsberechtigung (SUBSCRIBE) für das Thema SYSTEM.FTE.
   - PUT-Berechtigung für die Warteschlange SYSTEM.FTE.LOG.RJCT.*Name_der_Protokollfunktion*.

   Die oben angegebenen Zurückweisungs- und Befehlswarteschlangennamen sind die Standardnamen. Wenn Sie bei der Konfiguration der Warteschlangen für die Protokollfunktion andere Namen für die Warteschlangen angegeben haben, müssen Sie die Berechtigungen diesen Namen zuordnen.
3. Führen Sie die Datenbankbenutzerkonfiguration aus, die für die von Ihnen verwendete Datenbank bestimmt ist.
   - Bei Verwendung einer Db2-Datenbank müssen Sie die folgenden Schritte ausführen:

     **Anmerkung:** Es gibt verschiedene Mechanismen für die Verwaltung von Datenbankbenutzern mit Db2. Diese Anweisungen gelten für das Standardschema, das auf Betriebssystembenutzern basiert.

     – Stellen Sie sicher, dass der Benutzer `ftelog` keiner Db2 -Verwaltungsgruppe (z. B. db2iadm1, db2fadm1oder dasadm1) zugeordnet ist.

     – Erteilen Sie dem Benutzer die Berechtigung für die Verbindung mit der Datenbank und die Berechtigung zum Auswählen, Einfügen und Aktualisieren in den Tabellen, die Sie im Rahmen von Schritt 2: Erforderliche Datenbanktabellen erstellen erstellt haben.
   - Bei einer Oracle-Datenbank müssen Sie die folgenden Schritte ausführen:

     – Stellen Sie sicher, dass sich der Benutzer `ftelog` in keiner Oracle -Verwaltungsgruppe befindet (z. B. ora_dba unter Windows oder dba unter AIX and Linux).

     – Erteilen Sie dem Benutzer die Berechtigung, eine Verbindung zu der Datenbank herzustellen, und die Berechtigung zum Auswählen, Einfügen und Aktualisieren in den Tabellen, die Sie im Rahmen von Schritt 2: Erforderliche Datenbanktabellen erstellen erstellt haben.

## Migration von der eigenständigen Datenbankprotokollfunktion in die Java EE-Datenbankprotokollfunktion für MFT

Sie können von der eigenständigen Datenbankprotokollfunktion auf die Java EE-Datenbankprotokollfunktion migrieren. Sie müssen die eigenständige Datenbankprotokollfunktion stoppen und die JEE-Datenbankprotokollfunktion installieren. Damit Protokolleinträge nicht verloren gehen oder dupliziert werden, müssen Sie die Veröffentlichung von Nachrichten im SYSTEM.FTE vor dem Stoppen der eigenständigen Datenbankprotokollfunktion und nach der Installation der Java EE -Datenbankprotokollfunktion einen Neustart durchführen. Sichern Sie Ihre Datenbank vor der Migration.

### Informationen zu diesem Vorgang

### Vorgehensweise

1. Führen Sie vor dem Stoppen der Datenbank folgenden MQSC-Befehl an Ihrem Koordinationswarteschlangenmanager aus: `ALTER QM PSMODE(COMPAT)`

   Dadurch werden Nachrichten gestoppt, die gerade für das Thema SYSTEM.FTE/Log veröffentlicht werden. Warten Sie, bis alle Nachrichten in Zusammenhang mit der Subskription von der Protokollfunktion verarbeitet wurden. Diese Subskription hat standardmäßig den Namen SYSTEM.FTE.LOGGER.AUTO.
2. Stoppen Sie die Datenbankprotokollfunktion mit dem Befehl **`fteStopLogger`**.
3. Sichern Sie die Datenbank mit den Tools, die mit der Datenbanksoftware bereitgestellt werden.
4. Löschen Sie die Subskription, die zur eigenständigen Datenbankprotokollfunktion gehört.

   Diese Subskription hat standardmäßig den Namen SYSTEM.FTE.LOGGER.AUTO.
5. Weist Ihr Datenbankschema eine frühere Version auf, müssen Sie es der Reihe nach auf die einzelnen nachfolgenden Stufen migrieren. Hat Ihr Datenbankschema beispielsweise die Version 7.0.1 und Sie möchten auf Version 7.0.4 migrieren, müssen Sie zunächst eine Migration von V7.0.1 auf V7.0.2, dann von V7.0.2 auf V7.0.3 und anschließend von V7.0.3 auf V7.0.4 durchführen. Migrieren Sie Ihr Datenbankschema von Version *old* auf Version *new*, wobei *old* und *new* Variablen sind, die eine Schemaversion beschreiben, indem Sie die eine der folgenden Aktionen für jede Version des Schemas ausführen, die Sie migrieren müssen:

   - z/OS Wenn Sie als Datenbank eine Db2-Datenbank unter z/OS verwenden und das Schema von Version 7.0.2 auf Version 7.0.3 oder von Version 7.0.3 auf Version 7.0.4 migrieren möchten, müssen Sie ein neues Datenbankschema erstellen und Ihre vorhandenen Schemadaten in dieses Schema kopieren. Weitere Informationen finden Sie in der Db2-Dokumentation.
   - Haben Sie keine Db2-Datenbank oder haben Sie Ihre Datenbank mit einer Seitengröße von mehr als 8 K erstellt, können Sie das Schema wie andere Versionen migrieren, indem Sie die nachfolgend aufgeführten Schritte ausführen.
   - Wenn Sie unter anderen Umständen zwischen Datenbanktabellen migrieren, führen Sie die folgenden Schritte aus:

     a. Wählen Sie die Datei aus, die für Ihre Datenbankplattform geeignet ist und deren Name die Zeichenfolge *old*-*new*enthält. Diese Datei befindet sich im Verzeichnis *MQ_INSTALLATION_PATH*`/mqft/sql` der Installation von Remote Tools and Documentation.

     b. Falls Sie am ursprünglichen Schema Änderungen vorgenommen haben, sollten Sie die Migrationsdatei überprüfen, um sicherzustellen, dass sie mit der geänderten Datenbank kompatibel ist.

     c. Führen Sie die SQL-Datei auf Ihrer Datenbank aus.
6. Installieren Sie die EAR-Datei der Java EE-Datenbankprotokollfunktion.
7. Implementieren Sie die Java EE-Datenbankprotokollfunktion. Weitere Informationen finden Sie unter „Java EE-Datenbankprotokollfunktion für MFT installieren” auf Seite 857.
8. Führen Sie folgenden MQSC-Befehl für Ihren Koordinationswarteschlangenmanager aus: `ALTER QMGR PSMODE(ENABLED)`

   Dadurch wird die Veröffentlichung von Nachrichten für das Thema SYSTEM.FTE/Log ermöglicht.

### Ergebnisse

## Connect:Direct-Bridge konfigurieren

Sie können die Connect:Direct-Bridge für die Übertragung von Dateien zwischen einem Managed File Transfer- und einem Connect:Direct-Netz konfigurieren. Die Connect:Direct-Bridge setzt sich aus einem Connect:Direct-Knoten und einem Managed File Transfer- Agenten zusammen, der für die Kommunikation mit diesem Knoten bestimmt ist. Dieser Agent wird als Connect:Direct-Bridgeagent bezeichnet.

### Vorbereitende Schritte

Der Agent und der Knoten, die zusammen die Connect:Direct-Bridge bilden, müssen sich auf demselben System befinden oder Zugriff auf dasselbe Dateisystem (beispielsweise über einen gemeinsam genutzten NFS-Mount) haben. Dieses Dateisystem wird verwendet, um Dateien während Dateiübertragungen, die die Connect:Direct-Bridge einbeziehen, temporär in einem Verzeichnis zu speichern, das durch den Parameter **cdTmpDir** definiert ist. Der Connect:Direct-Bridgeagent und der Connect:Direct-Bridgeknoten müssen auf dieses Verzeichnis unter Verwendung desselben Pfadnamens zugreifen können. Wenn sich Agent und Knoten beispielsweise jeweils auf einem eigenen Windows-System befinden, muss zum Anhängen des gemeinsam genutzten Dateisystems für beide Systeme derselbe Laufwerkbuchstabe verwendet werden. Bei den folgenden Konfigurationen können der Agent und der Knoten denselben Pfadnamen verwenden:

- Agent und Knoten befinden sich auf demselben System, das unter Windows oder Linux for x86-64 betrieben wird.
- Der Agent befindet sich auf einem System mit Linux for x86-64, der Knoten auf einem AIX-System.
- Der Agent befindet sich auf einem Windows-System, der Knoten auf einem anderen Windows-System.

Bei den folgenden Konfigurationen können der Agent und der Knoten nicht denselben Pfadnamen verwenden:

- Der Agent befindet sich auf einem System mit Linux for x86-64, der Knoten auf einem Windows-System.
- Der Agent befindet sich auf einem Windows-System, der Knoten auf einem UNIX-System.

Diese Einschränkungen sollten Sie bei der Planung der Connect:Direct-Bridgeinstallation bedenken.

Weitere Informationen zu den Betriebssystemversionen, die für die Connect:Direct-Bridge unterstützt werden, finden Sie auf der Webseite Systemvoraussetzungen für IBM MQ.

### Informationen zu diesem Vorgang

Ein Connect:Direct-Bridgeagent ist ein Managed File Transfer-Agent, der für die Kommunikation mit einem Connect:Direct-Knoten gedacht ist.

Für die Verbindung mit dem Connect:Direct-Knoten verwendet der Connect:Direct-Bridgeagent standardmäßig das TCP/IP-Protokoll. Wenn Sie eine sichere Verbindung zwischen Connect:Direct-Bridgeagenten und Connect:Direct-Knoten wünschen, können Sie auch das SSL- oder das TLS-Protokoll verwenden.

### Vorgehensweise

1. Wählen Sie Betriebssysteme für Connect:Direct-Bridgeagenten und -Bridgeknoten aus.
   a) Wählen Sie ein System, das entweder Windows oder Linux auf x86-64 verwendet, um darauf den Connect:Direct-Bridge-Agent zu installieren.
   b) Wählen Sie ein Betriebssystem aus, das von Connect:Direct für Windows oder Connect:Direct für UNIX unterstützt wird, um den Connect:Direct -Bridgeknoten zu installieren.
2. Wählen Sie einen Connect:Direct-Knoten aus und konfigurieren Sie diesen.

   Vergewissern Sie sich vor der Ausführung dieser Anweisungen, dass ein Connect:Direct-Knoten installiert ist.

a) Wählen Sie einen Connect:Direct-Knoten für den Managed File Transfer-Agenten aus, mit dem kommuniziert werden soll.

b) Überprüfen Sie die Netzmap im Hinblick auf Ihren gewählten Connect:Direct-Knoten. Falls die Netzmap Einträge für ferne Knoten enthält, die auf einem Windows-Betriebssystem ausgeführt werden, müssen Sie sicherstellen, dass in diesen Einträgen angegeben ist, dass die Knoten unter Windows ausgeführt werden.


Wenn der Connect:Direct-Knoten, den Sie für die Connect:Direct-Bridge ausgewählt haben, unter Windows ausgeführt wird, verwenden Sie den Connect:Direct-Requester, um die Netzmap zu bearbeiten. Stellen Sie sicher, dass das Feld **Betriebssystem** für alle fernen Knoten, die unter Windows ausgeführt werden, auf **Windows** gesetzt ist.

3. Erstellen und konfigurieren Sie einen Connect:Direct-Bridgeagenten.

a) Erstellen Sie einen Connect:Direct-Bridgeagenten mit dem Befehl **`fteCreateCDAgent`**.

- Sie müssen einen Wert für den Parameter **cdNode** angeben. Dieser Parameter gibt den Namen an, den der Agent für den zur Connect:Direct-Bridge gehörenden Connect:Direct-Knoten verwendet. Verwenden Sie den Namen des im vorherigen Abschnitt ausgewählten Connect:Direct-Knotens.
- Geben Sie Werte für die Parameter **`cdNodeHost`** und **`cdNodePort`** an. Diese Parameter definieren den Connect:Direct-Knoten, mit dem der Agent kommuniziert.

  Wenn Sie keinen Wert für den Parameter **`cdNodeHost`** angeben, wird der Hostname oder die IP-Adresse des lokalen Systems verwendet. Wenn Sie keinen Wert für den Parameter **`cdNodePort`** angeben, wird der Wert `1363` verwendet.
- Verwenden Sie optional die Informationen in fteCreateAgent, um zu ermitteln, ob Sie einen Wert für den Parameter **`cdTmpDir`** angeben müssen.

b) Ordnen Sie die von Managed File Transfer verwendeten Benutzerberechtigungsnachweise den Benutzerberechtigungsnachweisen auf einem Connect:Direct-Knoten zu. Sie können Berechtigungsnachweise mit einer der folgenden Methoden zuordnen:

- Erstellen Sie eine `ConnectDirectCredentials.xml`-Datei, um Informationen zur Berechtigungsnachweiszuordnung zu definieren. Weitere Informationen finden Sie unter „Berechtigungsnachweise für Connect:Direct unter Verwendung der Datei 'ConnectDirectCredentials.xml' zuordnen” auf Seite 871.
- Schreiben Sie einen Benutzerexit, um das Credential-Mapping für Ihre Connect:Direct-Bridge auszuführen. Weitere Informationen finden Sie unter „Berechtigungsnachweise für Connect:Direct mithilfe von Exitklassen zuordnen” auf Seite 874.

4. Konfigurieren Sie die `ConnectDirectNodeProperties.xml`-Datei so, dass sie Informationen zu den fernen Connect:Direct-Knoten enthält.

Vor der Ausführung der nachfolgenden Anweisungen müssen Sie einen Connect:Direct-Bridgeagenten erstellt haben.

Bearbeiten Sie die Schablone `ConnectDirectNodeProperties.xml` im Konfigurationsverzeichnis des Connect:Direct-Bridge-Agenten. Führen Sie für jeden Connect:Direct-Knoten bzw. für jede Knotengruppe, zu der Sie Informationen festlegen möchten, die folgenden Schritte aus:

a) Erstellen Sie innerhalb des nodeProperties-Elements ein Element node.

b) Fügen Sie dem Element node ein Attribut name hinzu. Geben Sie als Wert dieses Attributs ein Muster zum Abgleich der Namen eines oder mehrerer ferner Connect:Direct-Knoten ein.

c) Optional: Fügen Sie dem Element node ein Attribut pattern hinzu, das angibt, welche Art von Muster der Wert im Attribut name ist. Gültige Werte sind `regex` und `wildcard`. Die Standardoption ist `wildcard`.

d) Fügen Sie dem Element node ein Attribut type hinzu, das das Betriebssystem angibt, auf dem die fernen Connect:Direct-Knoten, die durch das Attribut name angegeben wurden, ausgeführt werden.

Folgende Werte sind gültig:

- `Windows` - Der Knoten wird unter Windows ausgeführt
- `UNIX` – der Knoten wird unter AIX and Linux ausgeführt.
- z/OS `z/OS`, `zos`, `os/390` oder `os390` - Der Knoten wird unter z/OS ausgeführt

  Bei dem Wert dieses Attributs wird die Groß-/Kleinschreibung nicht beachtet. Übertragungen an ferne Knoten unter anderen Betriebssystemen werden von der Connect:Direct-Bridge nicht unterstützt.

Weitere Informationen finden Sie im Abschnitt Connect:Direct-Knoteneigenschaftendateiformat.

5. Konfigurieren Sie eine sichere Verbindung zwischen dem Connect:Direct-Bridgeagenten und dem Connect:Direct-Knoten. Ein Beispiel für die Vorgehensweise finden Sie unter SSL oder TLS zwischen dem Connect: Direct-Bridge-Agenten und dem Connect: Direct-Knoten konfigurieren .

**Zugehörige Tasks**

Fehlerbehebung bei der Connect:Direct-Bridge

SSL oder TLS zwischen dem Connect: Direct-Bridge-Agenten und dem Connect: Direct-Knoten konfigurieren

Übertragen einer Datei an einen Connect: Direct-Knoten

Übertragen einer Datei von einem Connect: Direct-Knoten

z/OS Mehrere Dateien von einem Connect: Direct-Knoten übertragen

**Zugehörige Verweise**

Connect:Direct-Bridge

## Berechtigungsnachweise für Connect:Direct zuordnen

Benutzerberechtigungsnachweise in Managed File Transfer können den Berechtigungsnachweisen in einem Connect:Direct-Knoten entweder mithilfe der entsprechenden Standardfunktion des Connect:Direct-Bridgeagenten zugeordnet werden oder indem Sie einen eigenen Benutzerexit erstellen. Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Zuordnung der Benutzerberechtigungsnachweise durchführt.

**Zugehörige Tasks**

„Berechtigungsnachweise für Connect:Direct unter Verwendung der Datei 'ConnectDirectCredentials.xml' zuordnen” auf Seite 871

Benutzerberechtigungsnachweise in Managed File Transfer können mithilfe der entsprechenden Standardfunktion des Connect:Direct-Bridgeagenten den Benutzerberechtigungsnachweisen in Connect:Direct-Knoten zugeordnet werden. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Berechtigungsinformationen eingeben können.

„Berechtigungsnachweise für Connect:Direct mithilfe von Exitklassen zuordnen” auf Seite 874

Wenn Sie die Standardfunktion für Credential-Mapping des Connect:Direct -Bridgeagenten nicht verwenden wollen, können Sie Benutzerberechtigungsnachweise in Managed File Transfer Benutzerberechtigungsnachweisen auf einem Connect:Direct -Knoten zuordnen, indem Sie einen eigenen Benutzerexit schreiben. Wenn Sie Ihre eigenen Benutzerexits für die Zuordnung von Berechtigungsnachweisen konfigurieren, wird die standardmäßige Berechtigungszuordnungsfunktion inaktiviert.

**Zugehörige Verweise**

Schnittstelle 'CDCredentialExit.java'

Connect: Dateiformat für direkte Berechtigungsnachweise

### *Berechtigungsnachweise für Connect:Direct unter Verwendung der Datei 'ConnectDirectCredentials.xml' zuordnen*

Benutzerberechtigungsnachweise in Managed File Transfer können mithilfe der entsprechenden Standardfunktion des Connect:Direct-Bridgeagenten den Benutzerberechtigungsnachweisen in Connect:Direct-Knoten zugeordnet werden. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Berechtigungsinformationen eingeben können.

## Informationen zu diesem Vorgang

Nach der Erstellung eines Connect:Direct-Bridgeagenten mithilfe des Befehls **fteCreateCDAgent** muss manuell die Datei `ConnectDirectCredentials.xml` erstellt werden. Dieser Datei müssen Sie vor der Verwendung des Connect:Direct-Bridgeagenten Host-, Benutzer- und Berechtigungsinformationen hinzufügen. Weitere Informationen finden Sie unter Connect: Direct credentials file format . Diese Datei wird standardmäßig aus dem Ausgangsverzeichnis des aktuellen Benutzers geladen, z. B. `/home/fteuser/ConnectDirectCredentials.xml`. Wenn Sie eine andere Position verwenden möchten, geben Sie diese über das Element <credentialsFile> in der Datei `ConnectDirectNodeProperties.xml` an.

## Vorgehensweise

1. Stellen Sie sicher, dass das Attribut name im Element <tns:pnode name="Connect:Direct node host" pattern="wildcard"> den Wert des Namens des Connect:Direct -Knotens enthält, zu dem der Connect:Direct -Bridgeagent eine Verbindung herstellt. Dieser Wert muss mit dem Wert identisch sein, den Sie für den Parameter **fteCreateCDAgent -cdNode** angeben.

   Der Wert des Attributs `pattern` kann entweder `wildcard` oder `regex` sein. Wenn dieses Attribut nicht angegeben wird, ist der Standardwert `wildcard`.
2. Fügen Sie Benutzer-ID und Berechtigungsnachweisdaten als untergeordnete Elemente von <tns:pnode> in die Datei ein.

   Sie können eine oder mehrere Instanzen des folgenden <tns:user>-Elements in die Datei einfügen:

   ```
   <tns:user name="name"
             pattern="pattern"
             ignorecase="ignorecase"
             cdUserId="cdUserId"
             cdPassword="cdPassword"
             pnodeUserId="pnodeUserId"
             pnodePassword="pnodePassword">
   </tns:user>
   ```

   Dabei gilt:

   - *name* ist ein Muster, mit dem die MQMD-Benutzer-ID der MFT-Übertragungsanforderung verglichen wird.
   - *pattern* gibt an, ob das für das Attribut name angegebene Muster ein Platzhalterausdruck oder ein regulärer Java-Ausdruck ist. Der Wert des Attributs `pattern` kann entweder `wildcard` oder `regex` sein. Wenn dieses Attribut nicht angegeben wird, ist der Standardwert `wildcard`.
   - *ignorecase* gibt an, ob das Muster, das durch das Attribut name angegeben wird, als Groß-/Kleinschreibung beachtet werden soll. Wenn dieses Attribut nicht angegeben wird, ist der Standardwert `true`.
   - *cdUserId* ist die Benutzer-ID, mit der der Connect:Direct -Bridgeagent eine Verbindung zu dem Connect:Direct -Knoten herstellt, der durch das Attribut name des Elements <tns:pnode> angegeben ist. Stellen Sie sicher, dass es sich bei *cdUserId* nach Möglichkeit um eine Connect:Direct-Administrator-ID handelt. Kann *cdUserId* keine Connect:Direct-Administrator-ID sein, muss die Benutzer-ID, die stattdessen verwendet wird, im Connect:Direct-Bridgeknoten über die folgenden Funktionsberechtigungen verfügen:

     – Legen Sie für einen Windows-Knoten die folgenden Berechtigungen fest. Dieses Beispiel wird mit Wagenrückgaben formatiert, um die Lesbarkeit zu unterstützen:

     ```
     View Processes in the  value: yes
      TCQ
     Issue the copy         value: yes
      receive, copy send,
      run job, and run task
      Process statements
     Issue the submit       value: yes
      Process statement
     ```

```
Monitor, submit,       value: all
 change,and delete all
 Processes
Access Process         value: all
 statistics
Use the trace tool or  value: yes
 issue traceon and
 traceoff commands
Override Process       value: yes
 options such as file
 attributes and remote
 node ID
```

– Legen Sie für einen AIX- oder Linux-Knoten die folgenden Parameter in der `userfile.cfg`-Datei fest:

```
pstmt.copy             value: y
pstmt.upload           value: y
pstmt.download         value: y
pstmt.runjob           value: y
pstmt.runtask          value: y
cmd.submit             value: y
pstmt.submit           value: y
cmd.chgproc            value: y
cmd.delproc            value: y
cmd.flsproc            value: y
cmd.selproc            value: a
cmd.selstats           value: a
cmd.trace              value: y
snode.ovrd             value: y
```

- *cdPassword* ist das Kennwort, das der Benutzer-ID zugeordnet ist, die durch das Attribut `cdUserId` angegeben wird.
- Sie können optional das Attribut `pnodeUserId` angeben. Der Wert dieses Attributs ist die Benutzer-ID, die vom Connect:Direct -Knoten verwendet wird, der im Attribut `name` des Elements `<tns:pnode>` angegeben ist, um den Connect:Direct -Prozess zu übergeben. Wenn Sie das Attribut `pnodeUserId` nicht angeben, verwendet der Connect:Direct-Knoten die über das Attribut `cdUserId` angegebene Benutzer-ID zur Übergabe des Connect:Direct-Prozesses.
- Sie können optional das Attribut `pnodePassword` angeben. Der Wert dieses Attributs ist das Kennwort, das der Benutzer-ID zugeordnet ist, die durch das Attribut `pnodeUserId` angegeben wird.

Wenn kein Benutzerelement mit der MQMD-Benutzer-ID übereinstimmt, schlägt die Übertragung fehl.

3. Optional: Sie können ein oder mehrere `<tns:snode>`-Elemente als untergeordnete Elemente des Elements `<tns:user>` einschließen. Das Element `<tns:snode>` gibt Berechtigungsnachweise an, die von dem Connect:Direct-Knoten verwendet werden, der Teil der Connect:Direct-Bridge ist. Der Berechtigungsnachweis besteht aus der Benutzer-ID und dem Kennwort, die der Connect:Direct-Bridgeknoten zur Verbindung mit dem Connect:Direct-Knoten verwendet, der Quelle oder Ziel der Dateiübertragung ist.

   Fügen Sie eine oder mehrere der folgenden Elemente in die Datei ein:

```
<tns:snode name="name"
           pattern="pattern"
           userId="userId"
           password="password"/>
```

   Dabei gilt:

   - *name* ist ein Muster, das dem Namen des Connect:Direct-Knoten entsprechen soll, der Quelle oder Ziel der Dateiübertragung ist.
   - *pattern* gibt an, ob das für das Attribut `name` angegebene Muster ein Platzhalterausdruck oder ein regulärer Java-Ausdruck ist. Der Wert des Musterattributs kann entweder `wildcard` oder `regex` sein. Wenn dieses Attribut nicht angegeben wird, ist der Standardwert `wildcard`.
   - *userId* ist die Benutzer-ID, die vom Connect:Direct -Knoten verwendet wird, der im Attribut `name` des Elements `<tns:pnode>` angegeben ist, um eine Verbindung zu einem Connect:Direct -Knoten

herzustellen, der mit dem Muster übereinstimmt, das im Attribut `name` von `<tns:snode>`angegeben ist.

- *password* ist das Kennwort, das der Benutzer-ID zugeordnet ist, die durch das Attribut `userId` angegeben wird.

Wenn kein Element `<tns:snode>` mit dem Sekundärknoten der Dateiübertragung übereinstimmt, schlägt dies nicht zum Fehlschlagen der Übertragung vor. Die Übertragung wird gestartet, und es werden keine Benutzer-ID und kein Kennwort für die Verwendung mit dem Knoten "snode" angegeben.

### Ergebnisse

Bei der Suche nach einer Musterübereinstimmung im Fall von Benutzernamen oder Connect:Direct-Knotennamen durchsucht der Connect:Direct-Bridgeagent die Datei von oben nach unten. Die erste gefunden, die gefunden wird, wird verwendet.

**Zugehörige Tasks**

„Connect:Direct-Bridge konfigurieren” auf Seite 869

Sie können die Connect:Direct-Bridge für die Übertragung von Dateien zwischen einem Managed File Transfer- und einem Connect:Direct-Netz konfigurieren. Die Connect:Direct-Bridge setzt sich aus einem Connect:Direct-Knoten und einem Managed File Transfer- Agenten zusammen, der für die Kommunikation mit diesem Knoten bestimmt ist. Dieser Agent wird als Connect:Direct-Bridgeagent bezeichnet.

**Zugehörige Verweise**

Connect: Dateiformat für direkte Berechtigungsnachweise

fteCreateCDAgent: Connect:Direct-Bridgeagenten erstellen

### *Berechtigungsnachweise für Connect:Direct mithilfe von Exitklassen zuordnen*

Wenn Sie die Standardfunktion für Credential-Mapping des Connect:Direct -Bridgeagenten nicht verwenden wollen, können Sie Benutzerberechtigungsnachweise in Managed File Transfer Benutzerberechtigungsnachweisen auf einem Connect:Direct -Knoten zuordnen, indem Sie einen eigenen Benutzerexit schreiben. Wenn Sie Ihre eigenen Benutzerexits für die Zuordnung von Berechtigungsnachweisen konfigurieren, wird die standardmäßige Berechtigungszuordnungsfunktion inaktiviert.

### Informationen zu diesem Vorgang

Benutzerexits, die Sie zur Zuordnung von Connect:Direct-Berechtigungsnachweisen erstellen, müssen die Schnittstelle `com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit` implementieren. Weitere Informationen finden Sie im Abschnitt CDCredentialExit.java-Schnittstelle .

## IBM MQ Console und REST API konfigurieren

Der mqweb-Server, auf dem die IBM MQ Console und REST API ausgeführt werden, wird mit einer Standardkonfiguration bereitgestellt. Um eine dieser Komponenten verwenden zu können, müssen eine Reihe von Konfigurationstasks ausgeführt werden, z. B. die Konfiguration der Sicherheit, damit Benutzer sich anmelden können. In diesem Thema werden alle Konfigurationsoptionen beschrieben, die verfügbar sind.

### Prozedur


- „Basiskonfiguration für den mqweb-Server” auf Seite 875
- „Sicherheit konfigurieren” auf Seite 880
- „Konfigurieren des HTTP-Host-Namens” auf Seite 881
- „HTTP-und HTTPS-Ports konfigurieren” auf Seite 882
- „Konfigurieren des Antwortzeitlimits” auf Seite 883
- „Autostart konfigurieren” auf Seite 884
- „Protokollierung konfigurieren” auf Seite 885
- „LTPA-Token konfigurieren” auf Seite 889

- „Verbindungsverhalten des fernen Warteschlangenmanagers für IBM MQ Console konfigurieren” auf Seite 891
- „administrative REST API-Gateway konfigurieren” auf Seite 893
- „messaging REST API konfigurieren” auf Seite 894
- „REST API für MFT konfigurieren” auf Seite 901
- „Die JVM des mqweb-Servers optimieren” auf Seite 906
- „Dateistruktur der Installationskomponente IBM MQ Console und REST API” auf Seite 908

# Basiskonfiguration für den mqweb-Server

Bevor Sie mit der Verwendung von REST API oder IBM MQ Console beginnen können, müssen Sie die richtigen Komponenten installieren und den mqweb-Server konfigurieren, auf dem sich REST API oder IBM MQ Console befindet.

## Informationen zu diesem Vorgang

Bei der Prozedur für diese Task liegt der Schwerpunkt auf einer Basiskonfiguration für den mqweb-Server, so dass ein schneller Einstieg bei der REST API und der IBM MQ Console möglich ist. Die Schritte zum Konfigurieren der Sicherheit beschreiben, wie eine Basisbenutzerregistry konfiguriert wird, aber andere Optionen für die Konfiguration von Benutzern und Rollen vorhanden sind. Weitere Informationen zum Konfigurieren der Sicherheit für den mqweb-Server finden Sie unter IBM MQ Console -und REST API -Sicherheit.

**Anmerkung:** Sie müssen Zugriff auf die Datei `mqwebuser.xml` haben, um diese Prozedur ausführen zu können:

- z/OS Unter z/OS müssen Sie ein Benutzer sein, der Schreibzugriff auf die Datei `mqwebuser.xml` hat.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein, um auf die Datei `mqwebuser.xml` zugreifen zu können.
- V 9.3.5 Linux Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

## Vorgehensweise

1. Installieren Sie die Komponente IBM MQ Console und REST API:
   - AIX Installieren Sie unter AIX die Dateigruppe `mqm.web.rte`. Weitere Informationen zum Installieren von Dateigruppen in AIX finden Sie unter AIX-Installationstasks.
   - IBM i Installieren Sie unter IBM idie WEB-Komponente. Wenn Sie dieses Feature verwenden wollen, müssen Sie auch die Voraussetzungen für 5724L26 IBM MQ Java Messaging and Web Services und 5770JV1 Java SE 8 installieren. Weitere Informationen zum Installieren von Funktionen unter IBM i finden Sie im Abschnitt IBM i-Installationstasks.
   - Linux Installieren Sie unter Linux die Komponente `MQSeriesWeb`. Weitere Informationen zum Installieren von Komponenten unter Linux finden Sie im Abschnitt Linux-Installationstasks.

     V 9.3.5 Ab IBM MQ 9.3.5können Sie den mqweb-Server auch in einer eigenständigen IBM MQ Web Server -Installation unter Linuxausführen. Weitere Informationen zur Installation von IBM MQ Web Serverfinden Sie unter Eigenständige IBM MQ Web Serverinstallieren.
   - Windows Installieren Sie unter Windows das Feature `Web Administration`. Weitere Informationen zum Installieren von Funktionen unter Windows finden Sie im Abschnitt Windows-Installationstasks.

- z/OS Installieren Sie das Feature `IBM MQ for z/OS UNIX System Services Web Components`. Weitere Informationen zur Installation von Komponenten und Funktionen unter z/OS finden Sie im Abschnitt z/OS-Installationstasks.

2. Erstellen Sie den mqweb-Server, der IBM MQ Console und REST APIhostet.

   - z/OS Führen Sie unter z/OSdas Script **crtmqweb** aus.

     Dieses Script erstellt das WebSphere Liberty-Benutzerverzeichnis, das die Konfigurations- und Protokolldateien des mqweb-Servers enthält. Weitere Informationen zum Ausführen des Scripts **crtmqweb** finden Sie unter „Mqweb-Server erstellen" auf Seite 1039.

   - V 9.3.5 Linux Führen Sie bei einer eigenständigen IBM MQ Web Server -Installation die Schritte in „Eigenständigen IBM MQ Web Server konfigurieren" auf Seite 879aus.
   - In allen anderen Umgebungen müssen Sie keine Aktionen ausführen, um den mqweb-Server zu erstellen.

3. z/OS

   Erstellen Sie unter z/OS eine katalogisierte Prozedur zum Starten des mqweb-Servers.

   Weitere Informationen finden Sie unter „Prozedur für den mqweb-Server erstellen" auf Seite 1042.

4. Ersetzen Sie die vorhandene Konfigurationsdatei `mqwebuser.xml` durch die Beispieldatei für die Basisregistry, die für die Basissicherheit konfiguriert ist. Kopieren Sie die Datei `basic_registry.xml` aus dem Verzeichnis *MQ_INSTALLATION_PATH*`/web/mq/samp/configuration` in das entsprechende Verzeichnis für Ihr System und benennen Sie die Datei in `mqwebuser.xml` um:

   - Kopieren Sie in einer IBM MQ -Installation die Datei in das folgende Verzeichnis:
     - Linux AIX Unter AIX and Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Unter Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`

       Dabei steht *MQ_DATA_PATH* für den IBM MQ-Datenpfad; dies ist der Pfad, der bei der Installation von IBM MQ ausgewählt wird. Dieser Pfad ist standardmäßig `C:\ProgramData\IBM\MQ`.
     - z/OS Unter z/OS: *WLP_user_directory*`/servers/mqweb`

       Dabei steht *WLP_user_directory* für das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.
   - V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     Dabei ist *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis, auf das die **MQ_OVERRIDE_DATA_PATH** -Umgebungsvariable verweist.

   Die Beispieldatei `basic_registry.xml` konfiguriert vier Benutzer:

   **mqadmin**
   Ein Benutzer mit Verwaltungsaufgaben, der Mitglied der Rolle `MQWebAdmin` ist

   **mqreader**
   Ein schreibgeschützter Benutzer mit Verwaltungsaufgaben, der Mitglied der Rolle `MQWebAdminRO` ist

   **mftadmin**
   Ein Benutzer mit Verwaltungsaufgaben, der Mitglied der Rolle `MFTWebAdmin` ist

   **mftreader**
   Ein schreibgeschützter Benutzer mit Verwaltungsaufgaben, der Mitglied der Rolle `MFTWebAdmin-RO` ist

   Alle Benutzer sind auch Mitglieder der Rolle `MQWebUser` .

Weitere Informationen zu den verfügbaren Rollen finden Sie unter Rollen auf dem IBM MQ Console und REST API.

5. Optional: Bearbeiten Sie die Datei `mqwebuser.xml`, um weitere Benutzer und Gruppen hinzuzufügen. Weisen Sie diesen Benutzern und Gruppen die entsprechenden Rollen zu, so dass sie berechtigt sind, die REST API bzw. IBM MQ Console zu verwenden. Sie können auch die Kennwörter für die Benutzer ändern, die standardmäßig definiert sind, und die neuen Kennwörter codieren. Weitere Informationen hierzu finden Sie im Abschnitt Benutzer und Rollen konfigurieren .

   **Anmerkung:**

   - z/OS Wenn Sie unter z/OSBenutzer zur Rolle `MQWebUser` hinzufügen, müssen Sie auch der Benutzer-ID der gestarteten mqweb-Task alternativen Benutzerzugriff auf die Benutzer-IDs mit der Rolle `MQWebUser` erteilen. For example:

     ```
     RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
     PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
     ```

   - Multi z/OS Um die Schritte für die ersten Schritte mit messaging REST API abzuschließen, müssen Sie einen Benutzer zur Datei `mqwebuser.xml` hinzufügen. Dieser Benutzer muss den gleichen Namen wie ein vorhandener IBM MQ-Benutzer auf Ihrem System haben. Fügen Sie nach dem gleichen Format wie die anderen Benutzer in der XML-Datei die Benutzer-ID und ein Kennwort nach der folgenden Zeile in der XML-Datei hinzu: `<user name="mftreader" password="mftreader"/>`.

6. Legen Sie Ihre Umgebung so fest, dass sie auf die mqweb-Serverkonfiguration verweist.

   - z/OS Stellen Sie unter z/OS die Umgebungsvariable WLP_USER_DIR so ein, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist, indem Sie den folgenden Befehl eingeben:

     ```
     export WLP_USER_DIR=WLP_user_directory
     ```

     Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an den Befehl **crtmqweb** übergeben wird. For example:

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

     Weitere Informationen finden Sie unter „Mqweb-Server erstellen” auf Seite 1039.

   - V 9.3.5 Linux Setzen Sie in einer eigenständigen IBM MQ Web Server -Installation die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das Datenverzeichnis IBM MQ Web Server .

     Wenn Sie beispielsweise `/var/mqweb` als IBM MQ Web Server -Datenverzeichnis auswählen, geben Sie den folgenden Befehl aus:

     ```
     export MQ_OVERRIDE_DATA_PATH=/var/mqweb
     ```

   - In allen anderen Umgebungen müssen Sie keine Aktionen ausführen, um Ihre Umgebung festzulegen.

7. Standardmäßig sind REST API und IBM MQ Console nur von demselben Host wie der mqweb-Server verfügbar. Aktivieren Sie die Remoteverbindungen zum mqweb-Server, indem Sie den folgenden Befehl eingeben:

   ```
   setmqweb properties -k httpHost -v hostname
   ```

   Dabei gibt *hostname* die IP-Adresse, den Hostnamen des Domänennamensservers (DNS) mit dem Domänennamenssuffix oder den DNS-Hostnamen des Servers an, auf dem IBM MQ installiert ist. Verwenden Sie einen Stern (*) in doppelte Anführungszeichen, um alle verfügbaren Netzschnittstellen anzugeben, wie im folgenden Beispiel dargestellt:

   ```
   setmqweb properties -k httpHost -v "*"
   ```

8. Optional: Standardmäßig ist die administrative REST API für MFT nicht aktiviert. Wenn Sie diese Funktion verwenden möchten, müssen Sie sie aktivieren und einen Koordinationswarteschlangenmanager konfigurieren:
   a) Aktivieren Sie die administrative REST API für MFT, indem Sie den folgenden Befehl eingeben:

      ```
      setmqweb properties -k mqRestMftEnabled -v true
      ```

   b) Konfigurieren Sie, welcher Warteschlangenmanager der Koordinations-WS-Manager ist, indem Sie den folgenden Befehl eingeben:

      ```
      setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
      ```

      Dabei ist *qmgrName* der Name des Koordinations-WS-Managers.
   c) Wenn Sie POST-Aufrufe aktivieren möchten, konfigurieren Sie, welcher Warteschlangenmanager der Befehlswarteschlangenmanager ist, indem Sie den folgenden Befehl eingeben:

      ```
      setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
      ```

      Dabei steht *WS-Managername* für den Namen des Befehlswarteschlangenmanagers.
9. Starten Sie den mqweb-Server, der die REST API und IBM MQ Console unterstützt:
   - ALW Geben Sie unter AIX, Linux, and Windows als privilegierter Benutzer den folgenden Befehl ein:

     ```
     strmqweb
     ```

   - IBM i Geben Sie unter IBM i als privilegierter Benutzer den folgenden Befehl in Qshell ein:

     ```
     /QIBM/ProdData/mqm/bin/strmqweb
     ```

   - z/OS Starten Sie unter z/OS die Prozedur, die Sie in „Prozedur für den mqweb-Server erstellen” auf Seite 1042 erstellt haben.

     Die folgenden Nachrichten werden an die STDOUT DD ausgegeben, um anzuzeigen, dass der mqweb-Server erfolgreich gestartet wurde.

     ```
     [AUDIT   ] MQWB2019I: MQ Console level: 9.2.4 - V924-CD924-L211028
     [AUDIT   ] MQWB0023I: MQ REST API level: 9.2.4 - V924-CD924-L211028
     [AUDIT   ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.763 seconds.
     [AUDIT   ] CWWKZ0001I: Application com.ibm.mq.console started in 2.615 seconds.
     [AUDIT   ] CWWKF0011I: The mqweb server is ready to run a smarter planet. The mqweb
     server started in 10.016 seconds.
     ```

   Sie können den mqweb-Server jederzeit stoppen, indem Sie die gestartete Task des mqweb-Servers unter z/OS stoppen oder indem Sie den Befehl **endmqweb** verwenden. Wenn der mqweb-Server nicht aktiv ist, können Sie die REST API oder IBM MQ Console allerdings nicht verwenden.
10. z/OS
    Optional: Wenn Sie unter z/OSzulassen möchten, dass Systemautomatisierungsprodukte die MQWB2019I -und MQWB0023I -Nachrichten abfangen, die beim Start von IBM MQ Console und REST API ausgegeben werden, konfigurieren Sie den mqweb-Server so, dass diese Nachrichten in die MVS -Konsole geschrieben werden. Um den mqweb-Server für das Schreiben der MQWB2019I- und MQWB0023I-Nachrichten in die MVS-Konsole zu konfigurieren, bearbeiten Sie die Datei `mqweb-buser.xml`, die Sie in Schritt „4” auf Seite 876 erstellt haben, und fügen Sie der Datei die folgende Zeile hinzu:

    ```
    <zosLogging enableLogToMVS="true" wtoMessage="MQWB2019I,MQWB0023I"/>
    ```

    Weitere Informationen zur z/OS-Protokollierung im Mqweb-Server finden Sie unter z/OS-Protokollierung (zosLogging).

### Nächste Schritte

1. Konfigurieren Sie die Einstellungen des mqweb-Servers, einschließlich der Aktivierung von HTTP-Verbindungen, und ändern Sie die Portnummer. Weitere Informationen finden Sie unter „IBM MQ Console und REST API konfigurieren” auf Seite 874.
2. Optional können Sie die REST API konfigurieren:
   a. Konfigurieren Sie die Cross-Origin-Ressourcenfreigabe für die REST API. Standardmäßig können Sie auf die REST API nicht über Webressourcen zugreifen, die nicht in derselben Domäne gehostet werden wie die REST API. Dies bedeutet, dass Kreuzursprungsanforderungen nicht aktiviert sind. Sie können Cross Origin Resource Sharing (CORS) konfigurieren, um Cross-Origin-Anforderungen von angegebenen URLs zu ermöglichen. Weitere Informationen finden Sie im Abschnitt CORS für den REST API konfigurieren.
   b. Konfigurieren Sie die REST API für MFT. Weitere Informationen finden Sie unter „REST API für MFT konfigurieren” auf Seite 901.
3. Verwenden Sie die REST API oder IBM MQ Console:
   - Erste Schritte mit der administrative REST API
   - Erste Schritte mit der messaging REST API
   - Erste Schritte mit der IBM MQ Console

## V 9.3.5 Linux Eigenständigen IBM MQ Web Server konfigurieren

Ab IBM MQ 9.3.5können Sie den mqweb-Server, der die IBM MQ Console und REST API hostet, in einer eigenständigen IBM MQ Web Server -Installation ausführen.

### Vorbereitende Schritte

Die eigenständige Version von IBM MQ Web Server ist ausschließlich unter Linuxverfügbar.

Bevor Sie den mqweb-Server konfigurieren können, müssen Sie IBM MQ Web Server installieren, indem Sie die Schritte im Abschnitt Eigenständige IBM MQ Web Serverinstallierenausführen.

### Informationen zu diesem Vorgang

Befolgen Sie die Prozedur in dieser Task, um einen neuen mqweb-Server zu erstellen und zu konfigurieren, der in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird. Sie können mehrere mqweb-Server für die Ausführung in einer eigenständigen IBM MQ Web Server -Installation konfigurieren, indem Sie diese Prozedur wiederholen.

### Vorgehensweise

1. Erstellen Sie das IBM MQ Web Server -Datenverzeichnis.

   Das Datenverzeichnis wird zum Speichern der Konfigurations-und Protokolldateien für den mqweb-Server verwendet, auf dem IBM MQ Console und REST APIausgeführt werden. Sie können jedes Verzeichnis verwenden, das Sie als IBM MQ Web Server -Datenverzeichnis auswählen.

   Der Benutzer-ID, die Sie zum Starten des mqweb-Servers verwenden, muss Lese-und Schreibzugriff auf das Datenverzeichnis erteilt werden.
2. Setzen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das Datenverzeichnis, das Sie in Schritt „1” auf Seite 879erstellt haben.
   Wenn Sie beispielsweise `/var/mqweb` als IBM MQ Web Server -Datenverzeichnis auswählen, geben Sie den folgenden Befehl aus:

   ```
   export MQ_OVERRIDE_DATA_PATH=/var/mqweb
   ```
3. Verwenden Sie den Befehl **setmqenv** , um die IBM MQ -Umgebung einzurichten.

Wechseln Sie in das Verzeichnis `bin` des IBM MQ Web Server -Installationsverzeichnisses und setzen Sie anschließend den folgenden Befehl ab:

```
. setmqenv -s
```

4. Erstellen Sie mit dem Befehl **crtmqdir** die IBM MQ -Verzeichnisse und -Dateien im Datenverzeichnis. Die erstellten Dateien enthalten eine Vorlagendefinition für den mqweb-Server.

   Geben Sie den folgenden Befehl ein:

   ```
   crtmqdir -s -f
   ```

5. Optional: Wenn dieser mqweb-Server der erste ist, den Sie für die Ausführung mit dieser Installation des eigenständigen IBM MQ Web Servers erstellt haben, verwenden Sie den Befehl **mqlicense** , um die IBM MQ -Lizenz zu überprüfen und zu akzeptieren.

   Sie müssen diesen Befehl als Benutzer mit Schreibzugriff auf das IBM MQ Web Server -Installationsverzeichnis ausführen.

   Geben Sie beispielsweise den folgenden Befehl aus, um die IBM MQ -Lizenz anzuzeigen:

   ```
   mqlicense
   ```

   Weitere Informationen finden Sie unter mqlicense.

6. Optional: Führen Sie die folgenden Schritte aus, um einen vorhandenen mqweb-Server für die Ausführung in der neu konfigurierten eigenständigen IBM MQ Web Server -Installation zu migrieren:

   a. Sichern Sie Ihre vorhandene mqweb-Serverkonfiguration.

   b. Stellen Sie die Dateien im Verzeichnis *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST` wieder her, wobei *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis ist, das Sie in Schritt „1” auf Seite 879erstellt haben.

   Weitere Informationen finden Sie unter „Mqweb-Serverkonfiguration sichern und wiederherstellen” auf Seite 911.

   **Anmerkung:** Einige Funktionen von IBM MQ Console und REST API sind in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar. Wenn Sie einen mqweb-Server von einer IBM MQ -Installation auf eine eigenständige IBM MQ Web Server -Installation migrieren, können diese Funktionen nach der Migration nicht verwendet werden. Weitere Informationen zu den Einschränkungen, die in einer eigenständigen IBM MQ Web Server -Installation gelten, finden Sie unter IBM MQ Console und REST API.

### Nächste Schritte

Konfigurieren Sie den mqweb-Server, indem Sie die im Abschnitt „Basiskonfiguration für den mqweb-Server” auf Seite 875beschriebenen Schritte ausführen.

## Sicherheit konfigurieren

Sie können die Sicherheit für die IBM MQ Console und die REST API konfigurieren, indem Sie die `mqwebuser.xml`-Datei bearbeiten. Zur Konfiguration und Authentifizierung von Benutzern können Sie entweder eine Basisbenutzerregistry oder eine LDAP-Registry oder einen der anderen mit WebSphere Liberty bereitgestellten Registry-Typen konfigurieren. Anschließend können Sie diese Benutzer berechtigen, indem Sie Benutzer und Gruppen einer Rolle zuordnen.

### Informationen zu diesem Vorgang

Um die Sicherheit für die IBM MQ Console und die REST API zu konfigurieren, müssen Sie Benutzer und Gruppen konfigurieren. Diese Benutzer und Gruppen können dann berechtigt werden, die IBM MQ Console und/oder die REST API zu verwenden. Weitere Informationen zur Konfiguration von Benutzern und Gruppen und zur Authentifizierung und Berechtigung von Benutzern finden Sie unter IBM MQ Console und REST API -Sicherheit.

Wenn sich Benutzer bei der IBM MQ Console authentifizieren, wird ein LTPA-Token generiert. Mit diesem Token kann der Benutzer die IBM MQ Console bis zum Ablauf des Tokens ohne erneute Authentifizierung verwenden.

Wenn Sie die tokenbasierte Authentifizierung mit dem REST APIverwenden, wird ein anderes LTPA-Token generiert, wenn sich der Benutzer über die `/login` REST API -Ressource mit der Methode HTTP POST anmeldet. Sie können konfigurieren, wann dieses Token abläuft, und ob dieses Token sowohl für HTTP-als auch für HTTPS-Verbindungen verwendet werden kann. Weitere Informationen finden Sie unter „LTPA-Token konfigurieren” auf Seite 889.

### Prozedur

- IBM MQ Console -und REST API -Sicherheit
- „LTPA-Token konfigurieren” auf Seite 889

## Konfigurieren des HTTP-Host-Namens

Standardmäßig ist der mqweb-Server, der die IBM MQ Console und REST API hostet, so konfiguriert, dass nur lokale Verbindungen erlaubt sind. Das heißt, dass auf die IBM MQ Console und REST API nur auf dem System zugegriffen werden kann, auf dem IBM MQ Console und REST API installiert sind. Sie können den Hostnamen so konfigurieren, dass ferne Verbindungen mit dem Befehl **setmqweb** zugelassen werden.

### Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration des HTTP-Hostnamens an, indem Sie den folgenden Befehl verwenden:

  ```
  dspmqweb properties -a
  ```

  Im Feld `httpHost` wird der HTTP-Hostname angezeigt.
- Legen Sie den HTTP-Hostnamen fest, indem Sie den folgenden Befehl verwenden:

  ```
  setmqweb properties -k httpHost -v hostName
  ```

  Dabei gibt *hostname* die IP-Adresse, den Hostnamen des Domänennamensservers (DNS) mit dem Domänennamenssuffix oder den DNS-Hostnamen des Servers an, auf dem IBM MQ installiert ist. Verwenden Sie einen Stern in doppelten Anführungszeichen, um alle verfügbaren Netzschnittstellen anzugeben. Verwenden Sie den Wert `localhost` , um nur lokale Verbindungen zuzulassen.
- Geben Sie den folgenden Befehl ein, um den HTTP-Hostnamen zu dekonfigurieren:

  ```
  setmqweb properties -k httpHost -d
  ```

## HTTP-und HTTPS-Ports konfigurieren

Standardmäßig verwendet der mqweb-Server, auf dem die IBM MQ Console und REST API ausgeführt werden, den HTTPS-Port 9443. Der Port, der den HTTP-Verbindungen zugeordnet ist, ist inaktiviert. Sie können den HTTP-Port aktivieren, einen anderen HTTPS-Port konfigurieren oder den HTTP-oder HTTPS-Port inaktivieren. Sie können die Ports mit dem Befehl **setmqweb** konfigurieren.

### Vorbereitende Schritte

Wenn Sie den HTTP-Port aktivieren und die tokenbasierte Authentifizierung verwenden, müssen Sie dasselbe LTPA-Token aktivieren, das sowohl für HTTP-als auch für HTTPS-Verbindungen verwendet werden kann. Weitere Informationen finden Sie unter „LTPA-Token konfigurieren” auf Seite 889.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

**Achtung:** Standardmäßig erfordert der mqweb-Server, dass LTPA-Token für alle Anforderungen gesichert werden. Wenn der mqweb-Server so konfiguriert ist, dass LTPA-Token geschützt werden müssen, können Sie die folgenden Aktionen nicht ausführen, wenn Sie eine Verbindung zum HTTP -Port herstellen:

- Melden Sie sich am IBM MQ Console an.
- Verwenden Sie die tokenbasierte Authentifizierung mit REST API.

Damit LTPA-Tokens von HTTP -Anforderungen verwendet werden können, setzen Sie die Eigenschaft **secureLTPA** auf `false`. Weitere Informationen finden Sie unter „LTPA-Token konfigurieren” auf Seite 889.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration der HTTP-und HTTPS-Ports mit dem folgenden Befehl an:

  ```
  dspmqweb properties -a
  ```

  Im Feld `httpPort` wird der HTTP-Port angezeigt, und im Feld `httpsPort` wird der HTTPS-Port angezeigt.
- Aktivieren oder konfigurieren Sie den HTTP-Port. Verwenden Sie dazu den folgenden Befehl:
  - Aktivieren oder setzen Sie den HTTP-Port mit dem folgenden Befehl:

    ```
    setmqweb properties -k httpPort -v portNumber
    ```

    Dabei gibt *portNumber* den Port an, den Sie für HTTP-Verbindungen verwenden möchten. Sie können den Port mit einem Wert von `-1` inaktivieren.
  - Setzen Sie den HTTP-Portwert auf den Standardwert von `-1` zurück, indem Sie den folgenden Befehl verwenden:

    ```
    setmqweb properties -k httpPort -d
    ```
- Konfigurieren Sie den HTTPS-Port:
  - Legen Sie die HTTPS-Portnummer fest, indem Sie den folgenden Befehl verwenden:

    ```
    setmqweb properties -k httpsPort -v portNumber
    ```

    Dabei gibt *portNumber* den Port an, den Sie für HTTPS-Verbindungen verwenden möchten. Sie können den Port mit einem Wert von `-1` inaktivieren.
  - Setzen Sie die HTTPS-Portnummer auf den Standardwert von 9443 zurück, indem Sie den folgenden Befehl verwenden:

    ```
    setmqweb properties -k httpsPort -d
    ```

## Konfigurieren des Antwortzeitlimits

Das IBM MQ Console- und das REST API-Zeitlimit werden standardmäßig überschritten, wenn die Zeit, die zum Senden einer Antwort an einen Client benötigt wird, länger als 30 Sekunden ist. Die IBM MQ Console und die REST API können Sie mit dem Befehl **setmqweb** so konfigurieren, dass sie einen anderen Zeitlimitwert verwenden.

### Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration des Anforderungszeitlimits mit dem folgenden Befehl an:

  ```
  dspmqweb properties -a
  ```

  Im Feld `mqRestRequestTimeout` wird der aktuelle Wert für das Antwortzeitlimit angezeigt. Weitere Informationen finden Sie unter dspmqweb properties.
- Legen Sie das Anforderungszeitlimit fest, indem Sie den folgenden Befehl verwenden:

  ```
  setmqweb properties -k mqRestRequestTimeout -v timeout
  ```

  Dabei gibt *timeout* die Zeit (in Sekunden) vor dem Zeitlimit an.
- Setzen Sie das Anforderungszeitlimit mit dem folgenden Befehl auf den Standardwert von 30 Sekunden zurück:

  ```
  setmqweb properties -k mqRestRequestTimeout -d
  ```

## Autostart konfigurieren

Standardmäßig wird die IBM MQ Console beim Start des mqweb-Servers automatisch gestartet. Mit dem Befehl **setmqweb** können Sie konfigurieren, ob IBM MQ Console und REST API automatisch gestartet werden sollen.

### Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration des automatischen Startstarts mit dem folgenden Befehl an:

  ```
  dspmqweb properties -a
  ```

  Im Feld `mqRestAutostart` wird angezeigt, ob die REST API automatisch gestartet wird, und im Feld `mqConsoleAutostart`, ob die IBM MQ Console automatisch gestartet wird.
- Konfigurieren Sie, ob die IBM MQ Console automatisch gestartet wird, indem Sie den folgenden Befehl verwenden:

  ```
  setmqweb properties -k mqConsoleAutostart -v start
  ```

  Dabei nimmt *start* den Wert `true` an, wenn die IBM MQ Console automatisch gestartet werden soll, andernfalls lautet der Wert `false`.
- Konfigurieren Sie, ob die REST API automatisch gestartet wird, indem Sie den folgenden Befehl verwenden:

  ```
  setmqweb properties -k mqRestAutostart -v start
  ```

  Dabei nimmt *start* den Wert `true` an, wenn die REST API automatisch gestartet werden soll, andernfalls lautet der Wert `false`.

## Protokollierung konfigurieren

Sie können die Protokollierungsstufen, die maximale Protokolldateigröße und die maximale Anzahl der Protokolldateien konfigurieren, die von dem mqweb-Server verwendet werden, der die IBM MQ Console und REST APIhostet. Sie können die Protokollierung mit dem Befehl **setmqweb** konfigurieren.

## Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

## Informationen zu diesem Vorgang

Der mqweb-Server schreibt Protokollnachrichten und Traces in die folgenden Protokolldateien:

**`console.log` und `messages.log`**
Diese Dateien enthalten Nachrichten, die von IBM MQ Console, REST APIund dem mqweb-Server ausgegeben werden, auf dem diese Komponenten ausgeführt werden.

**`trace.log`**
Diese Datei enthält den Trace für IBM MQ Console und REST API. Der Trace wird nur in diese Datei geschrieben, wenn der Trace aktiviert ist.

Die Protokolldateien für den mqweb-Server befinden sich in einem der folgenden Verzeichnisse:

- In einer IBM MQ -Installation:
  - Linux AIX Unter AIX oder Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`
  - Windows Unter Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb\logs`, wobei *MQ_DATA_PATH* der IBM MQ -Datenpfad ist. Dieser Pfad ist der Datenpfad, der während der Installation von IBM MQausgewählt wird. Standardmäßig lautet dieser Pfad `C:\ProgramData\IBM\MQ`.
  - z/OS Unter z/OS: *WLP_user_directory*`/servers/mqweb/logs`

Dabei ist *WLP_Benutzerverzeichnis* das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

- V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb/logs

  Dabei ist *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis, auf das die **MQ_OVERRIDE_DATA_PATH** -Umgebungsvariable verweist.

Die Messaging-Tracedateien für den Messaging- REST API -Code, der im mqweb-Server ausgeführt wird, befinden sich in einem der folgenden Verzeichnisse:

- In einer IBM MQ -Installation:
  - Linux AIX Unter AIX oder Linux: /var/mqm/web/installations/*installationName*/servers/mqweb
  - Windows Unter Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb, wobei *MQ_DATA_PATH* der IBM MQ -Datenpfad ist. Dieser Pfad ist der Datenpfad, der während der Installation von IBM MQausgewählt wird. Standardmäßig lautet dieser Pfad C:\ProgramData\IBM\MQ.
  - z/OS Unter z/OS: *WLP_user_directory*/servers/mqweb

    Dabei ist *WLP_Benutzerverzeichnis* das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.
- V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

  Dabei ist *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis, auf das die **MQ_OVERRIDE_DATA_PATH** -Umgebungsvariable verweist.

Weitere Informationen zum Aktivieren der Tracefunktion finden Sie hier:

- Für die REST API lesen Sie Tracing the REST API
- Für die IBM MQ Console lesen Sie Tracing the IBM MQ Console

## Prozedur

- Zeigen Sie die aktuelle Konfiguration der REST API-Protokollierung mit Hilfe des folgenden Befehls an:

  ```
  dspmqweb properties -a
  ```

  - Im Feld maxTraceFileSize wird die maximale Protokolldateigröße angezeigt.
  - Im Feld maxTraceFiles wird die maximale Anzahl Protokolldateien angezeigt.
  - Im Feld traceSpec wird die verwendete Tracestufe angezeigt
  - Im Feld maxMsgTraceFileSize wird die maximale Größe der Messaging-Tracedatei angezeigt.
  - Im Feld maxMsgTraceFiles wird die maximale Anzahl der Messaging-Tracedateien angezeigt.
- Konfigurieren Sie die maximale Größe der Dateien messages.log und trace.log :
  - Legen Sie die maximale Größe der Protokolldatei fest, indem Sie den folgenden Befehl verwenden:

    ```
    setmqweb properties -k maxTraceFileSize -v size
    ```

    Dabei gibt *size* die Größe (in MB) an, die jede Protokolldatei erreichen kann.
  - Setzen Sie die maximale Protokolldateigröße auf den Standardwert von 20 MB zurück, indem Sie den folgenden Befehl verwenden:

    ```
    setmqweb properties -k maxTraceFileSize -d
    ```
- Konfigurieren Sie die maximale Anzahl der Dateien messages.log und trace.log :
  - Legen Sie die maximale Anzahl der Protokolldateien mit dem folgenden Befehl fest:

```
setmqweb properties -k maxTraceFiles -v max
```

Dabei gibt *max* die maximale Anzahl Dateien an.

- Setzen Sie die maximale Anzahl jeder Protokolldatei mit dem folgenden Befehl auf den Standardwert 2 zurück:

```
setmqweb properties -k maxTraceFiles -d
```

- Konfigurieren Sie die maximale Größe der Messaging-Tracedatei:
  - Legen Sie die maximale Größe der Messaging-Tracedatei mit dem folgenden Befehl fest:

```
setmqweb properties -k maxMsgTraceFileSize -v size
```

Dabei gibt *size* die Größe in MB an, die jede Messaging-Tracedatei erreichen kann.

  - Setzen Sie die maximale Größe der Messaging-Tracedatei mit dem folgenden Befehl auf den Standardwert von 200 MB zurück:

```
setmqweb properties -k maxMsgTraceFileSize -d
```

- Konfigurieren Sie die maximale Anzahl der zu verwendenden Messaging-Tracedateien:
  - Legen Sie die maximale Anzahl von Dateien fest, die für den Messaging-Trace verwendet werden sollen, indem Sie den folgenden Befehl verwenden:

```
setmqweb properties -k maxMsgTraceFiles -v max
```

Dabei gibt *max* die maximale Anzahl Dateien an.

  - Setzen Sie die maximale Anzahl der Dateien, die für den Messaging-Trace verwendet werden sollen, auf den Standardwert 5 zurück, indem Sie den folgenden Befehl verwenden:

```
setmqweb properties -k maxMsgTraceFiles -d
```

- Konfigurieren Sie die Tracestufe, die der mqweb-Server schreibt:
  - Legen Sie die verwendete Tracespezifikation mit folgendem Befehl fest:

```
setmqweb properties -k traceSpec -v level
```

Dabei steht *Ebene* für einen der Werte, die in Tabelle 51 auf Seite 888aufgelistet sind. In der Tabelle werden die Protokollebenen nach zunehmender Detaillierungsebene sortiert dargestellt. Wenn Sie eine Protokollierungsstufe aktivieren, aktivieren Sie auch die einzelnen Ebenen vor ihr. Wenn Sie beispielsweise die Protokollierungsstufe ***=warning** aktivieren, aktivieren Sie auch die Protokollebenen ***=severe** und ***=fatal** .

Ändern Sie diesen Wert, wenn Sie vom IBM Support dazu aufgefordert werden.

  - Setzen Sie die verwendete Tracespezifikation mit dem folgenden Befehl auf den Standardwert ***=info** zurück:

```
setmqweb properties -k traceSpec -d
```

*Tabelle 51. Gültige Protokollierungsstufen*

| **Wert** | **Protokollierungsstufe angewendet** |
|---|---|
| * =Aus | Die Protokollierung ist inaktiviert. |
| * =fatal | Die Task kann nicht fortgesetzt werden, und die Komponente, die Anwendung und der Server können nicht funktionieren. |
| * =schwer | Die Task kann nicht fortgesetzt werden, aber die Komponente, die Anwendung und der Server können weiterhin funktionieren. Diese Stufe kann auch einen bevorstehenden nicht behebbaren Fehler anzeigen. |

| *Tabelle 51. Gültige Protokollierungsstufen (Forts.)* | |
|---|---|
| **Wert** | **Protokollierungsstufe angewendet** |
| * =Warnung | Potenzieller Fehler oder drohenden Fehler. Diese Stufe kann auch auf einen progressiven Fehler hinweisen (z. B. die potenzielle Ressourcenlecks). |
| * =audit | Signifikanes Ereignis, das den Serverstatus oder die Server |
| * =Info | Allgemeine Informationen zur Gesamtaufgaben-fortschritt |
| * =Konfiguration | Konfigurationsänderung oder -status |
| * =detail | Allgemeine Informationen zum Fortschritt der Subtask |
| * =fein | Traceinformationen-Allgemeine Trace-und Methodeneintrags-, Exit-und Rückgabewerte |
| * =finer | Traceinformationen-Ausführlicher Trace |
| * =finest | Trace-Informationen-Ein ausführlicher Trace, der alle Details enthält, die für das Debugging von Problemen erforderlich sind. |
| * =alle | Alle Ereignisse werden protokolliert |

# LTPA-Token konfigurieren

LTPA-Token können verwendet werden, um zu vermeiden, dass ein Benutzer Berechtigungsnachweise für Benutzername und Kennwort für jede Anforderung an den mqweb-Server zur Verfügung stellt. Mit dem Befehl **setmqweb** können Sie den Namen des LTPA-Token-Cookies und das Ablaufintervall für LTPA-Authentifizierungstoken konfigurieren und konfigurieren, ob LTPA-Token von HTTP -Verbindungen verwendet werden können.

## Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Anmerkung:** Wenn Sie sowohl die IBM MQ Console als auch die Tokenauthentifizierung mit der REST API verwenden, wird das Ablaufintervall gemeinsam genutzt.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

## Informationen zu diesem Vorgang

Wenn sich Benutzer bei der IBM MQ Console anmelden, wird ein LTPA-Token generiert. Wenn Sie die tokenbasierte Authentifizierung mit der REST APIverwenden, wird ein LTPA-Token generiert, wenn sich der Benutzer mit der `/login` REST API -Ressource mit der Methode HTTP POST anmeldet. Dieses Token wird in einem Cookie zurückgegeben. Das Token wird verwendet, um den Benutzer zu authentifizieren, ohne dass der Benutzer mit seiner Benutzer-ID und dem Kennwort erneut angemeldet werden muss, bis das Token verfällt. Das Standardablaufintervall ist 120 Minuten.

Der Name des Cookies, das das LTPA-Token enthält, variiert je nach Plattform:

- MQ Appliance Auf IBM MQ Appliance ist das LTPA-Token `LtpaToken2`. Dieser Wert kann nicht geändert werden.
- z/OS ALW Standardmäßig wird auf allen anderen Plattformen der Name des Cookies, das das LTPA-Token enthält, mit `LtpaToken2` gestartet und enthält ein Suffix, das sich ändern kann, wenn der mqweb-Server erneut gestartet wird. Dieser randomisierte Cookiename ermöglicht es, dass mehr als ein mqweb-Server auf demselben System ausgeführt wird. Wenn der Cookiename jedoch ein konsistenter Wert bleiben soll, können Sie den Namen des Cookies mit dem Befehl **setmqweb** angeben.

z/OS IBM i ALW Wenn Sie sowohl die HTTP-als auch die HTTPS-Ports aktivieren, kann ein LTPA-Token, das für eine HTTPS-Anforderung ausgegeben wird, für eine HTTP-Anforderung wiederverwendet werden. Dieses Verhalten ist standardmäßig inaktiviert, aber Sie können dieses Verhalten mit dem Befehl **setmqweb** aktivieren.

## Prozedur

- Zeigen Sie den aktuellen Ablauf des LTPA-Tokens, den Namen des LTPA-Token-Cookies und die Frage an, ob das LTPA-Token für HTTP-Anforderungen mit dem folgenden Befehl verwendet werden kann:

  ```
  dspmqweb properties -a
  ```

  - Im Feld `ltpaCookieName` wird der Name des LTPA-Token-Cookies angezeigt. Wenn Sie noch keinen Cookienamen festgelegt haben, lautet der Wert dieser Eigenschaft `LtpaToken2_${env.MQWEB_LTPA_SUFFIX}` auf AIX, Linux, and Windows oder `LtpaToken2_${httpsPort}` auf z/OS, . Die Variable nach dem Präfix `LtpaToken2_` wird vom mqweb-Server verwendet, um einen eindeutigen Namen für das Cookie zu generieren. Sie können diese Variable nicht festlegen, aber Sie können die `ltpaCookieName` in einen Wert Ihrer Wahl ändern.
  - Im Feld `ltpaExpiration` wird die Ablaufzeit des LTPA-Tokens angezeigt.
  - Das Feld `secureLtpa` wird auf `false` gesetzt, wenn LTPA-Token von HTTP-Anforderungen verwendet werden können.
- Konfigurieren Sie den Ablauf des LTPA-Tokens:
  - Definieren Sie den Ablauf des LTPA-Tokens, indem Sie den folgenden Befehl eingeben:

    ```
    setmqweb properties -k ltpaExpiration -v time
    ```

Dabei gibt *time* die Zeit in Minuten an, bevor das LTPA-Token abläuft und der Benutzer abgemeldet ist.

- Setzen Sie den Standardwert für das LTPA-Token auf den Standardwert von 120 Minuten zurück, indem Sie den folgenden Befehl eingeben:

  ```
  setmqweb properties -k ltpaExpiration -d
  ```

- z/OS ALW

  Konfigurieren Sie den Cookienamen für LTPA-Token:

  - Legen Sie den Namen des LTPA-Token-Cookies fest, indem Sie den folgenden Befehl eingeben:

    ```
    setmqweb properties -k ltpaCookieName -v name
    ```

    Dabei gibt *name* einen eindeutigen Namen für das LTPA-Token-Cookie an.
  - Setzen Sie den Namen des LTPA-Token-Cookies auf den Standardwert zurück, wobei dem Präfix `LtpaToken2_` zufällige Zeichen folgen, indem Sie den folgenden Befehl eingeben:

    ```
    setmqweb properties -k ltpaCookieName -d
    ```

- z/OS ALW

  Konfigurieren Sie, ob das LTPA-Token von HTTP-Verbindungen verwendet werden kann, indem Sie den folgenden Befehl eingeben:

  ```
  setmqweb properties -k secureLtpa -v secure
  ```

  Dabei gibt *secure* an, ob das LTPA-Token sowohl von insicheren HTTP-Verbindungen als auch von sicheren HTTPS-Verbindungen verwendet werden kann. Der Wert `false` ermöglicht sowohl HTTP-als auch HTTPS-Verbindungen für die Verwendung desselben LTPA-Tokens.

## V 9.3.0 Verbindungsverhalten des fernen Warteschlangenmanagers für IBM MQ Console konfigurieren

Wenn Sie die IBM MQ Console verwenden, können Sie Verbindungen zu fernen Warteschlangenmanagern erstellen. Das heißt, Sie können eine Verbindung zu Warteschlangenmanagern herstellen, die nicht zu derselben Installation gehören wie der mqweb-Server, der IBM MQ Console ausführt. Es gibt eine Reihe von Konfigurationsoptionen, die Sie festlegen können, um das Verhalten der fernen Warteschlangenmanagerverbindungen zu steuern.

### Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

**Achtung:** Linux V 9.3.5

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** in einer eigenständigen IBM MQ Web Server -Installation absetzen, müssen Sie die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das IBM MQ Web Server -Datenverzeichnis setzen.

## Informationen zu diesem Vorgang

Sie können die folgenden Konfigurationsoptionen einstellen:

- Ob ferne Warteschlangenmanagerverbindungen zulässig sind.
- Ob die Verbindungen über IBM MQ Console oder nur über die Befehlszeile hinzugefügt werden können.
- Ob lokale Warteschlangenmanager in IBM MQ Console angezeigt werden, wenn ferne Warteschlangenmanagerverbindungen zulässig sind.
- Gibt an, ob ferne Warteschlangenmanagerverbindungen automatisch hergestellt werden, wenn IBM MQ Console gestartet wird oder wenn ein Verbindungsfehler auftritt.
- Die Zeitdauer zwischen jeder Aktualisierung der Liste der fernen Warteschlangenmanager, die in IBM MQ Console angezeigt wird.

## Prozedur

- Geben Sie den folgenden Befehl ein, um die aktuellen Einstellungen für die Verbindungskonfiguration des fernen Warteschlangenmanagers anzuzeigen:

  ```
  dspmqweb properties -a
  ```

  - Das Feld `mqConsoleRemoteSupportEnabled` gibt an, ob Verbindungen zu fernen Warteschlangenmanagern zulässig sind.
  - Das Feld `mqConsoleRemoteUIAdmin` gibt an, ob ferne Warteschlangenmanagerverbindungen über die IBM MQ Console hinzugefügt werden können.
  - Das Feld `mqConsoleRemoteAllowLocal` gibt an, ob die lokalen Warteschlangenmanager angezeigt werden sollen.
  - Das Feld `mqConsoleRemotePollTime` gibt an, wie viele Sekunden zwischen jeder Aktualisierung der Liste der fernen Warteschlangenmanager liegen.
- Geben Sie den folgenden Befehl ein, um ferne Warteschlangenmanager-Verbindungen mit IBM MQ Console zu verhindern oder zuzulassen:

  ```
  setmqweb properties -k mqConsoleRemoteSupportEnabled -v true or false
  ```

  Dabei lässt *true* ferne Warteschlangenmanagerverbindungen zu oder `false` verhindert ferne Warteschlangenmanagerverbindungen.

  **Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist die Eigenschaft **mqConsoleRemoteSupportEnabled** ungültig. Der eigenständige IBM MQ Web Server unterstützt nur Verbindungen zu fernen Warteschlangenmanagern.
- Geben Sie den folgenden Befehl ein, um zu verhindern oder zuzulassen, dass ferne Warteschlangenmanagerverbindungen über IBM MQ Console oder nur über die Befehlszeile hinzugefügt werden:

  ```
  setmqweb properties -k mqConsoleRemoteUIAdmin -v true or false
  ```

Dabei ermöglicht `true` das Hinzufügen von Verbindungen zu fernen Warteschlangenmanagern über die IBM MQ Console und die Befehlszeile oder `false` das Hinzufügen von Verbindungen zu fernen Warteschlangenmanagern nur über den Befehl **setmqweb remote** in der Befehlszeile.

- Geben Sie den folgenden Befehl ein, um die Anzeige von lokalen Warteschlangenmanagern in IBM MQ Console zu verhindern oder zuzulassen, wenn ferne Warteschlangenmanagerverbindungen zulässig sind:

  ```
  setmqweb properties -k mqConsoleRemoteAllowLocal -v true or false
  ```

  Dabei lässt `true` die Anzeige lokaler Warteschlangenmanager zu oder `false` verdeckt die lokalen Warteschlangenmanager.

  **Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist die Eigenschaft **mqConsoleRemoteAllowLocal** ungültig. Der eigenständige IBM MQ Web Server unterstützt nur Verbindungen zu fernen Warteschlangenmanagern.

- Geben Sie den folgenden Befehl ein, um die Zeitspanne zwischen den einzelnen Aktualisierungen der in IBM MQ Console angezeigten Liste der fernen Warteschlangenmanager festzulegen:

  ```
  setmqweb properties -k mqConsoleRemotePollTime -v seconds
  ```

  Dabei wird *seconds* auf einen ganzzahligen Wert gesetzt, der die Anzahl der Sekunden zwischen jeder Aktualisierung der Liste der fernen Warteschlangenmanager angibt.

**Zugehörige Verweise**

setmqweb

dspmqweb

# administrative REST API-Gateway konfigurieren

Wenn das administrative REST API-Gateway aktiviert ist, können Sie die Fernverwaltung mit der REST API ausführen, indem Sie einen Gateway-Warteschlangenmanager verwenden. Sie können den Warteschlangenmanager konfigurieren, der als Standard-Gateway-Warteschlangenmanager verwendet wird, oder Sie können die Fernverwaltung verhindern, indem Sie das administrative REST API-Gateway mit dem Befehl **setmqweb** inaktivieren.

## Vorbereitende Schritte

**Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist diese Aufgabe nicht anwendbar. administrative REST API ist in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

### Informationen zu diesem Vorgang

Wenn der mqweb-Server in einer IBM MQ -Installation ausgeführt wird, ist das Gateway administrative REST API standardmäßig aktiviert.

Der Standard-Gateway-WS-Manager wird verwendet, wenn die beiden folgenden Anweisungen wahr sind:

- Es wurde kein Warteschlangenmanager im Header `ibm-mq-rest-gateway-qmgr` einer REST-Anforderung angegeben.
- Der in der Ressourcen-URL der REST API angegebene Warteschlangenmanager ist kein lokaler Warteschlangenmanager.

Weitere Informationen zur Fernverwaltung mit der REST APIfinden Sie im Abschnitt Fernverwaltung mit der REST API.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration des administrative REST API-Gateways an, indem Sie den folgenden Befehl verwenden:

  ```
  dspmqweb properties -a
  ```

  Das Feld `mqRestGatewayEnabled` zeigt an, ob das Gateway aktiviert ist, und das Feld `mqRestGatewayQmgr` zeigt den Namen des Standard-Gateway-Warteschlangenmanagers an.
- Konfigurieren Sie mit folgendem Befehl, ob das administrative REST API-Gateway aktiviert wird:

  ```
  setmqweb properties -k mqRestGatewayEnabled -v enabled
  ```

  Hierbei muss *enabled* auf den Wert **`true`** gesetzt werden, um das administrative REST API-Gateway zu aktivieren, andernfalls auf **`false`**.
- Konfigurieren Sie, welcher WS-Manager als Standard-Gateway-Warteschlangenmanager verwendet wird:
  - Definieren Sie den Standard-Gateway-WS-Manager mit dem folgenden Befehl:

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

    Dabei steht *qmgrName* für den Namen eines Warteschlangenmanagers in derselben Installation wie der mqweb-Server.
  - Sie können den Standard-Gateway-WS-Manager mit dem folgenden Befehl zurücknehmen:

    ```
    setmqweb properties -k mqRestGatewayQmgr -d
    ```

## messaging REST API konfigurieren

Sie können messaging REST API auf verschiedene Arten konfigurieren. Sie können das Feature messaging REST API aktivieren oder inaktivieren. Sie können die maximale Anzahl gepoolter Verbindungen, die von messaging REST APIverwendet werden können, und das Verhalten von messaging REST API auswählen, wenn alle Verbindungen verwendet werden. Sie können auch auswählen, welcher Benutzerkontext für die Berechtigung verwendet wird, wenn Sie die messaging REST API zum Senden, Empfangen, Durchsuchen oder Veröffentlichen einer Nachricht verwenden.

### Prozedur

- „messaging REST API aktivieren” auf Seite 895

- „Verbindungspooling für messaging REST API konfigurieren” auf Seite 896
- V 9.3.2 „Benutzerkontext konfigurieren, der für die Berechtigung in messaging REST API verwendet wird” auf Seite 899

## messaging REST API aktivieren

Mit dem Befehl **setmqweb** können Sie konfigurieren, ob die messaging REST API aktiviert ist. Standardmäßig ist messaging REST API aktiviert.

### Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration der messaging REST API mit dem folgenden Befehl an:

  ```
  dspmqweb properties -a
  ```

  Das Feld `mqRestMessagingEnabled` zeigt an, ob die messaging REST API aktiviert ist. Wenn der Wert `True` lautet, ist messaging REST API aktiviert.
- Aktivieren Sie messaging REST API mit dem folgenden Befehl:

  ```
  setmqweb properties -k mqRestMessagingEnabled -v true
  ```
- Inaktivieren Sie messaging REST API mit dem folgenden Befehl:

  ```
  setmqweb properties -k mqRestMessagingEnabled -v false
  ```

**Zugehörige Tasks**

„Verbindungspooling für messaging REST API konfigurieren” auf Seite 896

Sie können die maximale Anzahl gepoolter Verbindungen, die von messaging REST APIverwendet werden können, und das Verhalten von messaging REST API konfigurieren, wenn alle Verbindungen verwendet werden.

„Benutzerkontext konfigurieren, der für die Berechtigung in messaging REST API verwendet wird" auf Seite 899

V 9.3.2 Sie können konfigurieren, welcher Benutzerkontext für die Autorisierung verwendet wird, wenn Sie messaging REST API zum Senden, Empfangen, Durchsuchen oder Veröffentlichen einer Nachricht verwenden. Sie können also auswählen, ob der bei messaging REST APIangemeldete Benutzer oder der Benutzer, der den mqweb-Server gestartet hat, für die Berechtigung verwendet werden soll.

„Verbindungsmodus für messaging REST API konfigurieren" auf Seite 898
Sie können messaging REST API so konfigurieren, dass eine Verbindung zu lokalen oder fernen Warteschlangenmanagern hergestellt wird.

## Verbindungspooling für messaging REST API konfigurieren

Sie können die maximale Anzahl gepoolter Verbindungen, die von messaging REST APIverwendet werden können, und das Verhalten von messaging REST API konfigurieren, wenn alle Verbindungen verwendet werden.

### Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

### Informationen zu diesem Vorgang

Um die Leistung der messaging REST API zu optimieren, werden die Verbindungen zu den IBM MQ-Warteschlangenmanagern gebündelt. Das heißt, anstatt jede REST-Anforderung eine eigene Verbindung zu erstellen, zu verwenden und zu löschen, verwendet jede REST-Anforderung eine Verbindung aus einem Verbindungspool. Standardmäßig sind 20 Verbindungen für jeden Warteschlangenmanagerpool verfügbar und Sie können aus drei Optionen für die Verarbeitung von Anforderungen auswählen, wenn alle Verbindungen verwendet werden:

- Der messaging REST API kann eine neue, nicht gepoolte Verbindung erstellen, die für die Anforderung verwendet werden kann. Dies ist das Standardverhalten.
- Die messaging REST API kann einen Fehler zurückgeben.
- Die messaging REST API wartet, bis eine gepoolte Verbindung verfügbar wird. Die Wartezeit ist dabei unendlich.

Sie können die maximale Anzahl gepoolter Verbindungen und das Standardverhalten von messaging REST API ändern, wenn alle Verbindungen verwendet werden, indem Sie den Befehl **setmqweb properties** verwenden.

## Prozedur

- Zeigen Sie die aktuelle Konfiguration mit dem folgenden Befehl an:

  ```
  dspmqweb properties -a
  ```

  – Das Feld `mqRestMessagingFullPoolBehavior` zeigt das Verhalten von messaging REST API an, wenn alle Verbindungen innerhalb des Pools verwendet werden. Wenn der Wert `block`lautet, muss messaging REST API warten, bis eine Verbindung verfügbar wird. Wenn der Wert `error`lautet, muss messaging REST API einen Fehler zurückgeben. Wenn der Wert `overflow`lautet, muss der messaging REST API eine Verbindung ohne Pool erstellen, die verwendet werden soll, und die Verbindung nach der Verwendung verwerfen.
  – Im Feld `mqRestMessagingMaxPoolSize` wird die maximale Größe des Verbindungspools angezeigt.
- Konfigurieren Sie das Verhalten der messaging REST API, wenn alle Verbindungen im Pool belegt sind, indem Sie den folgenden Befehl eingeben:

  ```
  setmqweb properties -k mqRestMessagingFullPoolBehavior -v action
  ```

  Dabei wird mit *Aktion* die Aktion angegeben, die ausgeführt werden soll. Für *Aktion* kann einer der folgenden Werte verwendet werden:

  **block**
  Wenn alle Verbindungen im Pool im Gebrauch sind, wird gewartet, bis eine Verbindung verfügbar wird.

  **Fehler**
  Wenn alle Verbindungen im Pool im Gebrauch sind, wird ein Fehler zurückgegeben.

  **Überlauf**
  Wenn alle Verbindungen im Pool im Gebrauch sind, erstellen Sie eine Verbindung ohne Pool, die verwendet werden soll, und löschen Sie die Verbindung, nachdem sie verwendet wurde.
- Konfigurieren Sie die maximale Größe des Verbindungspools für jeden Warteschlangenmanager-Pool mit folgendem Befehl:

  ```
  setmqweb properties -k mqRestMessagingMaxPoolSize -v size
  ```

  Dabei wird mit *Größe* die Größe des Pools angegeben.

  **Anmerkung:** Wenn ein hoher Wert für *mqRestMessagingMaxPoolSize* festgelegt ist und viele Warteschlangenmanager verbunden sind, sollten Sie die maximale Größe des Heapspeichers des mqweb-Servers erhöhen. Weitere Informationen finden Sie unter JVM des mqweb-Servers optimieren .

**Zugehörige Tasks**

„messaging REST API aktivieren” auf Seite 895
Mit dem Befehl **setmqweb** können Sie konfigurieren, ob die messaging REST API aktiviert ist. Standardmäßig ist messaging REST API aktiviert.

„Benutzerkontext konfigurieren, der für die Berechtigung in messaging REST API verwendet wird” auf Seite 899

V 9.3.2 Sie können konfigurieren, welcher Benutzerkontext für die Autorisierung verwendet wird, wenn Sie messaging REST API zum Senden, Empfangen, Durchsuchen oder Veröffentlichen einer Nachricht verwenden. Sie können also auswählen, ob der bei messaging REST APIangemeldete Benutzer oder der Benutzer, der den mqweb-Server gestartet hat, für die Berechtigung verwendet werden soll.

„Verbindungsmodus für messaging REST API konfigurieren" auf Seite 898
Sie können messaging REST API so konfigurieren, dass eine Verbindung zu lokalen oder fernen Warteschlangenmanagern hergestellt wird.

## V 9.3.3 Verbindungsmodus für messaging REST API konfigurieren

Sie können messaging REST API so konfigurieren, dass eine Verbindung zu lokalen oder fernen Warteschlangenmanagern hergestellt wird.

### Vorbereitende Schritte

**Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist diese Aufgabe nicht anwendbar. Der eigenständige IBM MQ Web Server unterstützt nur Verbindungen zu fernen Warteschlangenmanagern.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

### Informationen zu diesem Vorgang

Der Standardverbindungsmodus für messaging REST API hängt vom Typ der Installation ab, die den mqweb-Server ausführt:

- In einer IBM MQ -Installation stellt messaging REST API standardmäßig nur eine Verbindung zu lokalen Warteschlangenmanagern in derselben Installation wie der mqweb-Server her. Führen Sie die Schritte in dieser Task aus, um die Verbindungskonfiguration anzuzeigen und zu ändern.
- V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation unterstützt messaging REST API nur Verbindungen zu fernen Warteschlangenmanagern. Die Verbindungskonfiguration kann nicht angezeigt oder geändert werden.

### Prozedur

- Zeigen Sie die aktuelle Konfiguration von messaging REST API mit dem folgenden Befehl an:

```
dspmqweb properties -a
```

Im Feld `mqRestMessagingConnectionMode` wird der aktuelle Verbindungsmodus angezeigt. Wenn der Wert `local`lautet, kann messaging REST API nur eine Verbindung zu Warteschlangenmanagern in derselben Installation wie der mqweb-Server herstellen. Wenn der Wert `remote`lautet, kann messaging REST API eine Verbindung zu fernen Warteschlangenmanagern herstellen.

- Konfigurieren Sie den mqweb-Server so, dass messaging REST API nur eine Verbindung zu Warteschlangenmanagern herstellen kann, die sich in derselben Installation wie der mqweb-Server befinden, indem Sie die folgenden Befehle verwenden:

```
setmqweb properties -k mqRestMessagingConnectionMode -v local
endmqweb
strmqweb
```

- Konfigurieren Sie den mqweb-Server so, dass der messaging REST API mit dem folgenden Befehl eine Verbindung zu fernen Warteschlangenmanagern herstellen kann:

```
setmqweb properties -k mqRestMessagingConnectionMode -v remote
endmqweb
strmqweb
```

## Nächste Schritte

Wenn Sie den mqweb-Server so konfigurieren, dass der messaging REST API eine Verbindung zu fernen Warteschlangenmanagern herstellen kann, müssen Sie Verbindungsinformationen für jeden Warteschlangenmanager angeben, zu dem Sie eine Verbindung herstellen möchten. Weitere Informationen zur Bereitstellung der Verbindungsinformationen finden Sie unter Fernen Warteschlangenmanager für die Verwendung mit messaging REST APIeinrichten.

**Zugehörige Tasks**

„messaging REST API aktivieren” auf Seite 895
Mit dem Befehl **setmqweb** können Sie konfigurieren, ob die messaging REST API aktiviert ist. Standardmäßig ist messaging REST API aktiviert.

„Verbindungspooling für messaging REST API konfigurieren” auf Seite 896
Sie können die maximale Anzahl gepoolter Verbindungen, die von messaging REST APIverwendet werden können, und das Verhalten von messaging REST API konfigurieren, wenn alle Verbindungen verwendet werden.

„Benutzerkontext konfigurieren, der für die Berechtigung in messaging REST API verwendet wird” auf Seite 899
V 9.3.2 Sie können konfigurieren, welcher Benutzerkontext für die Autorisierung verwendet wird, wenn Sie messaging REST API zum Senden, Empfangen, Durchsuchen oder Veröffentlichen einer Nachricht verwenden. Sie können also auswählen, ob der bei messaging REST APIangemeldete Benutzer oder der Benutzer, der den mqweb-Server gestartet hat, für die Berechtigung verwendet werden soll.

# V 9.3.2 Benutzerkontext konfigurieren, der für die Berechtigung in messaging REST API verwendet wird

V 9.3.2 Sie können konfigurieren, welcher Benutzerkontext für die Autorisierung verwendet wird, wenn Sie messaging REST API zum Senden, Empfangen, Durchsuchen oder Veröffentlichen einer Nachricht verwenden. Sie können also auswählen, ob der bei messaging REST APIangemeldete Benutzer oder der Benutzer, der den mqweb-Server gestartet hat, für die Berechtigung verwendet werden soll.

## Vorbereitende Schritte

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- Linux V 9.3.5 Wenn der mqweb-Server Teil einer eigenständigen IBM MQ Web Server -Installation ist, benötigen Sie Schreibzugriff auf die Datei `mqwebuser.xml` im IBM MQ Web Server -Datenverzeichnis.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

## Informationen zu diesem Vorgang

- Wenn die verwendete Benutzer-ID die Benutzer-ID ist, die bei messaging REST APIangemeldet ist, wird die **MQMD.UserIdentifier** auf die Benutzer-ID gesetzt, die bei der REST-API angemeldet ist. **MQMD.AppIdentityData** wird auf die Benutzer-ID gesetzt, die bei der REST API angemeldet ist.
- Wenn die verwendete Benutzer-ID die Benutzer-ID ist, die den mqweb-Server gestartet hat, bleibt das Feld **MQMD.UserIdentifier** leer. **MQMD.AppIdentityData** wird auf die Benutzer-ID gesetzt, die bei der REST API angemeldet ist.

Weitere Informationen zu den Nachrichtendeskriptorabschnitten der IBM MQ -Nachricht finden Sie im Abschnitt MQMD .

## Prozedur

- Zeigen Sie die aktuelle Konfiguration der messaging REST API mit dem folgenden Befehl an:

  ```
  dspmqweb properties -a
  ```

  Das Feld `mqRestMessagingAdoptWebUserContext` zeigt, welche Benutzer-ID für die Autorisierung beim Senden, Veröffentlichen, Empfangen oder Durchsuchen von Nachrichten verwendet wird. Wenn der Wert `True`lautet, wird der Benutzer, der bei messaging REST API angemeldet ist, für die Autorisierung verwendet. Ist der Wert `False` , wird der Benutzer, der den mqweb-Server gestartet hat, für die Berechtigung verwendet.
- Konfigurieren Sie messaging REST API mit dem folgenden Befehl so, dass die Benutzer-ID des Benutzers verwendet wird, der bei messaging REST API zur Autorisierung angemeldet ist:

  ```
  setmqweb properties -k mqRestMessagingAdoptWebUserContext -v true
  ```

  Wenn **mqRestMessagingAdoptWebUserContext** auf `true`gesetzt ist, wird **MQMD.UserIdentifier** auf die Benutzer-ID gesetzt, die bei der REST API angemeldet ist. **MQMD.AppIdentityData** wird auf die Benutzer-ID gesetzt, die bei der REST API angemeldet ist.
- Konfigurieren Sie messaging REST API für die Verwendung der Benutzer-ID des Benutzers, der den mqweb-Server gestartet hat, mit dem folgenden Befehl:

```
setmqweb properties -k mqRestMessagingAdoptWebUserContext -v false
```

Wenn **mqRestMessagingAdoptWebUserContext** auf `false`gesetzt ist, bleibt das Feld **MQMD.UserIdentifier** leer. **MQMD.AppIdentityData** wird auf die Benutzer-ID gesetzt, die bei der REST API angemeldet ist.

**Zugehörige Tasks**

„messaging REST API aktivieren” auf Seite 895
Mit dem Befehl **setmqweb** können Sie konfigurieren, ob die messaging REST API aktiviert ist. Standardmäßig ist messaging REST API aktiviert.

„Verbindungspooling für messaging REST API konfigurieren” auf Seite 896
Sie können die maximale Anzahl gepoolter Verbindungen, die von messaging REST APIverwendet werden können, und das Verhalten von messaging REST API konfigurieren, wenn alle Verbindungen verwendet werden.

„Verbindungsmodus für messaging REST API konfigurieren” auf Seite 898
Sie können messaging REST API so konfigurieren, dass eine Verbindung zu lokalen oder fernen Warteschlangenmanagern hergestellt wird.

# REST API für MFT konfigurieren

Standardmäßig ist die REST API für MFT nicht aktiviert. Sie können konfigurieren, ob REST API for MFT aktiviert ist, den Koordinationswarteschlangenmanager festlegen, den Befehlswarteschlangenmanager festlegen und das MFT -Zeitlimit für die Verbindungswiederherstellung mit dem Befehl **setmqweb properties** angeben.

## Prozedur

- „REST API für MFT aktivieren” auf Seite 901
- „Koordinationswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 902
- „Befehlswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 903
- „REST API für MFT -Zeitlimitwerte konfigurieren” auf Seite 905

## REST API für MFT aktivieren

Bevor Sie REST API für MFTverwenden können, müssen Sie zunächst REST API für MFTaktivieren. Mit dem Befehl **setmqweb** können Sie konfigurieren, ob REST API for MFT aktiviert ist. Standardmäßig ist die REST API für MFT nicht aktiviert.

### Vorbereitende Schritte

**Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist diese Aufgabe nicht anwendbar. REST API for MFT ist in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

### Vorgehensweise

1. Zeigen Sie die aktuelle Konfiguration von REST API für MFT an, indem Sie den folgenden Befehl verwenden:

   ```
   dspmqweb properties -a
   ```

   Das Feld `mqRestMftEnabled` zeigt, ob die REST API für MFT aktiviert ist. Der Wert lautet `True` , wenn REST API für MFT aktiviert ist, andernfalls `False` .
2. Aktivieren oder inaktivieren Sie REST API für MFT mit einem der folgenden Befehle:
   - Aktivieren Sie die REST API für MFT mit dem folgenden Befehl:

     ```
     setmqweb properties -k mqRestMftEnabled -v true
     ```
   - Inaktivieren Sie die REST API für MFT mit dem folgenden Befehl:

     ```
     setmqweb properties -k mqRestMftEnabled -v false
     ```
3. Starten Sie den mqweb-Server erneut, indem Sie die folgenden Befehle eingeben:

   ```
   endmqweb
   strmqweb
   ```

### Nächste Schritte

Wenn Sie REST API für MFTaktiviert haben, müssen Sie den Namen des Koordinationswarteschlangenmanagers festlegen, damit Sie REST API für MFTverwenden können. Weitere Informationen zum Festlegen des Koordinationswarteschlangenmanagers finden Sie im Abschnitt „Koordinationswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 902.

## Koordinationswarteschlangenmanager für REST API for MFT konfigurieren

Bevor Sie REST API for MFTverwenden können, müssen Sie einen Warteschlangenmanager als Koordinationswarteschlangenmanager für die MFT -Transaktionen konfigurieren. Mit dem Befehl **setmqweb** können Sie festlegen, welcher Warteschlangenmanager der Koordinationswarteschlangenmanager ist.

### Vorbereitende Schritte

**Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist diese Aufgabe nicht anwendbar. REST API for MFT ist in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

## Vorgehensweise

1. Zeigen Sie die aktuelle Konfiguration von REST API für MFT an, indem Sie den folgenden Befehl verwenden:

   ```
   dspmqweb properties -a
   ```

   Im Feld `mqRestMftCoordinationQmgr` wird der Name des Koordinations-WS-Managers angezeigt.
2. Konfigurieren Sie den Koordinationswarteschlangenmanager mit dem folgenden Befehl:

   ```
   setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
   ```

   Dabei steht *qmgrName* für den Namen des Koordinations-WS-Managers. Der Koordinationswarteschlangenmanager muss sich auf der Maschine befinden, auf der der mqweb-Server ausgeführt wird. Standardmäßig ist dieser WS-Manager-Name leer. Wenn kein Wert festgelegt ist, funktioniert die REST API für MFT nicht.
3. Starten Sie den mqweb-Server erneut, indem Sie die folgenden Befehle eingeben:

   ```
   endmqweb
   strmqweb
   ```

## Nächste Schritte

- Stellen Sie sicher, dass REST API for MFT aktiviert ist. Weitere Informationen finden Sie unter „REST API für MFT aktivieren” auf Seite 901.
- Wenn Sie REST API for MFT zum Übergeben von Erstellungsanforderungen verwenden wollen, müssen Sie den Namen des Befehlswarteschlangenmanagers festlegen. Wenn Sie beispielsweise einen REST API -Befehl wie **create transfer**verwenden möchten, müssen Sie den Namen des Befehlswarteschlangenmanagers festlegen. Weitere Informationen finden Sie unter „Befehlswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 903.
- Sie können REST API für MFT -Zeitlimitwerte konfigurieren. Das Standardzeitlimit beträgt 30 Minuten. Weitere Informationen finden Sie unter „REST API für MFT -Zeitlimitwerte konfigurieren” auf Seite 905.
- Um REST API für MFTverwenden zu können, muss ein Benutzer beim mqweb-Server authentifiziert sein und einer oder mehreren der Rollen `MFTWebAdmin`oder `MFTWebAdminRO` angehören. Weitere Informationen zum Konfigurieren von Benutzern finden Sie unter Benutzer und Rollen für REST APIkonfigurieren.

# Befehlswarteschlangenmanager für REST API for MFT konfigurieren

Bevor Sie mit REST API for MFT Erstellungsanforderungen übergeben können, müssen Sie den Namen des Befehlswarteschlangenmanagers festlegen. Um beispielsweise die Ressource **create transfer**

zu verwenden, müssen Sie den Namen des Befehlswarteschlangenmanagers festlegen. Sie können den Namen des Befehlswarteschlangenmanagers mit dem Befehl **setmqweb** festlegen.

## Vorbereitende Schritte

**Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist diese Aufgabe nicht anwendbar. REST API for MFT ist in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

## Vorgehensweise

1. Zeigen Sie die aktuelle Konfiguration von REST API für MFT an, indem Sie den folgenden Befehl verwenden:

   ```
   dspmqweb properties -a
   ```

   Im Feld `mqRestMftCommandQmgr` wird der Name des Befehlswarteschlangenmanagers angezeigt.
2. Konfigurieren Sie den Befehlswarteschlangenmanager mit folgendem Befehl:

   ```
   setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
   ```

   Dabei steht *WS-Managername* für den Namen des Befehlswarteschlangenmanagers. Der Befehlswarteschlangenmanager muss sich auf dem System befinden, auf dem der mqweb-Server ausgeführt wird. Standardmäßig ist dieser WS-Manager-Name leer. Wenn kein Wert festgelegt ist, funktioniert REST API for MFT für einen Erstellungsbefehl nicht.
3. Starten Sie den mqweb-Server erneut, indem Sie die folgenden Befehle eingeben:

   ```
   endmqweb
   strmqweb
   ```

## Nächste Schritte

- Stellen Sie sicher, dass REST API for MFT aktiviert ist. Weitere Informationen finden Sie unter „REST API für MFT aktivieren” auf Seite 901.

- Stellen Sie sicher, dass ein Koordinationswarteschlangenmanager definiert ist. Weitere Informationen finden Sie unter „Koordinationswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 902.
- Sie können REST API für MFT -Zeitlimitwerte konfigurieren. Das Standardzeitlimit beträgt 30 Minuten. Weitere Informationen finden Sie unter „REST API für MFT -Zeitlimitwerte konfigurieren” auf Seite 905.
- Um REST API für MFTverwenden zu können, muss ein Benutzer beim mqweb-Server authentifiziert sein und einer oder mehreren der Rollen `MFTWebAdmin`oder `MFTWebAdminRO` angehören. Weitere Informationen zum Konfigurieren von Benutzern finden Sie unter Benutzer und Rollen für REST APIkonfigurieren.

## REST API für MFT -Zeitlimitwerte konfigurieren

Sie können den Zeitraum in Minuten konfigurieren, nach dem REST API for MFT nicht mehr versucht, eine Verbindung zum Koordinationswarteschlangenmanager herzustellen, nachdem die Verbindung unterbrochen wurde. Das Standardzeitlimit beträgt 30 Minuten. Sie können dieses Zeitlimit mit dem Befehl **setmqweb** konfigurieren.

### Vorbereitende Schritte

**Anmerkung:** V 9.3.5 Linux Wenn der mqweb-Server in einer eigenständigen IBM MQ Web Server -Installation ausgeführt wird, ist diese Aufgabe nicht anwendbar. REST API for MFT ist in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar.

Um diese Task ausführen zu können, müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie die Befehle **dspmqweb** und **setmqweb** verwenden können:

- z/OS Unter z/OS müssen Sie über die Berechtigung zum Ausführen der Befehle **dspmqweb** und **setmqweb** und über Schreibzugriff auf die Datei `mqwebuser.xml` verfügen.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

**Achtung:** z/OS

Bevor Sie den Befehl **setmqweb** oder den Befehl **dspmqweb** unter z/OSabsetzen, müssen Sie die Umgebungsvariable WLP_USER_DIR so setzen, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist.

Setzen Sie den folgenden Befehl ab, um die Umgebungsvariable WLP_USER_DIR festzulegen:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb` übergeben wird. For example:

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

Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

### Informationen zu diesem Vorgang

Sie können das Zeitlimit für REST API for MFTkonfigurieren.

REST API for MFT versucht, die Verbindung sofort wiederherzustellen, nachdem die Verbindung zum Koordinationswarteschlangenmanager unterbrochen wurde. Wenn dieser Versuch fehlschlägt, gibt es ein Intervall von fünf Minuten zwischen jedem Verbindungsversuch, bis das Zeitlimit abgelaufen ist. Wenn Sie also einen Wert zwischen 0 und 5 festlegen, wird nur ein Versuch unternommen, die Verbindung herzustellen.

Nachdem das Zeitlimit für die Verbindungswiederholung überschritten wurde, wird der nächste Versuch zur Verbindungswiederherstellung unternommen, wenn eine der Ressourcen von REST API für MFT auf-

gerufen wird. Wenn dieser Verbindungsversuch fehlschlägt, versucht MFT erneut, alle fünf Minuten wieder eine Verbindung herzustellen, bis das Zeitlimit für die Verbindungswiederholung überschritten wurde.

### Vorgehensweise

1. Zeigen Sie die aktuelle Konfiguration von REST API für MFT an, indem Sie den folgenden Befehl verwenden:

   ```
   dspmqweb properties -a
   ```

   Das Feld `mqRestMftReconnectTimeoutInMinutes` zeigt den Zeitlimitwert für die Verbindungswiederherstellung an, bis die MFT Transfer REST-Services die Versuche einstellen, eine Verbindung zum Koordinationswarteschlangenmanager herzustellen.
2. Konfigurieren Sie das Zeitlimit (in Minuten), nach dem REST API für MFT den Versuch einstellt, eine Verbindung zum Koordinationswarteschlangenmanager herzustellen:
   - Setzen Sie das Zeitlimit auf den Standardwert von 30 Minuten zurück:

     ```
     setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -d
     ```
   - Legen Sie das Zeitlimit fest:

     ```
     setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
     ```

     Dabei gibt *time* die Zeit (in Minuten) vor dem Auftreten des Zeitlimits an.

     Wenn dieser Wert zwischen `0-5` eingestellt ist, versucht die REST API für MFT nur einmal, die Verbindung zum Koordinationswarteschlangenmanager wieder herzustellen. Wenn die Verbindung fehlschlägt, werden keine Versuche unternommen, die Verbindung erneut herzustellen, bis die REST API aufgerufen wird.

     Wenn dieser Wert auf `-1` gesetzt ist, versucht die REST API für MFT so lange, die Verbindung wiederherzustellen, bis der Verbindungsaufbau erfolgreich ist.
3. Starten Sie den mqweb-Server erneut, indem Sie die folgenden Befehle eingeben:

   ```
   endmqweb
   strmqweb
   ```

### Nächste Schritte

- Stellen Sie sicher, dass REST API for MFT aktiviert ist. Weitere Informationen finden Sie unter „REST API für MFT aktivieren” auf Seite 901.
- Stellen Sie sicher, dass ein Koordinationswarteschlangenmanager definiert ist. Weitere Informationen finden Sie unter „Koordinationswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 902.
- Wenn Sie REST API for MFT zum Übergeben von Erstellungsanforderungen verwenden wollen, müssen Sie den Namen des Befehlswarteschlangenmanagers festlegen. Wenn Sie beispielsweise einen REST API -Befehl wie **`create transfer`**verwenden möchten, müssen Sie den Namen des Befehlswarteschlangenmanagers festlegen. Weitere Informationen finden Sie unter „Befehlswarteschlangenmanager für REST API for MFT konfigurieren” auf Seite 903.
- Um REST API für MFTverwenden zu können, muss ein Benutzer beim mqweb-Server authentifiziert sein und einer oder mehreren der Rollen `MFTWebAdmin`oder `MFTWebAdminRO` angehören. Weitere Informationen zum Konfigurieren von Benutzern finden Sie unter Benutzer und Rollen für REST APIkonfigurieren.

## Die JVM des mqweb-Servers optimieren

Standardmäßig verwendet der mqweb-Server Java Virtual Machine (JVM) plattformspezifische Standardwerte für Konfigurationsparameter wie die minimale und maximale Größe des Heapspeichers und die Größe des Klassencache.

## Informationen zu diesem Vorgang

Möglicherweise müssen Sie die Standardwerte ändern, um die Leistung zu verbessern oder Probleme zu beheben. Wenn z. B. ein `java.lang.OutOfMemoryError` vom mqweb-Server ausgelöst wird, müssen Sie die maximale Größe des Heapspeichers erhöhen. Sie sollten auch die Größe des Heapspeichers erhöhen, wenn Sie versuchen, eine große Anzahl Warteschlangenobjekte zu laden.

Wenn Probleme mit der Anzeige von Dashboardkonfigurationsinformationen im IBM MQ Consoleauftreten, müssen Sie eine Variable festlegen, die die Dateicodierung der Konfiguration bestimmt. Sie können die Standardwerte in der Datei `jvm.options` ändern.

## Vorgehensweise

1. Öffnen Sie die Datei `jvm.options`.

   Die Datei `jvm.options` befindet sich in einem der folgenden Verzeichnisse:

   - In einer IBM MQ -Installation:
     - Linux AIX Unter AIX oder Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Unter Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`, wobei *MQ_DATA_PATH* der IBM MQ -Datenpfad ist. Dieser Pfad ist der Datenpfad, der während der Installation von IBM MQausgewählt wird. Standardmäßig lautet dieser Pfad `C:\ProgramData\IBM\MQ`.
     - IBM i Unter IBM i: *MQ_DATA_PATH*`/web/installations/Installation1/`
     - z/OS Unter z/OS: *WLP_user_directory*`/servers/mqweb`

       Dabei ist *WLP_Benutzerverzeichnis* das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.
   - V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     Dabei ist *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis, auf das die **MQ_OVERRIDE_DATA_PATH** -Umgebungsvariable verweist.
2. Optional: Legen Sie die maximale Heapspeichergröße fest, indem Sie der Datei die folgende Zeile hinzufügen:

   ```
   -Xmx<MaxSize>m
   ```

   Dabei gibt *MaxSize* die maximale Größe des Heapspeichers in Megabyte an.

   In der folgenden Zeile wird beispielsweise die maximale Größe des Heap-Speichers auf 1 GB gesetzt:

   ```
   -Xmx1024m
   ```
3. Optional: Legen Sie die Mindestgröße des Heapspeichers fest, indem Sie die folgende Zeile zur Datei hinzufügen:

   ```
   -Xms<MinSize>m
   ```

   Dabei gibt *MinSize* die Mindestgröße des Heapspeichers in Megabyte an. Wenn Sie die Mindestgröße des Heapspeichers von der Standardeinstellung erhöhen, kann die Zeit, die zum Starten des mqweb-Servers benötigt wird, reduziert werden.

   In der folgenden Zeile wird beispielsweise die Mindestgröße des Heapspeichers auf 512 MB festgelegt:

   ```
   -Xms512m
   ```

4. Optional: Legen Sie die Größe des Klassencache fest, indem Sie die folgende Zeile zur Datei hinzufügen:

   ```
   -XscmxSizem
   ```

   Dabei gibt *Size* die Größe des Klassencache in MB an.

   Die folgende Zeile setzt z. B. die Größe des Klassencache auf 100 MB:

   ```
   -Xscmx100m
   ```

   Der gemeinsam genutzte Klassencache von Java wird verwendet, um Daten, wie z. B. geladene Klassen und Ahead-Of-Time (AOT) kompilierten Code, zu speichern.

   Der Klassencache verkürzt die Zeit, die zum Starten des mqweb-Servers genommen wurde, erheblich. Wenn der mqweb-Server zum ersten Mal gestartet wird, wird der Klassencache erstellt, und der Server kann eine wichtige Zeit zum Starten nehmen. Nachfolgende Neustarts des Servers werden wesentlich schneller ausgeführt, wenn Klassen aus dem Cache für gemeinsam genutzte Klassen geladen werden können.

   Wenn Sie die Größe des Klassencache von der Standardeinstellung erhöhen, kann die Zeit, die zum Starten des mqweb-Servers benötigt wird, reduziert werden.

   z/OS Der Klassencache wird erneut erstellt, wenn der mqweb-Server auf einem anderen z/OS-System gestartet wird. Daher kann der Start des mqweb-Servers auf einem anderen z/OS-System in einem Sysplex erheblich länger dauern als ein Neustart des Servers auf demselben System.

   Beachten Sie, dass Änderungen an diesem Wert nur wirksam werden, wenn der Klassencache erstellt wird. Der Klassencache wird erstellt, wenn der mqweb-Server zum ersten Mal gestartet wird, oder nach dem Löschen des Klassencache mit dem Java-Klassencache-Dienstprogramm.
5. Erforderlich: Stellen Sie sicher, dass die Datei die folgenden Zeilen enthält, um die Dateicodierung anzugeben, die verwendet wird, wenn der REST API Daten verarbeitet, sowie für die Konfigurationsdaten des Benutzerdashboards in IBM MQ Console:

   ```
   -Dfile.encoding=UTF-8
   -Ddefault.client.encoding=UTF-8
   ```
6. Starten Sie den mqweb-Server erneut.

   z/OS Stoppen Sie unter z/OS die gestartete Task des mqweb-Servers und starten Sie sie erneut.

   Multi Geben Sie auf allen anderen Plattformen die folgenden Befehle in der Befehlszeile ein:

   ```
   endmqweb
   strmqweb
   ```

# Dateistruktur der Installationskomponente IBM MQ Console und REST API

Es gibt zwei Gruppen von Verzeichnisstrukturen, die der Installationskomponente IBM MQ Console und REST API zugeordnet sind. Eine Verzeichnisstruktur enthält Dateien, die bearbeitet werden können. Die andere Verzeichnisstruktur enthält Dateien, die nicht bearbeitet werden können.

## Editierbare Dateien

Die für Benutzer bearbeitbaren Dateien werden im Rahmen der Erstinstallation der Installationskomponente IBM MQ Console und REST API festgelegt. Da diese Dateien bearbeitet werden können, werden die Dateien nicht geändert, wenn die Pflege angewendet wird.

Die Position der vom Benutzer bearbeitbaren Dateien hängt vom Betriebssystem und dem installierten Produktab.

- In einer IBM MQ -Installation befinden sich die vom Benutzer bearbeitbaren Dateien in einem der folgenden Verzeichnisse:
  - Linux AIX Unter AIX oder Linux: `/var/mqm/web/installations/installationName`
  - Windows Unter Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*, wobei *MQ_DATA_PATH* der IBM MQ -Datenpfad ist. Dieser Pfad ist der Datenpfad, der während der Installation von IBM MQausgewählt wird. Standardmäßig lautet dieser Pfad `C:\ProgramData\IBM\MQ`.
  - z/OS Unter z/OS: Das Verzeichnis, das bei der Ausführung des Scripts **crtmqweb** zum Erstellen der mqweb-Serverdefinition angegeben wurde.
- V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST`

  Dabei ist *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis, auf das die **MQ_OVERRIDE_DATA_PATH** -Umgebungsvariable verweist.

Unter diesem Ausgangsverzeichnis sind die folgenden Verzeichnisse und Dateien vorhanden:

| Verzeichnisse und Dateien | Beschreibung |
|---|---|
| `angular.persistence/` | Verzeichnis, in dem die Konfiguration des IBM MQ Console-Dashboards gespeichert ist. |
| `servers/` | WebSphere Liberty -Serververzeichnis. |
| `servers/mqweb` | Verzeichnis, das die Verzeichnisstruktur des mqweb-Servers enthält. |
| `servers/mqweb/logs` | Verzeichnis, das Protokolle für den mqweb-Server enthält. |
| `servers/mqweb/logs/console.log` | Protokoll der grundlegenden Serverstatus-und Operationsnachrichten. |
| `servers/mqweb/logs/ffdc` | FFDC-Ausgabeverzeichnis (FFDC-First Failure Data Capture). |
| `servers/mqweb/logs/messages.log` | Protokoll der Laufzeitnachrichten vom mqweb-Server, einschließlich IBM MQ Console und REST API. Ältere Nachrichten werden in Dateien gespeichert, die als `messages_`*timestamp*`.log` bezeichnet werden. |
| `servers/mqweb/logs/trace.log` | Protokoll des Trace vom mqweb-Server, einschließlich IBM MQ Console und REST API. Der ältere Trace wird in Dateien gespeichert, die als `trace_`*timestamp*`.log` bezeichnet werden. Diese Dateien sind nur vorhanden, wenn die Tracefunktion aktiviert ist. |
| `servers/mqweb/logs/state` | Serverspezifischer Status. |
| `servers/mqweb/server.xml` | Hauptserverkonfigurationsdatei.<br>Diese Datei ist schreibgeschützt. Bearbeiten Sie die `mqwebuser.xml`-Datei, um die Standardkonfiguration zu überschreiben. |
| `servers/mqweb/mqwebuser.xml` | Konfigurationsdatei für IBM MQ Console und REST API. Einstellungen, die in dieser Datei konfiguriert sind, überschreiben die Standardkonfiguration. |

| Verzeichnisse und Dateien | Beschreibung |
|---|---|
| | Sie müssen ein privilegierter Benutzer sein, um diese Datei zu bearbeiten. |
| `servers/mqweb/resources` | Verzeichnis, das verschiedene Serverressourcen, wie z. B. Keystores, enthält. |
| `servers/mqweb/workarea` | Verzeichnis, das vom Server während seiner Funktion erstellt wird. Dieses Verzeichnis wird erstellt, nachdem der Server zum ersten ausgeführt wurde. |

## Nicht bearbeitbare Dateien

Die nicht bearbeitbaren Dateien werden im Rahmen der Erstinstallation der Installationskomponente von IBM MQ Console und REST API festgelegt. Diese Dateien werden aktualisiert, wenn die Wartung angewendet wird.

Die Position der nicht bearbeitbaren Dateien hängt vom Betriebssystem und dem installierten Produktab.

- In einer IBM MQ -Installation befinden sich die nicht bearbeitbaren Dateien in einem der folgenden Verzeichnisse:
  - ALW Unter AIX, Linux, and Windows: *MQ_INSTALLATION_PATH*`/web`
  - IBM i Unter IBM i: *MQ_INSTALLATION_PATH*`/web`
  - z/OS Unter z/OS: *`installation_directory`*`/web/`

    Dabei steht *Installationsverzeichnis* für den Installationspfad von IBM MQ for z/OS UNIX System Services Components.
- V 9.3.5 Linux In einer eigenständigen IBM MQ Web Server -Installation das Verzeichnis, in dem die IBM MQ Web Server -Installationsdatei dekomprimiert wurde.

Die folgende Verzeichnisstruktur und die folgenden Dateien sind an dieser Position vorhanden:

| Verzeichnisse und Dateien | Beschreibung |
|---|---|
| `bin/` | Verzeichnis, das WebSphere Liberty -Befehle enthält.<br>Sie müssen ein privilegierter Benutzer sein, um Scripts in diesem Verzeichnis ausführen zu können. |
| `mq/` | Verzeichnisstruktur, in der verschiedene IBM MQ-Ressourcen enthalten sind. |
| `mq/apps/` | Verzeichnis, das die Anwendungen IBM MQ Console und REST API enthält. |
| `mq/etc/` | |
| `mq/etc/mqweb.xml` | Schreibgeschützt Konfigurationsdatei für den mqweb-Server.<br>Bearbeiten Sie die `mqwebuser.xml`-Datei, um Konfigurationsänderungen vorzunehmen. |
| `mq/libs` | Verzeichnis, das gemeinsam genutzte Bibliotheken für die Verwendung durch IBM MQ Console und REST APIenthält. |
| `mq/samp` | Verzeichnis, das Muster enthält. |

| Verzeichnisse und Dateien | Beschreibung |
|---|---|
| `mq/samp/configuration` | Verzeichnis, das Beispielkonfigurationsdateien enthält, die in die `mqwebuser.xml`-Datei kopiert werden können. |

# Mqweb-Serverkonfiguration sichern und wiederherstellen

Sie können Ihre mqweb-Serverkonfiguration sichern und an derselben oder an einer anderen Position wiederherstellen.

## Vorbereitende Schritte

Bevor Sie Ihre mqweb-Serverkonfiguration wiederherstellen können, müssen Sie IBM MQoder das eigenständige IBM MQ Web Server auf dem System installieren, auf dem Sie den mqweb-Server wiederherstellen wollen. In einer eigenständigen IBM MQ Web Server -Installation müssen Sie den mqweb-Server erstellen, indem Sie die Schritte in „Eigenständigen IBM MQ Web Server konfigurieren” auf Seite 879ausführen.

## Informationen zu diesem Vorgang

Befolgen Sie die Prozedur in dieser Task, um Ihre mqweb-Serverkonfiguration zu sichern und wiederherzustellen. Wenn Sie Ihren mqweb-Server an einer anderen Position wiederherstellen, müssen Sie die mqweb-Serverkonfiguration aktualisieren, um sicherzustellen, dass Verweise auf Dateien korrekt sind.

V 9.3.5 Sie können diese Prozedur auch verwenden, um einen mqweb-Server, der derzeit in einer IBM MQ -Installation ausgeführt wird, auf eine eigenständige IBM MQ Web Server -Installation zu migrieren.

## Vorgehensweise

1. Um die mqweb-Serverkonfiguration zu sichern, kopieren Sie alle Dateien in dem Verzeichnis, das die mqweb-Serverkonfiguration enthält, an Ihre Sicherungsposition.
   - Kopieren Sie in einer IBM MQ -Installation den Inhalt des folgenden Verzeichnisses:
     - Linux AIX Unter AIX oder Linux: `/var/mqm/web/installations/`*`installationName`*
     - Windows Unter Windows: *`MQ_DATA_PATH`*`\web\installations\`*`installationName`*, wobei *MQ_DATA_PATH* der IBM MQ -Datenpfad ist. Dieser Pfad ist der Datenpfad, der während der Installation von IBM MQausgewählt wird. Standardmäßig lautet dieser Pfad `C:\ProgramData\IBM\MQ`.
     - z/OS Unter z/OS: Das WebSphere Liberty -Benutzerverzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen
   - V 9.3.5 Linux Kopieren Sie in einer eigenständigen IBM MQ Web Server -Installation den Inhalt des Verzeichnisses *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST` , wobei *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis ist, auf das die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** verweist.
2. Zum Wiederherstellen der mqweb-Serverkonfiguration ersetzen Sie den Inhalt des Verzeichnisses, das die mqweb-Serverkonfiguration enthält, durch die Dateien, die Sie in Schritt „1” auf Seite 911kopiert haben.
   - Ersetzen Sie in einer IBM MQ -Installation den Inhalt des folgenden Verzeichnisses:
     - Linux AIX Unter AIX oder Linux: `/var/mqm/web/installations/`*`installationName`*

– Windows Unter Windows: *MQ_DATA_PATH*\web\installations*installationName*, wobei *MQ_DATA_PATH* der IBM MQ -Datenpfad ist. Dieser Pfad ist der Datenpfad, der während der Installation von IBM MQausgewählt wird. Standardmäßig lautet dieser Pfad `C:\Program-Data\IBM\MQ`.

– z/OS Unter z/OS: Das WebSphere Liberty -Benutzerverzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen

- V 9.3.5 Linux Ersetzen Sie in einer eigenständigen IBM MQ Web Server -Installation den Inhalt des Verzeichnisses *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST , wobei *MQ_OVERRIDE_DATA_PATH* das IBM MQ Web Server -Datenverzeichnis ist, auf das die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** verweist.

3. Legen Sie das Eigentumsrecht für die Dateien fest, die Sie in Schritt „2” auf Seite 911 wiederhergestellt haben, damit die Benutzer-ID des mqweb-Servers auf die Dateien zugreifen kann.
4. Wenn Sie die mqweb-Serverkonfiguration an einer anderen Position wiederhergestellt haben, ändern Sie den Wert aller Eigenschaften in der mqweb-Serverkonfiguration, die auf Dateien im vorherigen Verzeichnis der mqweb-Serverkonfiguration verweisen.
   a) Bevor Sie den Befehl **setmqweb** oder **dspmqweb** absetzen, legen Sie Ihre Umgebung so fest, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

   - z/OS Legen Sie unter z/OSdie Umgebungsvariable **WLP_USER_DIR** so fest, dass die Variable auf Ihre mqweb-Serverkonfiguration verweist, indem Sie den folgenden Befehl eingeben:

     ```
     export WLP_USER_DIR=WLP_user_directory
     ```

     Dabei ist *WLP_Benutzerverzeichnis* der Name des Verzeichnisses, das an den Befehl **crtmqweb** übergeben wird.

     Weitere Informationen finden Sie im Abschnitt mqweb-Server erstellen.

   - V 9.3.5 Linux Setzen Sie in einer eigenständigen IBM MQ Web Server -Installation die Umgebungsvariable **MQ_OVERRIDE_DATA_PATH** auf das Datenverzeichnis IBM MQ Web Server .
   - In allen anderen Umgebungen müssen Sie keine Aktionen ausführen, um Ihre Umgebung festzulegen.

   b) Zeigt den Wert aller konfigurierbaren Eigenschaften des mqweb-Servers an, die ein Benutzer geändert hat. Geben Sie den folgenden Befehl ein:

   ```
   dspmqweb properties -u
   ```

   c) Wenn die Eigenschaft **remoteKeyfile** angezeigt wird, überprüfen Sie den Wert der Eigenschaft.

   Wenn der Wert der Eigenschaft auf einen Dateipfad im vorherigen Konfigurationsverzeichnis des mqweb-Servers verweist, ändern Sie den Wert so, dass er auf den Dateipfad im neuen Konfigurationsverzeichnis des mqweb-Servers verweist. Setzen Sie den folgenden Befehl ab, um den Wert der Eigenschaft **remoteKeyfile** zu ändern:

   ```
   setmqweb properties -k remoteKeyfile -v path_to_keyfile
   ```

   d) Die Konfiguration des fernen Warteschlangenmanagers des MQWeb-Servers anzeigen. Geben Sie den folgenden Befehl ein:

   ```
   dspmqweb remote -a
   ```

   e) Wenn eine der folgenden Eigenschaften angezeigt wird, überprüfen Sie den Wert der Eigenschaft:

   - **globalTrustStorePath**
   - **globalKeyStorePath**
   - **ccdtURL**

- **keyStorePath**
- **trustStorePath**

Ändern Sie den Wert einer Eigenschaft, die auf einen Dateipfad im vorherigen Konfigurationsverzeichnis des mqweb-Servers verweist, so, dass er auf den Dateipfad im neuen Konfigurationsverzeichnis des mqweb-Servers verweist. Setzen Sie den Befehl **setmqweb remote** ab, um den Wert jeder Eigenschaft zu ändern. Um beispielsweise den Wert der Eigenschaft **keyStorePath** für den fernen Warteschlangenmanager mit dem eindeutigen Namen `remote-QM1`zu ändern, geben Sie folgenden Befehl aus:

```
setmqweb remote -uniqueName remote-QM1 -keyStorePath new_keystore_path
```

Weitere Informationen finden Sie unter setmqweb remote (set mqweb server remote queue manager configuration).

Windows Linux MQ Adv. MQ Adv. VUE MQ Adv. z/OS

# Aspera gateway -Verbindung auf Linux -oder Windows -Plattformen definieren

IBM Aspera faspio Gateway stellt einen schnellen TCP/IP-Tunnel bereit, der den Netzdurchsatz für IBM MQerheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert.

## Informationen zu diesem Vorgang

Mit dem Aspera gateway kann die Leistung von Warteschlangenmanagerkanälen verbessert werden. Es ist besonders effektiv, wenn das Netz eine längere Latenzzeit hat oder dazu neigt, Pakete zu verlieren, und wird normalerweise dazu verwendet, die Verbindung zwischen Warteschlangenmanagern in verschiedenen Rechenzentren zu beschleunigen.

**Anmerkung:** Bei einem schnellen Netz, in das keine Pakete verliert, führt die Verwendung des Aspera gateways zu einer Leistungsminderung. Daher sollten Sie vor und nach dem Definieren einer Aspera gateway-Verbindung die Netzleistung überprüfen.

Sie definieren ein Aspera gateway an jedem Ende der IP-Netzverbindung und stellen anschließend mithilfe von TCP/IP eine Verbindung zwischen den Warteschlangenmanagerkanälen mit jedem Gateway her. Ein Warteschlangenmanager muss nicht im gleichen System wie das Aspera gateway ausgeführt werden, das er verwendet, und mehrere Warteschlangenmanager können das gleiche Gateway verwenden.

Um das Aspera gateway verwenden zu können, müssen Sie mindestens eine der folgenden Berechtigungen besitzen:

- IBM MQ Advanced for Multiplatforms
- IBM MQ Appliance
- IBM MQ Advanced for z/OS VUE
- V 9.3.4 LTS IBM MQ Advanced for z/OS, entweder Long Term Support oder Continuous Delivery aus IBM MQ 9.3.4

Sie können Aspera gateway auf einer der folgenden Plattformen bereitstellen:

- Linux for x86-64
- Linux on Power Systems - Little Endian
- Linux for IBM Z
- Windows -Weitere Informationen zur Plattformunterstützung unter Windows finden Sie im IBM Aspera faspio Gateway -Dokumentation.

Die Verwendung des Aspera gateways ist auf IBM MQ-Nachrichten begrenzt, es sei denn, das Gateway verfügt über eine gesonderte Berechtigung.

Warteschlangenmanager, die das Aspera gateway verwenden, können auf jeder unterstützten Plattform ausgeführt werden. Eine vollständige Liste der unterstützten Plattformen finden Sie im Abschnitt In der Produktdokumentation verwendete Symbole.

Überprüfen Sie für jeden Warteschlangenmanager, der sich nicht im gleichen System wie das Aspera gateway befindet, das er verwendet, ob es eine schnelle Netzverbindung zwischen dem Warteschlangenmanager und dem Aspera gateway gibt.

Sie verwenden eine `toml`-Datei, um eine Gateway-Definition zu erstellen, die die Ports für eingehende und abgehende Daten definiert, die das Gateway verwendet. Eine `toml`-Beispieldatei wird mit dem Aspera gateway geliefert. Die Definition des abgehenden Gateways definiert die Verbindung vom lokalen Warteschlangenmanager zum Gateway und vom lokalen Gateway zum fernen Gateway. Die Definition des eingehenden Gateways definiert die Verbindung vom fernen Gateway zum lokalen Gateway und vom lokalen Gateway zum lokalen Warteschlangenmanager.

In den folgenden Schritten finden Sie die Basisanweisungen zur Einrichtung und Ausführung. Ausführlichere Informationen finden Sie in der IBM Aspera faspio Gateway-Dokumentation.

## Vorgehensweise

1. Rufen Sie das Aspera gateway-Installationsimage ab.

   Multi Für Multiplatforms laden Sie Aspera gateway von Passport Advantage herunter. Der Download trägt die Bezeichnung "IBM Aspera faspio Continuous Delivery Release for IBM MQ V9.3 Multiplatform Multilingual eAssembly". Sie wird nur aufgrund der Geschwindigkeit der Änderung in diesem Bereich als Continuous Delivery (CD)-Image bereitgestellt. Dies bedeutet, dass Aktualisierungen bei der Häufigkeit der CD-Releases erforderlich sind und Sie sie auf einem beliebigen IBM MQ-System installieren können, für das die Berechtigung IBM MQ Advanced for Multiplatforms oder IBM MQ Appliance besteht. Um diese eAssemblyherunterzuladen, rufen Sie Download IBM MQ 9.3 auf und klicken Sie auf die Registerkarte für das erforderliche Release. Die eAssembly enthält Installationsimages für alle Plattformen, auf denen das Gateway verfügbar ist. V 9.3.0 V 9.3.0 Die eAssembly enthält auch eine `ibm-faspio-license.zip` -Datei, die eine Lizenzdatei enthält.

   V 9.3.4 LTS MQ Adv. VUE MQ Adv. z/OS Wenn Ihr IBM MQ -System über IBM MQ Advanced for z/OS VUE -Berechtigung oder IBM MQ Advanced for z/OS -Berechtigung, entweder Long Term Supportoder Continuous Delivery von IBM MQ 9.3.4verfügt, erhalten Sie die Aspera gateway aus der Connector Pack-Komponente, die Teil der SMP/E-Installation ist.

   MQ Adv. VUE MQ Adv. z/OS Die Dateien für IBM MQ Advanced for z/OS VUE und IBM MQ Advanced for z/OS lauten wie folgt:

   *Tabelle 52. Dateinamen und faspio-Versionsnummern nach Plattform und IBM MQ -Version*

   | Plattform | Dateiname | faspio-Versionsnummer |
   |---|---|---|
   | Linux for x86-64 | V 9.3.4 `M0C5LEN.zip` | 1.3.3 |
   | Linux on Power Systems - Little Endian | V 9.3.4 `M0C5MEN.zip` | 1.3.3 |
   | Linux for IBM Z | V 9.3.4 `M0C5NEN.zip` | 1.3.3 |
   | Windows | V 9.3.4 `M0C5PEN.zip` | 1.3.3 |
   | Linux for x86-64 | V 9.3.3 `M0B2XEN.zip` | 1.3.2 |

| *Tabelle 52. Dateinamen und faspio-Versionsnummern nach Plattform und IBM MQ -Version (Forts.)* | | |
|---|---|---|
| **Plattform** | **Dateiname** | **faspio-Versionsnummer** |
| Linux on Power Systems - Little Endian | V 9.3.3 M0B2YEN.zip | 1.3.2 |
| Linux for IBM Z | V 9.3.3 M0B2ZEN.zip | 1.3.2 |
| Windows | V 9.3.3 M0B30EN.zip | 1.3.2 |
| Linux for x86-64 | V 9.3.2 M090HEN.zip | 1.3.1 |
| Linux on Power Systems - Little Endian | V 9.3.2 M090JEN.zip | 1.3.1 |
| Linux for IBM Z | V 9.3.2 M090KEN.zip | 1.3.1 |
| Windows | V 9.3.2 M090LEN.zip | 1.3.1 |
| Linux for x86-64 | V 9.3.0 M0559EN.zip | 1.3.0 |
| Linux on Power Systems - Little Endian | V 9.3.0 M055BEN.zip | 1.3.0 |
| Linux for IBM Z | V 9.3.0 M055CEN.zip | 1.3.0 |
| Windows | V 9.3.0 M055DEN.zip | 1.3.0 |

Beachten Sie, dass Aspera gateway nicht nativ unter z/OSausgeführt werden kann.

V 9.3.0 MQ Adv. VUE V 9.3.0 MQ Adv. z/OS Neben den Installationsimages enthält das Verzeichnis `fasp` die Datei `M05QKEN.zip`, die eine Lizenzdatei enthält.

2. Kopieren Sie das Aspera gateway-Installationsimage auf die beiden Maschinen, die das Gateway ausführen. Extrahieren Sie dann das Gateway und installieren Sie es.

   V 9.3.0 V 9.3.0 Verwenden Sie die Lizenzdatei in `ibm-faspio-license.zip` (Multiplatforms) oder `M05QKEN.zip` (z/OS). Weitere Informationen finden Sie in der Dokumentation zu IBM Aspera faspio Gateway :

   - Linux Installation unter Linux
   - Windows Installation unter Windows
3. Konfigurieren und sichern Sie jedes Gateway.

   V 9.3.0 V 9.3.0 Weitere Informationen finden Sie in der Dokumentation zu IBM Aspera faspio Gateway:

   - Gateway-Konfigurationsdatei konfigurieren
   - Gateway schützen
4. Ändern Sie an jedem Ende der Netzverbindung die Kanaldefinition, um eine Verbindung zu dem Port herzustellen, an dem das lokale Gateway empfangsbereit ist.
5. Starten Sie den Gateway-Service.

V 9.3.0 V 9.3.0 Weitere Informationen finden Sie in der Dokumentation zu IBM Aspera faspio Gateway:

- Linux Starten auf Linux
- Windows Starten auf Windows

6. Starten Sie die Kanäle neu.

   Ihre Warteschlangenmanager kommunizieren jetzt über eine Aspera gateway-Verbindung.

**Beispiel**

In diesem Beispiel wird eine Aspera gateway-Verbindung auf zwei Systemen definiert, die unter Linux ausgeführt werden. Die Konfiguration sieht wie folgt aus:

- Die IP-Adresse des Systems mit dem lokalen Gateway lautet 9.20.193.107. Die IP-Adresse des Systems mit dem fernen Gateway lautet 9.20.192.115.
- Der lokale Warteschlangenmanager wird auf einem System mit der IP-Adresse 9.20.121.5 ausgeführt. Der ferne Warteschlangenmanager wird auf einem System mit der IP-Adresse 9.20.121.25 ausgeführt. Beide Warteschlangenmanager sind an Port 1414 empfangsbereit.
- Der Warteschlangenmanagerkanal im lokalen Warteschlangenmanager wird so geändert, dass eine Verbindung zum lokalen Aspera gateway mit **conname** 9.20.193.107 (1500) hergestellt wird. Der Warteschlangenmanagerkanal auf dem fernen Warteschlangenmanager wird geändert, um über **conname** 9.20.192.115 (1500) eine Verbindung zum fernen Aspera gateway herzustellen.
- V 9.3.0 V 9.3.0 Ab IBM Aspera faspio Gateway 1.2ist TLS standardmäßig aktiviert. Wenn Sie TLS mit dem Gateway konfigurieren möchten, lesen Sie die Informationen in Gateway sichern in der Dokumentation zu IBM Aspera faspio Gateway.

1. Definieren Sie eine Aspera gateway-Verbindung auf dem System mit dem lokalen Gateway:
   - Installieren Sie das Aspera gateway:
     - Linux Verwenden Sie unter Linux den folgenden Befehl:

       ```
       rpm -ivh ibm-faspio-gateway-<version>.x86_64.rpm
       ```

   - Ändern Sie die `gateway.toml`-Datei in dem Verzeichnis, das von der Installation erstellt wurde:

     Bearbeiten Sie die Datei, um die Definitionen für das lokale Gateway festzulegen.

     ```
     [[bridge]]
         name = "Outbound"
         [bridge.local]
                protocol = "tcp"
                    host = "9.20.193.107"
                    port = 1500
         tls_enabled = false

         [bridge.forward]
                protocol = "fasp"
                    host = "9.20.192.115"
                    port = 1600
         tls_enabled = false

     [[bridge]]
         name = "Inbound"
         [bridge.local]
                protocol = "fasp"
                    host = "9.20.193.107"
                    port = 1600
         tls_enabled = false

         [bridge.forward]
                protocol = "tcp"
                    host = "9.20.121.5"
                    port = 1414
         tls_enabled = false
     ```

- V 9.3.0 V 9.3.0 Kopieren Sie die Datei `aspera-license` von `ibm-faspio-license.zip` (Multiplatforms) oder `M05QKEN.zip` (z/OS) in `/usr/local/etc/faspio/`.

2. Wiederholen Sie den vorherigen Schritt, um eine Aspera gateway-Verbindung auf dem System mit dem fernen Gateway zu definieren.
   - Ändern Sie die Datei `gateway.toml` in dem Verzeichnis, das von der Installation erstellt wurde. Bearbeiten Sie die Datei, um die Definitionen für das ferne Gateway festzulegen:

```
[[bridge]]
    name = "Outbound"
    [bridge.local]
           protocol = "tcp"
               host = "9.20.193.107"
               port = 1500
    tls_enabled = false

    [bridge.forward]
           protocol = "fasp"
               host = "9.20.192.115"
               port = 1600
    tls_enabled = false

[[bridge]]
    name = "Inbound"
    [bridge.local]
           protocol = "fasp"
               host = "9.20.193.107"
               port = 1600
    tls_enabled = false

    [bridge.forward]
           protocol = "tcp"
               host = "9.20.121.5"
               port = 1414
    tls_enabled = false
```

   - V 9.3.0 V 9.3.0 Kopieren Sie die Datei `aspera-license` von `ibm-faspio-license.zip` (Multiplatforms) oder `M05QKEN.zip` (z/OS) in `/usr/local/etc/faspio/`.

3. Ändern Sie an jedem Ende der Verbindung die Kanaldefinition, um eine Verbindung zu dem Port herzustellen, an dem das lokale Gateway empfangsbereit ist.
   - Ändern Sie den Warteschlangenmanagerkanal auf dem lokalen WS-Manager so, dass eine Verbindung zum lokalen Aspera gateway mit **conname** 9.20.193.107 (1500) hergestellt wird.
   - Ändern Sie den Warteschlangenmanagerkanal auf dem fernen Warteschlangenmanager, um über **conname** 9.20.192.115 (1500) eine Verbindung zum fernen Aspera gateway herzustellen.
4. Starten Sie das lokale Gateway, indem Sie den folgenden Befehl auf dem System mit dem lokalen Gateway ausführen:
   - Linux

```
sudo systemctl start faspio-gateway
```

5. Starten Sie das ferne Gateway, indem Sie den folgenden Befehl auf dem System mit dem fernen Gateway ausführen:
   - Linux

```
sudo systemctl start faspio-gateway
```

6. Starten Sie die Kanäle neu.

## Nächste Schritte

Das Aspera gateway übergibt die empfangenen Daten, ohne sie in irgendeiner Weise zu interpretieren. Dies bedeutet, dass Sie TLS zwischen den Warteschlangenmanagerkanälen konfigurieren können, die das Aspera gateway verwenden, weil die Gateway-Verbindung keine Kenntnis über das TLS-Handshake-

verfahren hat. Dies bedeutet auch, dass Warteschlangenmanager auf allen unterstützten IBM MQ-Plattformen das Aspera gateway verwenden können.

Für die Verwendung eines Multi-Instanz-Warteschlangenmanagers mit dem Gateway konfigurieren Sie Gateway-Definitionen für jede Instanz des Warteschlangenmanagers.

**Anmerkung:** Aspera gateway wurde nur mit Warteschlangenmanagerkanälen getestet. Es wurde nicht mit Clientkanälen getestet. Dies liegt daran, dass die geplante Verwendung für Aspera gateway darin besteht, ferne Warteschlangenmanager über ein langsames Netz zu verbinden, während Clientanwendungen normalerweise über ein schnelles Netz eine Verbindung zu Warteschlangenmanagern in einem lokalen Rechenzentrum herstellen.

**Zugehörige Konzepte**

Roadmap für Aspera gateway

**Zugehörige Verweise**

„Zu verwendende Übertragungsart" auf Seite 16
Unterschiedliche Plattformen unterstützen unterschiedliche Kommunikationsprotokolle. Ihre Auswahl des Übertragungsprotokolls hängt von Ihrer Kombination von IBM MQ MQI client- und Serverplattformen ab.

Dokumentation zu IBM Aspera faspio Gateway

# Multi IBM MQ für die Verwendung mit dem IBM Cloud Private -Messservice konfigurieren

IBM MQ für die Verwendung mit dem IBM Cloud Private -Messservice konfigurieren, um Start-und Nutzungsinformationen des Warteschlangenmanagers zu berichten und anzuzeigen.

## Vorbereitende Schritte

Bevor Sie Ihre IBM MQ-Warteschlangenmanager für die Verwendung eines IBM Cloud Private-Service konfigurieren, müssen Sie ein IBM Cloud-Konto erstellen. Informationen zum Erstellen Ihres Kontos finden Sie unter Für IBM Cloud anmelden .

## Informationen zu diesem Vorgang

Durch die Verwendung von IBM Cloud Private-Messservicekönnen Sie Ihre On-Premises-IBM-Produkte mit Ihrer Serviceinstanz in IBM Cloud Private verbinden und alle registrierten Produkte in Ihrem Unternehmen in einem einzigen Dashboard anzeigen.

Sie können Ihre AIX-, Linux- und Windows-Warteschlangenmanager für Ihre Messserviceinstanz konfigurieren und mit dieser verbinden und deren Start- und Nutzungsinformationen anzeigen. Auf anderen Plattformen als Linux-Container-Umgebungen können die Daten jedoch nicht zur Unterstützung von stundengenauen containerbasierten Preisstrukturlizenzen verwendet werden.

Wenn Sie anstelle der standardmäßigen stundenweisen Lizenzmetrik die Nutzungsdaten für einen monatlichen VPC-Lizenztyp aufzeichnen möchten, setzen Sie die Umgebungsvariable `AMQ_LICENSING_METRIC=VPCMonthlyPeak`. Dies bewirkt, dass der Warteschlangenmanager anstelle des standardmäßigen Hochladens von Daten, die sich auf stundenweise Container-basierte Lizenzen beziehen, Daten hochlädt, die sich auf monatliche VPC-Lizenztypen beziehen.

Verwenden Sie die folgenden Attribute mit der Zeilengruppe `ReportingService` in der Datei `qm.ini`:

**APIKeyFile**
Speicherort der Textdatei mit dem Wert **APIKey** für die Messserviceinstanz.

**CapacityReporting**
Schreibt die Fehlerprotokollnachrichten in regelmäßigen Abständen in die AMQERR-Protokolle im folgenden Format:

```
4/22/2020 01:44:29 PM - Process(1274.1) User(bld-adm) Program(amqmgr0)
Host(8b3b83f2bc7d) Installation(Docker)
```

```
VRMF(9.2.0.0)
Time(2020-04-22T13:44:29.295Z)
ArithInsert1(300)
CommentInsert1(8.5)
CommentInsert2(IBM MQ Advanced)
```

Die vom Attribut **CapacityReporting** erzeugten Informationen werden in die Nachricht AMQ5064 eingefügt. Dies gibt Ihnen einen besseren Einblick darin, wie stark IBM MQ von Ihrem Unternehmen verwendet wird:

**AMQ5064**
Dieser WS-Manager ist seit 300 Sekunden aktiv. Es wird derzeit mit 8,5 Kernen ausgeführt. Der Lizenztyp ist IBM MQ Advanced.

**Bewertung**
0: Informationen

**Erklärung**
Dies ist eine Informationsnachricht für die Verwendungsüberwachung.

**Antwort**
Keine.

**LicensingGroup**
Die Abrechnungsgruppe, zu der der Warteschlangenmanager gehört. Dies wirkt sich auf die Art und Weise aus, wie Daten in Berichten gruppiert werden, die vom Messservice generiert werden.

**ServiceURL**
Die IBM Cloud Private-Serviceadresse.

**ServiceProxy**
Die URL und der Port für den HTTP-Proxy, der verwendet werden kann, wenn die Warteschlangenmanager keinen direkten Zugriff auf das Netz haben, auf dem der Messservice ausgeführt wird.

Sie können die Hosts sehen, auf denen Ihre Produkte installiert sind, die Produktversionen, die Sie verwenden, und die Plattformen, auf denen sie ausgeführt werden. Aus den für jedes Produkt angezeigten Messwerten auf hoher Ebene können Sie einen Überblick darüber geben, wie hoch die Auslastung der Workloads ist. Bei IBM MQ können Sie sehen, welche Warteschlangenmanager eine höhere Arbeitslast haben und welche weniger stark ausgelastet sind.

Wenn ein Warteschlangenmanager so konfiguriert ist, dass er eine Verbindung zu einer Instanz des Messservice herstellt, werden die folgenden Informationen an IBM Cloud Private gemeldet:

- Name des IBM MQ-Warteschlangenmanagers
- IBM MQ-Warteschlangenmanagerkennung
- IBM MQ-Installationsstammverzeichnis
- Installierte IBM MQ-Komponenten (Name und Version)
- Hostname
- Name des Hostbetriebssystems
- Version des Hostbetriebssystems
- Nutzungsinformationen des virtuellen Prozessorkerns (VPC) für den IBM MQ-Warteschlangenmanager

Sie können die Warteschlangenmanager-VPC-Nutzungsmetriken in Ihrem Dashboard für die Messservice-Instanz überwachen.

## Prozedur

- Konfigurieren Sie einen Warteschlangenmanager für die Verwendung mit der Messservice-Instanz in IBM Cloud Private.
- Stellen Sie über einen HTTP-Proxy eine Verbindung zum IBM Cloud Private-Messservice her.
- Beheben Sie die Fehler bei der Verbindung zum IBM Cloud Private-Messservice.

**Zugehörige Verweise**

Preismessgröße für virtuelle Prozessorkerne (VPCs)

## Multi Warteschlangenmanager für die Verwendung mit der Messservice-Instanz unter IBM Cloud Private konfigurieren

Richten Sie die Sicherheits- und IBM Cloud-Registrierungsinformationen für Ihren Warteschlangenmanager ein und stellen Sie dann eine Verbindung zu der bereits erstellten Messservice-Instanz her.

### Informationen zu diesem Vorgang

Das Dashboard Ihrer IBM Cloud Private-Messservice-Instanz zeigt nur für diejenigen Warteschlangenmanager Daten an, die so konfiguriert sind, dass sie die Sicherheits- und IBM Cloud Private-Registrierungsinformationen enthalten.

### Vorgehensweise

1. Befolgen Sie die dokumentierten ICP-Schritte zum Erstellen einer Service-ID unter:
   Erstellen einer Service-ID mithilfe von IBM Cloud Private-CLI.
2. Befolgen Sie die dokumentierten ICP-Schritte zum Erstellen eines API-Schlüssels unter:
   API-Schlüsselmanagement-APIs .
3. Laden Sie die TLS-Zertifikate aus dem ICP-Cluster herunter.
   Notieren Sie sich die Position, an die Sie die Zertifikate heruntergeladen haben. Sie können die heruntergeladenen Zertifikate dem Keystore für Ihren Warteschlangenmanager hinzufügen, siehe Schritt „9” auf Seite 921.
4. Erstellen Sie eine Textdatei `apikeyfile.txt` und fügen Sie den **`API key`** -Wert hinzu, den Sie in die vorherige Task kopiert haben.
   Notieren Sie sich die Position des `apikeyfile.txt`, damit Sie den Pfad in Schritt 8 einschließen können. Diese Datei muss vom Warteschlangenmanager-Benutzer (*'mqm*' auf AIX and Linux-Systemen) gelesen werden. Die Datei darf nur die **`API key`** selbst enthalten, keine JSON-Nutzdaten, z. B. `d9c11b45-4dda-4de4-c0b2-2e4e1004dc64`.
5. Erstellen Sie den Warteschlangenmanager, z. B. *QM1*.
   Weitere Informationen finden Sie im Abschnitt Warteschlangenmanager auf Multiplatforms erstellen und verwalten .
6. Starten Sie den WS-Manager *QM1*.
   Weitere Informationen finden Sie unter Warteschlangenmanager starten.
7. Denken Sie daran, Ihre IBM MQ-Befehlszeilenumgebung vor der Ausführung von IBM MQ-Befehlen einzurichten.
   Führen Sie den Befehl **`setmqenv`** aus.

   AIX Unter AIX:

   ```
   . /usr/mqm/bin/setmqenv -s
   ```

   Linux Unter Linux:

   ```
   . /opt/mqm/bin/setmqenv -s
   ```

   Windows Unter Windows:

   ```
   "C:\Program Files\IBM\MQ\bin\setmqenv.cmd" -n installation name
   ```
8. Erstellen Sie einen SSL-Truststore für den WS-Manager *QM1*.

AIX Beginnen Sie mit der Erstellung des Zertifikatsspeichers für vertrauenswürdige Zertifikate in AIX:

```
runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-expire 30 -stash
```

Linux Unter Linux:

```
runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-expire 30 -stash
```

Windows Unter Windows:

```
runmqckm -keydb -create -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type
cms -expire 30 -stash
```

9. Fügen Sie die digitalen Zertifikate, die Sie in Schritt „3” auf Seite 920 heruntergeladen haben, zum Truststore des WS-Managers hinzu.

   AIX Unter AIX:

   ```
   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -la
   bel RootCA
   -file Download_location/RootCA.crt -format ascii -trust enable

   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -la
   bel ServerCert
   -file Download_location/CERT.crt -format ascii -trust enable
   ```

   Linux Unter Linux:

   ```
   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -la
   bel RootCA
   -file Download_location/RootCA.crt -format ascii -trust enable

   runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -la
   bel ServerCert
   -file Download_location/CERT.crt -format ascii -trust enable
   ```

   Windows Unter Windows:

   ```
   runmqckm -cert -add -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms
   -label RootCA
   -file "Download_location\RootCA.crt" -format ascii -trust enable

   runmqckm -cert -add -db "C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key.kdb" -pw password -type
   cms -label ServerCert
   -file "Download_location\CERT.crt" -format ascii -trust enable
   ```

10. Fügen Sie die neue Zeilengruppe `ReportingService` mit dem Pfad `apikeyfile` der Datei `qm.ini` des Warteschlangenmanagers hinzu:

    ```
    ReportingService:
       APIKeyFile=APIKey file location/apikeyfile.txt
    ```

11. Fügen Sie den **`API host`** -Wert zur Datei `qm.ini` hinzu.

    Der Zeilengruppenabschnitt `ReportingService` enthält jetzt Pfad zu den Werten `apikeyfile` und **`API host`** (**`ServiceURL`**):

    ```
    ReportingService:
       APIKeyFile=APIKey file location/apikeyfile.txt
       ServiceURL=https://productinsights-api.ng.bluemix.net
    ```

    Speichern und beenden Sie die Datei `qm.ini`.

12. Starten Sie den WS-Manager erneut, damit die Änderungen wirksam werden.

Möglicherweise werden Sie aufgefordert, die Berechtigung für den WS-Managerprozess **amqzmur0** zu erteilen, um auf das Netz zuzugreifen. Der Zugriff ist erforderlich, damit der Warteschlangenmanager Kontakt zum Messservice aufnehmen kann.

13. Zeigen Sie in Ihrer Messservice-Instanz die Informationen zum Warteschlangenmanager *QM1* an.

    Wenn der Berichtsstatus aktiv ist, werden die Start- und Nutzungsinformationen für alle Integrationsserver auf dem angegebenen Integrationsknoten an den Messservice gemeldet. Die Nutzungsinformationen werden alle 15 Minuten aktualisiert.
14. Optional: Stoppen Sie die Berichterstattung eines Warteschlangenmanagers an den Messservice, indem Sie die Zeilengruppe `ReportingService` aus der Datei `qm.ini` des Warteschlangenmanagers entfernen, und starten Sie den Warteschlangenmanager erneut.
15. Optional: Überprüfen Sie die Diagnoseinformationen in der Protokolldatei des Warteschlangenmanagers, wenn der Warteschlangenmanager keine Start- oder Nutzungsinformationen an den Messservice melden kann.

    Amend für AIX

    AIX Unter AIX:

    ```
    /var/mqm/qmgrs/QM1/errors/AMQERR0*.log
    ```

    Linux Unter Linux:

    ```
    /var/mqm/qmgrs/QM1/errors/AMQERR0*.log
    ```

    Windows Unter Windows:

    ```
    C:\ProgramData\IBM\MQ\errors\AMQERR0*.log
    ```

### Ergebnisse

Sie haben eine Messservice-Instanz erstellt und Ihren Warteschlangenmanager so konfiguriert, dass er eine Verbindung zu der Instanz herstellt. Sie können die Informationen zu Ihrem Warteschlangenmanager im Dashboard der Messservice-Instanz anzeigen.

## Multi Verbindung zum IBM Cloud Private-Messservice über einen HTTP-Proxy herstellen

Wenn Ihr Warteschlangenmanager auf einem System ausgeführt wird, das keinen direkten Zugriff auf Ihren ICP-Cluster hat, können Sie einen von Ihrem Unternehmen bereitgestellten HTTP-Proxy verwenden, um eine Verbindung zu Ihrer Messservice-Instanz in IBM Cloud Private herzustellen.

### Vorbereitende Schritte

Sie haben die Sicherheit konfiguriert, die **API key** -und Service-URL der `qm.ini` -Datei für Ihren Warteschlangenmanager hinzugefügt.

### Informationen zu diesem Vorgang

Verwenden Sie diese Task, um Ihren Warteschlangenmanager so zu konfigurieren, dass er über einen von Ihrem Unternehmen bereitgestellten HTTP-Proxy eine Verbindung zu der Messservice-Instanz in IBM Cloud Private herstellt.

### Prozedur

- Fügen Sie ein Service-Proxy-Attribut zur IBM Cloud Private-Registrierungszeilengruppe Ihrer `qm.ini`-Datei hinzu.

  Sie können das Attribut **ServiceProxy** wie folgt festlegen:

– Eine URL, die das Präfix http:// und optional den Port enthält. Wenn Sie den Port nicht angeben, wird *1080* verwendet.

```
ReportingService:
   ServiceProxy=http://myorgproxy.net:1080
```

**Anmerkung:** Der Parameter **ServiceProxy** muss auf eine gültige URL http:// gesetzt werden. Andere Proxy-Protokolle, z. B. HTTPS und SOCKS, werden nicht unterstützt.

- Starten Sie den WS-Manager erneut, bevor die Änderungen wirksam werden.

## Multi Fehlerbehebung für die Verbindung zum Messservice

Fehlerbehebungsempfehlungen für Fehler, die auftreten können, wenn Sie den Warteschlangenmanager mit einer Messservice-Instanz verbinden.

### Der Warteschlangenmanager kann sich bei dem konfigurierten Messservice nicht registrieren oder kann keine Nutzungsmetriken auf diesen hochladen.

Überprüfen Sie, ob der WS-Manager Zugriff auf das Netz hat. Der **APIKey** -Wert in der API-Schlüsseldatei ist falsch. Stellen Sie sicher, dass die IBM Global Security Kit (GSKit)-Komponente installiert ist.

### Ungültige Zeilengruppe `qm.ini`

Es wurde eine ungültige Zeilengruppe `qm.ini` gefunden. Überprüfen Sie das Fehlerprotokoll auf weitere Informationen.

### Ungültiger HTTP-Service-Proxy-Parameter

Der Wert für das Attribut **ServiceProxy** für die Zeilengruppe `ReportingService` des Warteschlangenmanagers ist nicht ordnungsgemäß konfiguriert. Der WS-Manager registriert sich nicht beim Service. Der Parameter **ServiceProxy** muss auf eine gültige URL http:// gesetzt werden. Andere Proxy-Protokolle, z. B. HTTPS und SOCKS, werden nicht unterstützt.

## Linux Deprecated IBM MQ für die Verwendung mit Push-Themen und Plattformereignissen für Salesforce konfigurieren

Verwenden Sie diese Informationen, um die Sicherheit und die Verbindungen zu Salesforce und Ihrem IBM MQ-Netz zu konfigurieren, indem Sie die IBM MQ Bridge to Salesforce konfigurieren und anschließend ausführen.

### Vorbereitende Schritte

**Anmerkung:** Deprecated IBM MQ Bridge to Salesforce ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-341). Die Salesforce -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.

- IBM MQ Bridge to Salesforce ist unter Linux für x86-64 (64 Bit) verfügbar. Für die Verbindung zu Warteschlangenmanagern, die unter IBM WebSphere MQ 6.0 und früher ausgeführt werden, wird die Bridge nicht unterstützt.
- Ab IBM MQ 9.2.0 kann ein Warteschlangenmanager mehrere Brückeninstanzen unterstützen, wenn sie entsprechend konfiguriert wurden. Weitere Informationen finden Sie im Abschnitt „Zusätzliche Konfigurationsoptionen für IBM MQ Bridge to Salesforce” auf Seite 930.
- Installieren Sie das **MQSeriesSFBridge** -Paket. Weitere Informationen finden Sie unter IBM MQ-Server unterLinux installieren und IBM MQ-RPM-Komponenten für Linux-Systeme.

## Informationen zu diesem Vorgang

Salesforce ist eine Cloud-basierte Customer-Relationship-Management-Plattform. Wenn Sie Salesforce für die Verwaltung von Kundendaten und Interaktionen einsetzen, können Sie IBM MQ Bridge to Salesforce verwenden, um Salesforce-Push-Themen und -Plattformereignisse zu abonnieren, die dann in Ihrem IBM MQ-Warteschlangenmanager veröffentlicht werden können. Anwendungen, die eine Verbindung zu diesem Warteschlangenmanager herstellen, können das Push-Thema und die Plattformereignisdaten in einer nützlichen Weise konsumieren. Sie können die Bridge auch verwenden, um Ereignisnachrichten für Plattformereignisse in Salesforce zu erstellen.

Eine Übersicht über die IBM MQ Bridge to Salesforce finden Sie in dem Diagramm in Abbildung 1.

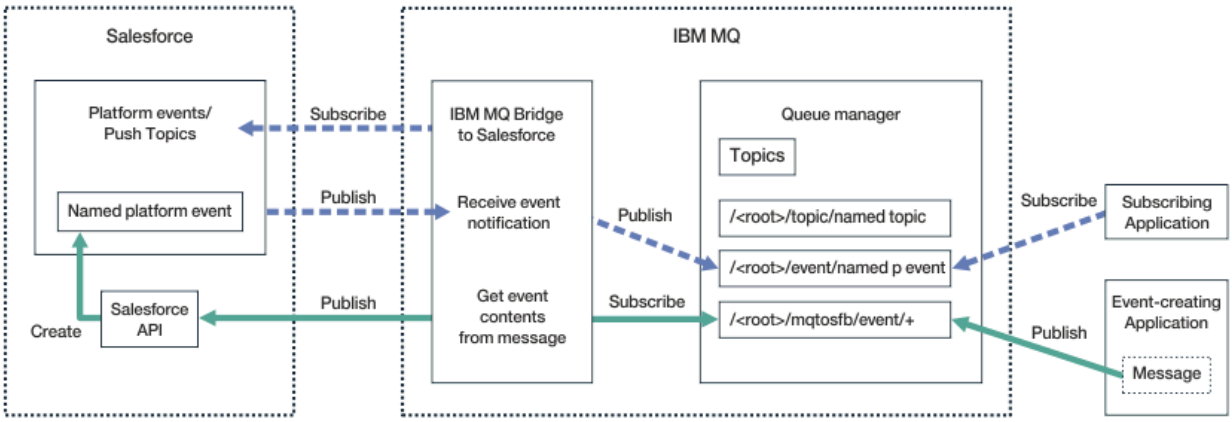



*Abbildung 97. IBM MQ Bridge to Salesforce*

Bei Push-Themen handelt es sich um Abfragen, die Sie für die Verwendung der Streaming-API von `Force.com` definieren, um Benachrichtigungen für Änderungen an Datensätzen in Salesforcezu empfangen. Weitere Informationen zum Konfigurieren von Push-Themen und zur Verwendung der Streaming-API finden Sie unter Introducing Streaming API und Working with PushTopics .

Plattformereignisse sind anpassbare Ereignisnachrichten, die definiert werden können, um die Ereignisdaten zu bestimmen, die von der `Force.com`-Plattform erzeugt oder konsumiert werden. Weitere Informationen zu Plattformereignissen und dem Unterschied zwischen Salesforce-Ereignissen finden Sie unter Enterprise messaging platform events und What is the difference between the Salesforce events.

- Informationen zum Erstellen der Konfiguration für das Abonnieren von Push-Themen und Plattformereignissen finden Sie in „IBM MQ Bridge to Salesforce konfigurieren” auf Seite 925.
- Informationen zum Erstellen der Konfiguration zum Erstellen von Ereignisnachrichten für Salesforce-Plattformereignisse finden Sie in „Ereignisnachrichten für Salesforce-Plattformereignisse erstellen” auf Seite 933.

Sie können die Daten von der Bridge auf zwei Arten überwachen, über die IBM MQ Console und mithilfe des Parameters **-p** im Befehl **amqsrua**. Eine Gruppe von Daten wird für den Gesamtüberbrückungsstatus veröffentlicht:

- Die Gesamtzahl der Push-Topic-Nachrichten, die in einem Intervall (unter dem `STATUS/PUSHTOPIC` -Baum) verarbeitet werden.
- Anzahl der in diesem Intervall sichtbaren Push-Themen.
- Gesamtzahl der Plattformereignisse, die in einem Intervall (unter dem `STATUS/PLATFORM`-Baum) verarbeitet werden.
- Die Anzahl der Plattformereignisse, die in diesem Intervall angezeigt werden.
- Gesamtzahl der von IBM MQ erstellten Plattformereignisse, die in einem Intervall verarbeitet werden (in der Struktur `STATUS/MQPE`).
- Eindeutige Anzahl der von IBM MQ erstellten Plattformereignisse, die in diesem Intervall angezeigt werden.

- Anzahl der fehlgeschlagenen Veröffentlichungen der von IBM MQ erstellten Plattformereignisse, die in diesem Intervall angezeigt werden.

Für jedes konfigurierte Salesforce-Topic wird eine weitere Nachricht veröffentlicht. Das IBM MQ-Thema verwendet den vollständigen Salesforce-Themennamen und den `/event` oder `/topic` im Objektnamen:

- Die Anzahl der Nachrichten, die in einem Intervall verarbeitet werden.

Informationen zum Konfigurieren der IBM MQ Console für die Überwachung von Bridgedaten finden Sie in den Schritten 9 und 10 im Abschnitt IBM MQ Bridge to Salesforce konfigurieren. Weitere Informationen zur Verwendung des Befehls **amqsrua** finden Sie im Abschnitt IBM MQ Bridge to Salesforce überwachen.

Führen Sie die Schritte in den folgenden Tasks aus, um die IBM MQ Bridge to Salesforce zu konfigurieren und auszuführen:

## Vorgehensweise

1. Konfigurieren Sie den IBM MQ Bridge to Salesforce.
2. Erstellen Sie Ereignisnachrichten für Salesforce-Plattformereignisse.
3. Führen Sie die IBM MQ Bridge to Salesforce aus.

**Zugehörige Tasks**

Tracefunktion für IBM MQ Bridge to Salesforce

**Zugehörige Verweise**

runmqsfb (IBM MQ Bridge to Salesforce ausführen)

# Linux Deprecated IBM MQ Bridge to Salesforce konfigurieren

Sie können IBM MQ konfigurieren und IBM MQ Bridge to Salesforce-Parameter eingeben, um die Konfigurationsdatei zu erstellen und Salesforce-Push-Themen und -Plattformereignisse mit Ihrem IBM MQ-Warteschlangenmanager zu verbinden.

## Vorbereitende Schritte

**Anmerkung:** Deprecated IBM MQ Bridge to Salesforce ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-431). Die Salesforce -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.

Bevor Sie mit dieser Task beginnen, müssen Sie sicherstellen, dass das `MQSeriesSFBridge`-Paket in Ihrer IBM MQ-Installation auf einer x86-64 Linux-Plattform installiert ist.

Weitere Informationen finden Sie unter IBM MQ-Server unterLinux installieren und IBM MQ-RPM-Komponenten für Linux-Systeme.

## Informationen zu diesem Vorgang

Diese Task führt Sie durch die minimale Konfiguration, die zum Erstellen der IBM MQ Bridge to Salesforce-Konfigurationsdatei erforderlich ist, und Sie können erfolgreich eine Verbindung zu Salesforce und IBM MQ herstellen, sodass Sie Salesforce-Push-Themen und -Plattformereignisse abonnieren können. Weitere Informationen über die Bedeutung und Optionen für alle Parameter finden Sie im Befehl runmqsfb . Sie müssen Ihre eigenen Sicherheitsanforderungen berücksichtigen und die Parameter anpassen, die für Ihre Implementierung geeignet sind.

Informationen zum Erstellen der Konfiguration zum Erstellen von Ereignisnachrichten für Salesforce-Plattformereignisse finden Sie in „Ereignisnachrichten für Salesforce-Plattformereignisse erstellen” auf Seite 933.

**Salesforce-Push-Themen und-Plattformereignisse abonnieren**

Wenn der IBM MQ Bridge to Salesforce Verbindungen zu Salesforce und IBM MQ herstellt, erstellt er Subskriptionen für Salesforce-Push-Themen und Plattformereignisse. Der Push-Topic-oder -Plattformereignisname, den die Bridge abonnieren möchte, muss in der Konfigurationsdatei enthalten sein oder in der Befehlszeile hinzugefügt werden, bevor die Verbindung hergestellt wird.

Eines der Konfigurationsattribute ist das Stammverzeichnis der IBM MQ-Themenstruktur, und die Ereignisse werden unterhalb dieses Stammverzeichnisses veröffentlicht. Die Bridge greift auf dieses Stammverzeichnis zu und fügt den vollständigen Salesforce-Themennamen hinzu, z. B. `/MQ/SF/ROOT/topic/EscalatedCases`. Das Monitoring-Thema und die Anwendungen, die eine Verbindung zu IBM MQ herstellen, können unter `/topic/EscalatedCases` und Plattformereignissen unter `/event/NewCustomer__e` nach Push-Themen suchen.

Die veröffentlichte Nachricht enthält Steuerinformationen und die Datenstruktur, die die angeforderten Datenfelder enthält. Bei Push-Themen ist die Datenstruktur ein **`sobject`** und für Plattformereignisse die Struktur **`payload`**. Die Bridge kann ein Thema oder ein Ereignis nicht subskribieren, wenn sie nicht in Salesforce definiert sind. Wenn die Brücke einen Fehler feststellt, wenn sie versucht, ein Thema zu subskribieren, wird die Brücke gestoppt.

Ein Themenobjekt muss in IBM MQ nicht definiert werden, aber es müssen geeignete Berechtigungen vorhanden sein, die auf dem nächstgelegenen übergeordneten Element in der Baumstruktur basieren. Die erneut veröffentlichte Nachricht enthält standardmäßig nur die relevante Datenstruktur aus der ursprünglichen Nachricht. Die Steuerinformationen werden entfernt. Bei Plattformereignissen weist die Veröffentlichung eine Nutzdatenstruktur auf. Die Konfigurationsoption **`Publish control data with the payload`** in der Gruppe **Verhalten des Bridge-Programms** aktiviert die erneute Veröffentlichung der gesamten Nachricht, einschließlich der Steuerdaten. Weitere Informationen hierzu finden Sie im Abschnitt Konfigurationsparameter .

Jedem Push-Topic und jedem Plattformereignis ist ein *ReplayID* in der Veröffentlichung von Salesforce zugeordnet. Der *ReplayID* kann zum Anfordern des Startpunkts für die Veröffentlichung verwendet werden, wenn die Verbindung zum Server hergestellt wird. Salesforce verwaltet eine Historie für bis zu 24 Stunden und ermöglicht es der Bridge, aktuelle Push-Themen und Plattformereignisse nicht zu übersehen, selbst wenn sie zu dem Zeitpunkt, zu dem sie generiert werden, nicht gestartet wurde. Die Brücke unterstützt zwei Servicemodi-Qualitäten:

**At-meist-einmal**
: Die Brücke verwendet den *ReplayId* nicht zum Neustart. Nach dem Neustart der Brücke werden nur neu generierte Push-Themen und Plattformereignisse verarbeitet. Anwendungen müssen darauf vorbereitet sein, fehlende Veröffentlichungen zu bearbeiten. Der *ReplayId* wird immer noch von der Bridge überwacht und in einer Warteschlange gespeichert, so dass die Brücke mit der anderen Servicequalität erneut gestartet werden kann und den aktuellen Status kennt.

**Wenigstens-einmal**
: Der *ReplayId* wird von der Bridge verfolgt und in eine Warteschlange verfestigt. Bei einem Neustart der Bridge wird der persistzierte *ReplayId* verwendet, um den Startpunkt für Veröffentlichungen vom Server anzufordern. Wenn die Lücke nicht mehr als 24 Stunden alt ist, werden ältere Veröffentlichungen gesendet. Die *ReplayId* für ein Thema wird nicht in jeder Nachricht verfestigt. Es wird in regelmäßigen Abständen in einer persistenten Nachricht geschrieben und wenn die Brücke heruntergefahren wird. Anwendungen müssen bereit sein, doppelte Veröffentlichungen anzuzeigen.

Der *ReplayId* wird als Nachricht in eine neu definierte Warteschlange geschrieben. Sie müssen diese Warteschlange **`SYSTEM.SALESFORCE.SYNCQ`** definieren, bevor die Bridge gestartet wird. Wenn der **`SYSTEM.SALESFORCE.SYNCQ`** nicht vorhanden ist, wird die Brücke nicht fortgesetzt, unabhängig von der Servicequalität. Es wird ein MQSC-Script bereitgestellt, um die Warteschlange mit relevanten Attributen zu erstellen. Die Warteschlange muss mit der Option `DEFSOPT (EXCL) NOSHARE` konfiguriert werden, um sicherzustellen, dass nur eine Instanz des Bridge-Programms die **`SYSTEM.SALESFORCE.SYNCQ`** -Warteschlange aktualisieren kann.

Informationen zum Erstellen der Konfiguration zum Erstellen von Ereignisnachrichten für Plattformereignisse finden Sie in „Ereignisnachrichten für Salesforce-Plattformereignisse erstellen” auf Seite 933.

## Vorgehensweise

1. Erstellen und starten Sie einen WS-Manager.
   a) Erstellen Sie einen WS-Manager, z. B. SQM1.

   ```
   crtmqm SQM1
   ```

   b) Starten Sie den WS-Manager.

   ```
   strmqm SQM1
   ```

2. **Anmerkung:** Wenn Sie die vorhandenen Berechtigungsnachweise für die Anmeldung und Sicherheit Salesforce und das selbst signierte Zertifikat verwenden möchten, fahren Sie mit Schritt „3” auf Seite 927 fort.

   Optional: Erstellen Sie ein Sicherheitstoken für Ihr Salesforce-Konto.
   a) Melden Sie sich an Ihrem Account von Salesforce an.
   b) Führen Sie die Schritte im Hilfeartikel Salesforce help: Reset your security token aus, um das Sicherheitstoken zu erstellen oder zurückzusetzen.
3. Erstellen Sie ein CA-signiertes Sicherheitszertifikat in Salesforce.
   a) Wählen Sie im Menü **Verwalten** Ihrer Seite **Force.com Home** die Option **Sicherheitskontrollen** und anschließend **Zertifikat-und Schlüsselverwaltung** aus.

      Die Seite **Zertifikat-und Schlüsselmanagement** wird geöffnet.
   b) Klicken Sie auf **CA-signiertes Zertifikat erstellen**.

      Die Seite **Certificates** wird geöffnet.
   c) Geben Sie im Feld **Bezeichnung** einen Namen für das Zertifikat ein, drücken Sie Tabund klicken Sie dann auf **Speichern**.

      Die Informationen zum Zertifikat und zu den Schlüsseldetails werden angezeigt.
   d) Klicken Sie auf **Back to list: Certificates and keys** .
   e) Klicken Sie auf **In Schlüsselspeicher exportieren** .
   f) Geben Sie ein Kennwort für den Schlüsselspeicher ein und klicken Sie dann auf **Exportieren** .
   g) Speichern Sie den exportierten Keystore in Ihrem lokalen Dateisystem.
4. Verwenden Sie die IBM Key Management-GUI, um den Keystore zu öffnen, den Sie aus Salesforce exportiert haben, und die Unterzeichnerzertifikate zu füllen.
   a) Führen Sie den Befehl **strmqikm** aus, um die grafische Benutzerschnittstelle von IBM Key Management zu öffnen.

      Weitere Informationen finden Sie unter Using runmqckm, runmqakm und strmqikm to manage digital certificates .
   b) Klicken Sie auf **Open a key database file** (Schlüsseldatenbankdatei öffnen) und navigieren Sie zum Speicherort des Salesforce-Keystore.
   c) Klicken Sie auf **Öffnen** , stellen Sie sicher, dass Sie **JKS** aus den Optionen für **Schlüsseldatenbanktyp** auswählen, und klicken Sie dann auf **OK** .
   d) Geben Sie das Kennwort ein, das Sie für den Keystore in Schritt 3f erstellt haben, und klicken Sie dann auf **OK** .
   e) Wählen Sie in den Optionen für **Key database content** die Option **Signer Certificates** aus.
   f) Klicken Sie auf **Populate** .
   g) Wählen Sie das Kontrollkästchen **Verisign Inc.** in der Liste **Add CA Certificates** aus und klicken Sie dann auf **OK** .
5. Optional: Generieren Sie einen OAuth-Konsumentenschlüssel und einen geheimen OAuth-Konsumentenschlüssel. Erstellen Sie hierzu eine Anwendungsverbindung für die IBM MQ Bridge to Salesforce in Ihrem Salesforce-Konto.

Die Codes des Konsumentenschlüssels (**Consumer Key**) und des geheimen Konsumentenschlüssels (**Consumer Secret**) benötigen Sie, wenn Sie die IBM MQ Bridge to Salesforce in Produktionsumgebungen einsetzen.

a) Wählen Sie **Erstellen** und dann **Apps** im Menü **Build** Ihrer **Force.com Home** -Seite aus.

   Die Seite 'Apps' wird geöffnet.

b) Klicken Sie im Abschnitt **Connected Apps** auf **Neu** .

   Die Seite **Neue verbundene Anwendung** wird geöffnet.

c) Geben Sie im Feld **Name der verbundenen App**einen Namen für IBM MQ Bridge to Salesforce ein, z. B. **MQBridgeToSalesforce**.

d) Geben Sie den **API-Namen** ein.

   Wenn Sie die Registerkarte mit dem nächsten Feld durchlaufen, wird der **Name der verbundenen App** in das Namensfeld **API Name** kopiert.

e) Geben Sie Ihre **Kontaktmail-Adresse** ein.

f) Wählen Sie die Option **OAuth-Einstellungen aktivieren** im Abschnitt **API (Enable OAuth Settings)** aus.

   Weitere Optionen in diesem Abschnitt werden dann angezeigt.

g) Fügen Sie Ihre **Callback URL**hinzu, z. B. `https://www.ibm.com`.

h) Wählen Sie die Option **Vollständiger Zugriff (full)** in der Liste **Verfügbare OAuth-Scopes** im Unterabschnitt **Ausgewählte OAuth-Scopes** aus und klicken Sie dann auf **Hinzufügen** , um vollständigen Zugriff auf die Liste **Ausgewählte OAuth-Scopes** hinzuzufügen.

i) Klicken Sie auf **Speichern**.

j) Klicken Sie auf **Weiter** .

k) Notieren Sie die Codes **Consumer Key** (Consumer Key) und **Consumer Secret** (Consumer Secret).

6. Erstellen Sie die erforderliche Synchronisationswarteschlange auf dem WS-Manager.

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc SQM1
```

   Die Synchronisationswarteschlange verwaltet den Ereignisstatus in der gesamten Anwendungs-oder WS-Manager-Neustarts. Die Warteschlangenlänge kann klein sein, da in der Warteschlange nur eine einzige Nachricht erwartet wird. Es kann nur eine Instanz der Bridge zu einem Zeitpunkt für diese Warteschlange ausgeführt werden. Daher werden die Standardoptionen für exklusiven Zugriff festgelegt.

7. Erstellen Sie eine Konfigurationsdatei mit den Verbindungs- und Sicherheitsparametern für IBM MQ, Salesforce und das Verhalten der IBM MQ Bridge to Salesforce.

```
runmqsfb -o new_config.cfg
```

   Die vorhandenen Werte werden in den eckigen Klammern angezeigt. Drücken Sie `Enter` , um vorhandene Werte zu akzeptieren, drücken Sie `Space` dann `Enter` , um Werte zu löschen, und geben Sie `Enter` ein, um neue Werte hinzuzufügen.

   a) Geben Sie Werte für die Verbindung zum WS-Manager SQM1 ein:

      Als Minimalkonfiguration für die Verbindung werden der Warteschlangenmanagername, der Basisthemen-Root von IBM MQ und der Kanalname benötigt.

```
Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : []SQM1
MQ Base Topic              : []/sf
MQ Channel                 : []A channel you have defined or for example SYS
TEM.DEF.SVRCONN
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
```

```
MQ Userid                  : []
MQ Password                : []
```

**Anmerkung:** Der Kanalname ist nicht erforderlich, wenn Sie lokal eine Verbindung herstellen. Sie müssen den Namen und das Basisthema des Warteschlangenmanagers nicht in der Konfigurationsdatei angeben, da sie später in die Befehlszeile aufgenommen werden können, wenn Sie die Brücke ausführen.

b) Geben Sie die Werte für die Verbindung mit Salesforce ein:

Die Mindestwerte, die für die Verbindung benötigt werden, sind Salesforce-Benutzer-ID, Kennwort, Sicherheitstoken und Anmeldeendpunkt. In Produktionsumgebungen können Sie den Consumer-Schlüssel und den geheimen Schlüssel für die OAuth-Sicherheit hinzufügen.

```
Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)      : []Security_Token
Login Endpoint             : [https://login.salesforce.com]
Consumer ID                : []
Consumer Secret Key        : []
```

c) Geben Sie Werte für Zertifikatsspeicher für TLS-Verbindungen ein:

Mindestwerte, die für TLS-Verbindungen benötigt werden, sind der Pfad zum Schlüsselspeicher für TLS-Zertifikate und das Schlüsselspeicherkennwort. Wenn kein vertrauenswürdiger Speicherpfad oder kein Kennwort angegeben ist, werden die Parameter für den Schlüsselspeicher und das Kennwort für den vertrauenswürdigen Speicher und das Kennwort verwendet. Wenn Sie TLS für Ihre IBM MQ-Warteschlangenmanagerverbindung verwenden, können Sie denselben Schlüsselspeicher verwenden.

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
SQM1/ssl/key.jks
Keystore password          : []keystore_password
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]
```

d) Geben Sie die Werte für die Konfiguration des Verhaltens der IBM MQ Bridge to Salesforce ein:

Sie müssen keine dieser Werte ändern oder angeben, aber wenn Sie Ihre Push-Topic-oder Plattformereignisnamen kennen, fügen Sie sie hier hinzu. Sie können auch zu einem späteren Zeitpunkt in der Befehlszeile hinzugefügt werden, wenn Sie bereit sind, die Brücke zu starten. Sie müssen die Protokolldatei, in der Konfigurationsdatei oder in der Befehlszeile angeben.

```
Behaviour of bridge program
---------------------------
PushTopic Names          : []
Platform Event Names     : []
MQ Monitoring Frequency  : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [N]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

8. Optional: Erstellen Sie den IBM MQ-Service zur Steuerung der Programmausführung. Bearbeiten Sie die `mqsfbService.mqsc`-Beispieldatei so, dass sie auf die neu erstellte Konfigurationsdatei verweist, und nehmen Sie alle anderen Änderungen an den Befehlsparametern vor.

```
cat modified mqsfbService.mqsc | runmqsc SQM1
```

9. Optional: Befolgen Sie die Anweisungen im Abschnitt Einführung in die IBM MQ Console, um die IBM MQ Console einzurichten.

10. Optional: Konfigurieren Sie IBM MQ Bridge to Salesforce für die Ausführung als Rootbenutzer.

Damit IBM MQ Bridge to Salesforce als *rootless user*ausgeführt werden kann, z. B. in einem *rootless Container*, müssen die Verzeichnisse Java `userRoot` und `systemRoot` ordnungsgemäß festgelegt sein, damit der Benutzer, der den Bridge-Prozess ausführt, Schreib-/Lesezugriff hat. Legen Sie dazu die folgenden JVM-Eigenschaften fest:

```
export MQSFB_EXTRA_JAVA_OPTIONS="-Djava.util.prefs.userRoot=directory_with_read_write_ac
cess"
```

```
export MQSFB_EXTRA_JAVA_OPTIONS="-Djava.util.prefs.systemRoot=directory_with_read_write_ac
cess"
```

### Ergebnisse

Sie haben die Konfigurationsdatei erstellt, die die IBM MQ Bridge to Salesforce zur Subskription von Salesforce-Push-Themen und -Plattformereignissen sowie zur Veröffentlichung derselben in Ihrem IBM MQ-Netz verwendet.

### Nächste Schritte

Führen Sie die in „IBM MQ Bridge to Salesforce ausführen” auf Seite 939 beschriebenen Schritte aus.

**Zugehörige Tasks**

Tracefunktion für IBM MQ Bridge to Salesforce

IBM MQ Bridge to Salesforce überwachen

**Zugehörige Verweise**

runmqsfb (IBM MQ Bridge to Salesforce ausführen)

## Linux Deprecated Zusätzliche Konfigurationsoptionen für IBM MQ Bridge to Salesforce

Ab IBM MQ 9.2.0 sind zusätzliche Konfigurationsoptionen verfügbar, mit denen zwei übergeordnete Klassen mit einer zusätzlichen Topologie ermöglicht werden, in denen "eingehende" Ereignisse (die von Salesforce generiert und in IBM MQ-Anwendungen veröffentlicht werden) und "ausgehende" Ereignisse (IBM MQ-Anwendungen, die Ereignisse veröffentlichen, welche an Salesforce gesendet werden) verarbeitet werden. Außerdem gibt es eine Änderung bei der Funktionsweise von Tracing und Protokollierung.

### Änderungen ab IBM MQ 9.1.0 IBM MQ Bridge to Salesforce

Mit Ausnahme des Rotierens von Protokolldateien gibt es ab IBM MQ 9.2.0 keine Änderungen am Verhalten der IBM MQ 9.1.0-Bridge. Weitere Informationen finden Sie unter „Protokollrotation” auf Seite 931.

Die wesentlichste Änderung besteht darin, dass ein Warteschlangenmanager mehrere Bridge-Instanzen unterstützt Zum Aktivieren dieser Funktion und der restlichen Komponenten der zusätzlichen Topologien müssen Sie einige manuelle Konfigurationsänderungen vornehmen.

Weitere Inforationen zu den zusätzlichen Konfigurationsoptionen finden Sie unter runmqfsb und ein Beispiel der überarbeiteten Konfigurationsinformationen finden Sie im Abschnitt „Beispielkonfigurationsausgabe für IBM MQ Bridge to Salesforce” auf Seite 932.

### Separate eingehende Ereignisse

Eingehende Ereignisse von Salesforce für IBM MQ werden von mehreren Instanzen der Bridge verarbeitet, aber diese Instanzen müssen in unabhängigen Gruppen aus Push-Themen und Push-Ereignissen für Salesforce ausgeführt werden. Andernfalls könnten diese Ereignisse von IBM MQ-Anwendungen als wiederholte Ereignisse angezeigt werden, da kein Protokoll für mehrere Bridges vorhanden ist, mit dem das Duplizieren von Ereignissen gestoppt wird. Jede Instanz verwendet eine eigene konfigurierbare Synchronisationswarteschlange, in der **ReplayId** gespeichert wird.

Diese Vorgehensweise ist voraussichtlich in den folgenden Situationen angemessen:

- Verschiedene Salesforce-Themen haben unterschiedliche Sicherheitsberechtigungen. Jede Bridge-Instanz hat eine andere Gruppe von Berechtigungsnachweisen für den Zugriff auf Salesforce.
- Sie haben Bedenken, dass der Workload von Salesforce zu groß ist, um von einer einzelnen Bridge verarbeitet zu werden. Daher können Sie eine Aufteilung einrichten, in der die Themen mit "A-M" eine Bridge und "N-Z" eine andere Bridge durchlaufen.

## Gemeinsam genutzte abgehende Ereignisse

Die Bridge unterstützt mehrere Instanzen, damit das Senden von abgehenden Ereignisse von IBM MQ an Salesforce unterstützt wird. Wenn eine Instanz der Bridge fehlschlägt, können andere Instanzen, die die gleichen Themen auf dem gleichen Warteschlangenmanager subskribiert haben, die Verarbeitung der Veröffentlichungen fortsetzen.

**Anmerkung:** Dazu sind keine Änderungen an der Konfiguration von IBM MQ-Themen erforderlich.

Diese kooperierenden Instanzen müssen so eingerichtet sein, dass mindestens eine der Instanzen eingehende Ereignisse von Salesforce verarbeitet, da diese Instanz über exklusiven Zugriff auf die Synchronisationswarteschlange verfügen muss.

Diese Vorgehensweise ist voraussichtlich bei Bedenken hinsichtlich der folgenden Komponenten angemessen:

- Workload von IBM MQ. Da die Anforderungen an Salesforce synchron sind, kann die Bridge keine neue Arbeit behandeln, während sie noch eine andere Nachricht verarbeitet. Das Verwenden mehrerer Konsumenten erleichtert diese Situation.
- Architektur der Verfügbarkeit. Es ist jetzt beispielsweise möglich, mehrere Instanzen in separaten Rechenzentren mit verbesserten Optionen für Ausfälle und Disaster-Recovery auszuführen. Bei der Ausführung als IBM MQ-Client wird die Bridge ebenfalls von der Position des Warteschlangenmanagers getrennt.

## Interaktion von Trace und Debug

Ab IBM MQ 9.2.0 wird das Debug-Flag wie zuvor für IBM MQ 9.1.0 ausgeführt. Das bedeutet, dass *-d1* Debuginformationen übermittelt und *-d2* die Debugprotokollierung für die vorausgesetzten Komponenten aktiviert. Wenn Sie allerdings den IBM MQ-Trace beim Start der Bridge aktiviert haben, ist die Berichterstellung auf Ebene *-d2* automatisch aktiviert.

## Protokollrotation

Ab IBM MQ 9.2.0 werden als Standardverhalten für die Protokolldatei drei Protokolldateien mit einer Größe von jeweils 2 MB verwendet. Sie können diese Werte mithilfe zusätzlicher Konfigurationseigenschaften überschreiben. Das vorhandene Konfigurationsattribut oder der Befehlszeilenparameter für die Protokolldatei wird als Basisname für die Protokolle verwendet, wobei noch ein Index hinzugefügt wird.

Wenn die konfigurierte Protokolldatei

- keinen Dateityp hat, wird der Index am Ende des Dateinamens hinzugefügt.

  Wenn die Protokolldatei auf `abc` gesetzt ist, werden die Protokolle `abc.0`, `abc.1` usw. erzeugt.
- einen Dateityp hat, wird der Index vor dem Dateityp hinzugefügt.

  Wenn die Protokolldatei auf `abc.log` gesetzt ist, werden die Protokolle `abc.0.log`, `abc.1.log` usw. erzeugt.

**Anmerkungen:**

1. Da die Bridges mit einer beliebigen Benutzerberechtigung ausgeführt werden können, ist es nicht möglich, für die Protokolle ein bestimmtes Verzeichnis, z. B. `/var/mqm/qmgrs/<qm>/errors`, zu erzwingen.
2. Die gleichen Informationen werden weiterhin in die Datenströme *stdout* und *stderr* geschrieben.

3. Sobald eine einzelne Protokolldatei erneut geöffnet wird, werden die Informationen zu Basiskonfiguration erneut gedruckt. Die Informationen sind immer verfügbar, statt nur ein einziges Mal beim Start des Programms gedruckt zu werden.

### Protokolle beibehalten

Durch die IBM MQ 9.2.0-Topologien ist es wahrscheinlicher, dass mehrere Instanzen der Bridge auf einem bestimmten Warteschlangenmanager ausgeführt werden.

Um zu vermeiden, dass die Instanzen sich gegenseitig beeinträchtigen, und um das Überschreiben vorheriger Ausführungen der Bridge zu verhindern, wird die Bridge nicht gestartet, wenn das Protokoll `.0` bereits vorhanden ist.

Sie benötigen eine Startprozedur, mit der vorherige Kopien des Protokolls vor dem Start der Bridge gelöscht werden oder mit der dem Namen beispielsweise eine Zeitmarke hinzugefügt wird.

**Zugehörige Tasks**

„IBM MQ für die Verwendung mit Push-Themen und Plattformereignissen für Salesforce konfigurieren" auf Seite 923

Verwenden Sie diese Informationen, um die Sicherheit und die Verbindungen zu Salesforce und Ihrem IBM MQ-Netz zu konfigurieren, indem Sie die IBM MQ Bridge to Salesforce konfigurieren und anschließend ausführen.

Tracefunktion für IBM MQ-Bridge für Salesforce

**Zugehörige Verweise**

runmqfsb

## Linux Deprecated Beispielkonfigurationsausgabe für IBM MQ Bridge to Salesforce

Ausgabe einer Beispielkonfiguration, in der die Änderungen von IBM MQ 9.1.0 IBM MQ Bridge to Salesforce gezeigt werden.

```
IBM MQ Bridge to Salesforce
5724-H72 (C) Copyright IBM Corp. 2017, 2024.
Level : <<unknown>>

Enter new values for the configuration attributes. The
current settings are shown.
Press ENTER to accept current values; use SPACE+ENTER
to clear values.

Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : [V9000_A]
MQ Base Topic              : [/sf]
MQ Channel                 : []
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.DEADQ]
MQ Replay Status Queue     : [SYSTEM.SALESFORCE.SYNCQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []

Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : [johndoe@<yourenterprise>.com]
Salesforce Password (reqd) : [********]
Security Token             : [********]
Login Endpoint             : [https://login.salesforce.com]
Consumer Key               : [3MVG9HxRZv05HarQhSy89qSKYNr1gDcv1wE3zN5kyFAa4Wxt]
Consumer Secret            : [********]

Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : [/var/mqm/ssl/key.jks]
Keystore password          : [********]
Trusted store for signer certificates : []
```

```
Trusted store password     : []
Use TLS for MQ connection  : [N]

Event processing
----------------
PushTopic Names            : []
Platform Event Names       : []
At-least-once delivery for Salesforce events? (Y/N) : [N]
At-least-once delivery for MQ publications? (Y/N) : [N]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [Y]
Treat unknown Salesforce topic as warning (Y/N) : [N]

Behaviour of bridge program
---------------------------
Bridge unique identifier   : []
MQ Monitoring Frequency    : [30]
Delay before starting to process events : [0]
Continue to retry after maximum reconnection attempts (Y/N) : [N]
Runtime logfile for copy of stdout/stderr : [/tmp/runmqsfb.log]
Number of logfiles         : [3]
Maximum size of each logfile : [2097152]
Done.
```

**Zugehörige Verweise**

runmqfsb

# Linux Deprecated Ereignisnachrichten für Salesforce-Plattformereignisse erstellen

Sie können IBM MQ konfigurieren und `IBM MQ Bridge to Salesforce parameters` eingeben, um die Konfigurationsdatei zu erstellen und über die Bridge Ereignisnachrichten für Salesforce -Plattformereignisse zu erstellen.

## Vorbereitende Schritte

- Sie haben das Paket **MQSeriesSFBridge** in Ihrer IBM MQ-Installation auf einer x86-64 Linux-Plattform installiert.

## Informationen zu diesem Vorgang

Diese Task führt Sie durch die minimale Konfiguration, die zum Erstellen der IBM MQ Bridge to Salesforce-Konfigurationsdatei erforderlich ist, und Sie können erfolgreich eine Verbindung zu Salesforce und IBM MQ herstellen, sodass Sie Ereignisnachrichten für Salesforce-Plattformereignisse erstellen können. Weitere Informationen über die Bedeutung und Optionen für alle Parameter finden Sie im Befehl runmqsfb . Sie müssen Ihre eigenen Sicherheitsanforderungen berücksichtigen und die Parameter anpassen, die für Ihre Implementierung geeignet sind.

Informationen zum Erstellen der Konfiguration für das Abonnieren von Push-Themen und Plattformereignissen finden Sie in „IBM MQ Bridge to Salesforce konfigurieren” auf Seite 925.

**Ereignisnachrichten für Salesforce-Plattformereignisse erstellen**

Sie können eine IBM MQ-Anwendung verwenden, um Nachrichten zu erstellen, die auf einem Warteschlangenmanager-Artikel `/root/mqtosfb/event/+` eingereiht werden. Die Bridge subskribiert das Thema, ruft Inhalt aus den Nachrichten ab und verwendet diesen zum Veröffentlichen von Ereignisnachrichten für ein Salesforce-Plattformereignis. Weitere Informationen zu Plattformereignissen finden Sie im Abschnitt Angepasste Benachrichtigungen mit Plattformereignissen bereitstellen in der Salesforce-Entwicklerdokumentation.

Um die Bridge für das Erstellen von Ereignisnachrichten zu aktivieren, müssen Sie zwei Attribute bereitstellen, die zusätzlich zu den Attributen für die Subskribierung von Push-Themen und Plattformereignissen verwendet werden:

- Erstellen Sie den Namen der **`MQ Publication Error Queue`** in den Brückenkonfigurationsattributen für **Verbindung zum Warteschlangenmanager** und fügen Sie ihn hinzu.

- Setzen Sie die Option **Subscribe to MQ publications for platform events** in den Brückenkonfigurationsattributen für die Definition von **Verhalten des Brückenprogramms** auf *J*.

Sie müssen ein Plattformereignis in Salesforce erstellen und die Inhaltsfelder definieren, bevor Sie die Bridge verwenden können, um Ereignisnachrichten für dieses Plattformereignis zu erstellen. Der Plattformereignisname und sein Inhalt bestimmen, wie Sie die IBM MQ-Nachricht formatieren müssen, die von der Bridge verarbeitet wird. Wenn Ihr Salesforce Plattformereignis **Object name** beispielsweise *MQPlatformEvent1* ist und Ihre beiden benutzerdefinierten Felder Textfelder mit den **API name** *MeinText__c* und *Name__c* sind, muss Ihre IBM MQ-Nachricht, die im Thema `/root/mqtosfb/event/MQPlatformEvent1__e` veröffentlicht wird, wie folgt ein ordnungsgemäß formatiertes JSON sein:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

Die Nachricht muss so formatiert sein, dass sie von IBM MQ Bridge to Salesforce als Nachrichtenhauptteil im Format MQFMT_STRING erkannt werden kann.

Lesen Sie Schritt „7” auf Seite 936, um Ihr Plattformereignis in Salesforce zu erstellen, oder überspringen Sie diesen Schritt, wenn Sie bereits über ein Plattformereignis verfügen, für das Ereignisnachrichten erstellt werden sollen. Sie müssen Ihre IBM MQ-Nachricht so formatieren, dass sie mit den Feldern übereinstimmt, die in Ihrem Salesforce-Plattformereignis festgelegt sind. Felder innerhalb des Salesforce-Plattformereignisses können als optional oder obligatorisch festgelegt werden. Weitere Informationen finden Sie im Abschnitt Plattformereignisfelder in der Salesforce-Entwicklerdokumentation.

Wenn die Bridge aktiv ist, subskribiert sie das angegebene IBM MQ-Thema.

- Wenn Sie die Servicequalität **At-most-once** in der Brückenkonfiguration angeben, ist die Subskription, die die Brücke erstellt, nicht permanent. Alle Veröffentlichungen, die von IBM MQ-Anwendungen ausgeführt werden, während die Brücke nicht aktiv ist, werden nicht verarbeitet.
- Wenn Sie die Servicequalität **At-least-once** in der Brückenkonfiguration angeben, ist die Subskription, die die Brücke herstellt, permanent. Dies bedeutet, dass die Brücke Veröffentlichungen verarbeiten kann, die von IBM MQ-Anwendungen ausgeführt werden, während die Brücke nicht aktiv ist. Dauerhafte Subskriptionen erfordern eine bekannte Subskription und eine bekannte Client-ID. Die Brücke verwendet *D_SUB_RUNMQSFB* als Subskriptionsnamen und *runmqsfb_1* als Client-ID.

Wenn die Brücke für die Subskription von Salesforce-Push-Themen und -Plattformereignissen und nicht zum Erstellen von Ereignisnachrichten verwendet wird, versucht sie, die permanente Subskription zu löschen, wenn die Konfiguration geändert wird, und die Subskription jetzt verwaist ist.

Sie können permanente Subskriptionen, die die Brücke erstellt, wie folgt entfernen:

**Verwenden Sie die IBM MQ Explorer.**
Öffnen Sie den **Subskriptionsordner** für den Warteschlangenmanager, den die Bridge verwendet, und suchen Sie nach dem Subskriptionsnamen, der in *:D_SUB_RUNMQSFB* endet, in dem die Themenzeichenfolge `/sf/mqtosfb/event+` ist. Klicken Sie auf den Subskriptionsnamen, und klicken Sie auf Löschen. Wenn Sie einen Fehler erhalten, der angibt, dass die Subskription im Gebrauch ist, ist Ihre Brücke möglicherweise noch aktiv. Stoppen Sie die Brücke, und versuchen Sie erneut, die Subskription zu löschen.

**Verwenden Sie `runmqsc` , um die Subskription zu suchen und zu löschen.**
Starten Sie die Schnittstelle **`runmqsc`** und führen Sie `DISPLAY SUB (*)` aus. Suchen Sie nach dem Subskriptionsnamen **SUB**, der auf *:D_SUB_RUNMQSFB* endet. Geben Sie den Unterbefehl "delete" aus und schließen Sie die **SUBID** der Subskription ein, die Sie löschen möchten. Beispiel:`DELETE SUB SUBID(414D5120514D312020202020202020205C589459987E8620)`

**Stoppen Sie die Bridge und starten Sie sie anschließend mit der Servicequalität `At-most-once` .**
Wenn Sie die Bridge mit der **At-least-once** Servicequalität `At-least-once delivery? (Y/N) :[Y]` gestartet haben, ist die erstellte Subskription permanent. Ändern Sie zum Löschen der Subskription die Servicequalität in Ihrer Konfigurationsdatei in `At-least-once delivery? (Y/N) :[N]` und starten Sie die Bridge erneut. Die permanente Subskription wird gelöscht, und es wird eine nicht permanente Subskription erstellt.

## Vorgehensweise

1. Erstellen und starten Sie einen WS-Manager.
   a) Erstellen Sie einen WS-Manager, z. B. PEQM1.

   ```
   crtmqm PEQM1
   ```

   b) Starten Sie den WS-Manager.

   ```
   strmqm PEQM1
   ```

2. **Anmerkung:** Wenn Sie für die Anmeldung und die Sicherheit bestehende Salesforce-Berechtigungsnachweise oder ein bestehendes selbst signiertes Zertifikat verwenden möchten, gehen Sie zu Schritt 4.

   Optional: Erstellen Sie ein Sicherheitstoken für Ihr Salesforce-Konto.
   a) Melden Sie sich an Ihrem Account von Salesforce an.
   b) Führen Sie die Schritte im Hilfeartikel Salesforce help: Reset your security token aus, um das Sicherheitstoken zu erstellen oder zurückzusetzen.
3. Erstellen Sie in Salesforce ein selbst signiertes Sicherheitszertifikat.
   a) Wählen Sie im Menü **Verwalten** Ihrer Seite **Force.com Home** die Option **Sicherheitskontrollen** und anschließend **Zertifikat-und Schlüsselverwaltung** aus.

      Die Seite **Zertifikat-und Schlüsselmanagement** wird geöffnet.
   b) Klicken Sie auf **Create Self-Signed certificate** (Selbstsigniertes Zertifikat

      Die Seite **Certificates** wird geöffnet.
   c) Geben Sie im Feld **Bezeichnung** einen Namen für das Zertifikat ein, drücken Sie Tabund klicken Sie dann auf **Speichern**.

      Die Informationen zum Zertifikat und zu den Schlüsseldetails werden angezeigt.
   d) Klicken Sie auf **Back to list: Certificates and keys** .
   e) Klicken Sie auf **In Schlüsselspeicher exportieren** .
   f) Geben Sie ein Kennwort für den Schlüsselspeicher ein und klicken Sie dann auf **Exportieren** .
   g) Speichern Sie den exportierten Keystore in Ihrem lokalen Dateisystem.
4. Verwenden Sie die IBM Key Management-GUI, um den Keystore zu öffnen, den Sie aus Salesforce exportiert haben, und die Unterzeichnerzertifikate zu füllen.
   a) Führen Sie den Befehl **strmqikm** aus, um die grafische Benutzerschnittstelle von IBM Key Management zu öffnen. Weitere Informationen finden Sie unter Using runmqckm, runmqakm und strmqikm to manage digital certificates .
   b) Klicken Sie auf **Open a key database file** (Schlüsseldatenbankdatei öffnen) und navigieren Sie zum Speicherort des Salesforce-Keystore.
   c) Klicken Sie auf **Öffnen** , stellen Sie sicher, dass Sie **JKS** aus den Optionen für **Schlüsseldatenbanktyp** auswählen, und klicken Sie dann auf **OK** .
   d) Geben Sie das Kennwort ein, das Sie für den Keystore in Schritt 3f erstellt haben, und klicken Sie dann auf **OK** .
   e) Wählen Sie in den Optionen für **Key database content** die Option **Signer Certificates** aus.
   f) Klicken Sie auf **Populate** .
   g) Wählen Sie das Kontrollkästchen **Verisign Inc.** in der Liste **Add CA Certificates** aus und klicken Sie dann auf **OK** .
5. Optional: Generieren Sie einen OAuth-Konsumentenschlüssel und einen geheimen OAuth-Konsumentenschlüssel. Erstellen Sie hierzu eine Anwendungsverbindung für die IBM MQ Bridge to Salesforce in Ihrem Salesforce-Konto.

Die Codes des Konsumentenschlüssels (**Consumer Key**) und des geheimen Konsumentenschlüssels (**Consumer Secret**) benötigen Sie, wenn Sie die IBM MQ Bridge to Salesforce in Produktionsumgebungen einsetzen.

a) Wählen Sie **Erstellen** und dann **Apps** im Menü **Build** Ihrer **Force.com Home** -Seite aus.

   Die Seite **Apps** wird geöffnet.

b) Klicken Sie im Abschnitt **Connected Apps** auf **Neu** .

   Die Seite **Neue verbundene Anwendung** wird geöffnet.

c) Geben Sie im Feld **Name der verbundenen App**einen Namen für IBM MQ Bridge to Salesforce ein, z. B. **MQBridgeToSalesforce**.

d) Geben Sie den **API-Namen** ein.

   Wenn Sie die Registerkarte mit dem nächsten Feld durchlaufen, wird der **Name der verbundenen App** in das Namensfeld **API Name** kopiert.

e) Geben Sie Ihre **Kontaktmail-Adresse** ein.

f) Wählen Sie die Option **OAuth-Einstellungen aktivieren** im Abschnitt **API (Enable OAuth Settings)** aus.

   Weitere Optionen in diesem Abschnitt werden dann angezeigt.

g) Fügen Sie Ihre **Callback URL**hinzu, z. B. `https://www.ibm.com`.

h) Wählen Sie die Option **Vollständiger Zugriff (full)** in der Liste **Verfügbare OAuth-Scopes** im Unterabschnitt **Ausgewählte OAuth-Scopes** aus und klicken Sie dann auf **Hinzufügen** , um vollständigen Zugriff auf die Liste **Ausgewählte OAuth-Scopes** hinzuzufügen.

i) Klicken Sie auf **Speichern**.

j) Klicken Sie auf **Weiter** .

k) Notieren Sie die Codes **Consumer Key** (Consumer Key) und **Consumer Secret** (Consumer Secret).

6. Erstellen Sie die erforderlichen Synchronisations- und Fehlerwarteschlangen auf dem Warteschlangenmanager.

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc PEQM1
```

   Die Synchronisationswarteschlange verwaltet den Ereignisstatus in der gesamten Anwendungs-oder WS-Manager-Neustarts. Die Warteschlangenlänge kann klein sein, da in der Warteschlange nur eine einzige Nachricht erwartet wird. Es kann nur eine Instanz der Bridge zu einem Zeitpunkt für diese Warteschlange ausgeführt werden. Daher werden die Standardoptionen für exklusiven Zugriff festgelegt. Die Fehlerwarteschlange muss erstellt werden, bevor Sie die Brücke verwenden können, um Ereignisnachrichten für Plattformereignisse zu erstellen. Die Fehlerwarteschlange wird für Nachrichten verwendet, die nicht erfolgreich von Salesforce verarbeitet werden können. Sie müssen den Namen der Fehlerwarteschlange im Abschnitt **`Connection to Queue Manager`** des Bridge-Konfigurationsparameters hinzufügen (siehe Schritt „8.a" auf Seite 937).

7. Optional: Erstellen Sie ein Plattformereignisobjekt in Ihrem Salesforce-Account.

a) Wählen Sie **Plattformereignisse** im Menü **Develop** Ihrer **Force.com Home** -Seite aus und klicken Sie dann auf **Neues Plattformereignis** .

   Die Seite **Neues Plattformereignis** wird geöffnet.

b) Füllen Sie die Felder **Bezeichnung** und **Plural Label** aus.

c) Klicken Sie auf **Speichern** .

   Die Seite **Details der Plattformereignisdefinition** wird geöffnet.

d) Definieren Sie den **Angepassten Feld-&Beziehungen**.

   Sie können z. B. zwei Textfelder mit den Bezeichnungen *MyText* und *Name* hinzufügen und die Feldlängen für **Datentyp** auf *Text(64)* bzw. *Text(32)* setzen.

Sie haben ein Plattformereignis erstellt und **`Custom Fields and Relationships`** dafür definiert. Verwenden Sie Ihr Plattformereignis *Platform Object name* oder *API name* als IBM MQ-Thema, in das Sie Nachrichten einfügen können, die von der Brücke verarbeitet werden sollen. Sie können

beispielsweise das Beispiel **AMQSPUBA** verwenden, um die folgende Nachricht im JSON-Format zum Thema /sf/mqtosfb/event/*Salesforce Platform Object Name/API name* hinzuzufügen:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

Sie können das Beispiel **AMQSPUBA** ausführen, um Nachrichten zu erstellen, nachdem die Bridge gestartet wurde. Geben Sie den folgenden Befehl aus dem Verzeichnis *MQ installation location*/samp/bin aus:

```
./amqspub /sf/mqtosfb/event/Salesforce Platform Object Name/API name PEQM1
```

Geben Sie an der Eingabeaufforderung die Nachricht im JSON-Format ein.

8. Erstellen Sie eine Konfigurationsdatei mit den Verbindungs- und Sicherheitsparametern für IBM MQ, Salesforce und das Verhalten der IBM MQ Bridge to Salesforce.

```
runmqsfb -o new_config.cfg
```

Die vorhandenen Werte werden in den eckigen Klammern angezeigt. Drücken Sie Enter , um vorhandene Werte zu akzeptieren, drücken Sie Space dann Enter , um Werte zu löschen, und geben Sie Enter ein, um neue Werte hinzuzufügen.

a) Geben Sie Werte für die Verbindung zum WS-Manager PEQM1 ein:

Als Minimalkonfiguration für die Verbindung werden der Warteschlangenmanagername, der Basisthemen-Root von IBM MQ, der Name der Fehlerwarteschlange und der Kanalname benötigt.

```
Connection to Queue Manager
---------------------------
Queue Manager or JNDI CF   : []PEQM1
MQ Base Topic              : []/sf
MQ Channel                 : []A channel you have defined or for example SYS□
TEM.DEF.SVRCONN
MQ Conname                 : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL          : []
MQ Userid                  : []
MQ Password                : []
```

**Anmerkung:** Wenn Sie eine lokale Verbindung herstellen, ist der Kanalname nicht erforderlich. Sie müssen den Namen und das Basisthema des Warteschlangenmanagers nicht in der Konfigurationsdatei angeben, da sie später in die Befehlszeile aufgenommen werden können, wenn Sie die Brücke ausführen.

b) Geben Sie die Werte für die Verbindung mit Salesforce ein:

Als Minimalkonfiguration für die Verbindung werden die Salesforce-Benutzer-ID, das zugehörige Kennwort, das Sicherheitstoken und der Anmeldungsendpunkt benötigt. In Produktionsumgebungen können Sie den Consumer-Schlüssel und den geheimen Schlüssel für die OAuth-Sicherheit hinzufügen.

```
Connection to Salesforce
------------------------
Salesforce Userid (reqd)   : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)      : []Security_Token
Login Endpoint             : [https://login.salesforce.com]
Consumer ID                : []
Consumer Secret Key        : []
```

c) Geben Sie Werte für Zertifikatsspeicher für TLS-Verbindungen ein:

Mindestwerte, die für TLS-Verbindungen benötigt werden, sind der Pfad zum Schlüsselspeicher für TLS-Zertifikate und das Schlüsselspeicherkennwort. Wenn kein vertrauenswürdiger Speicherpfad oder kein Kennwort angegeben ist, werden die Parameter für den Schlüsselspeicher und das Kennwort für den vertrauenswürdigen Speicher und das Kennwort verwendet. Wenn Sie TLS

für Ihre IBM MQ-Warteschlangenmanagerverbindung verwenden, können Sie denselben Schlüsselspeicher verwenden.

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
PEQM1/ssl/key.jks
Keystore password          : []keystore_password
Trusted store for signer certificates : []
Trusted store password     : []
Use TLS for MQ connection  : [N]
```

d) Geben Sie die Werte für die Konfiguration des Verhaltens der IBM MQ Bridge to Salesforce ein:

Sie müssen die Option **Subscribe to MQ publications for platform events** vom Standardwert *N* in *J* ändern, um die Bridge zum Erstellen von Ereignisnachrichten zu verwenden. Sie müssen auch die Protokolldatei, in der Konfigurationsdatei oder in der Befehlszeile angeben.

```
Behaviour of bridge program
---------------------------
PushTopic Names             : []
Platform Event Names        : []
MQ Monitoring Frequency     : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

9. Optional: Erstellen Sie den IBM MQ-Service zur Steuerung der Programmausführung. Bearbeiten Sie die `mqsfbService.mqsc`-Beispieldatei so, dass sie auf die neu erstellte Konfigurationsdatei verweist, und nehmen Sie alle anderen Änderungen an den Befehlsparametern vor.

```
cat modified mqsfbService.mqsc | runmqsc PEQM1
```

10. Optional: Befolgen Sie die Anweisungen im Abschnitt Einführung in die IBM MQ Console, um die IBM MQ Console einzurichten.
11. Optional: Fügen Sie Ihrer Instanz der IBM MQ Console Widgets hinzu und konfigurieren Sie diese so, dass Sie Salesforce-Daten anzeigen können.
    a) Klicken Sie auf **Widget hinzufügen** .

    Das neue Widget wird geöffnet.

    b) Wählen Sie **Diagramme** aus.
    c) Klicken Sie auf das Symbol **Widget konfigurieren** in der Titelleiste des neuen Widgets.
    d) Optional: Geben Sie einen **Widgettitel** ein.
    e) Wählen Sie **Salesforce Bridge** im Dropdown-Menü **Ressource to monitor** ( **Quelle** ) aus.
    f) Wählen Sie **Bridge-Status** im Dropdown-Menü **Ressourcenklasse** aus.
    g) Wählen Sie **MQ-erstellte Plattformereignisse** im Dropdown-Menü **Ressourcentyp** aus.
    h) Wählen Sie im Dropdown-Menü **Resource element** (Ressourcenelement) die Option **Total MQ-erstellte Plattformereignisse** aus.
    i) Klicken Sie auf **Speichern**.

    Sie haben IBM MQ Console für die Anzeige der Gesamtzahl der erstellten IBM MQ-Plattformereignisse konfiguriert. Wenn die Bridge aktiv ist und Sie mit dem Einreihen von Nachrichten in das `/sf/mqtosfb/event/`*`Salesforce Platform Object Name/API name`*-Thema beginnen, zeigt das Widget die Anzahl der gesamten Nachrichtenereignisse an, die die Bridge erstellt hat.

## Deprecated Nachrichtenformat und Fehlernachrichten für die IBM MQ Bridge to Salesforce

Informationen zur Formatierung der Nachrichten, die von der IBM MQ Bridge to Salesforce verarbeitet werden.

Eine Anwendung setzt eine Nachricht an ein bestimmtes WS-Manager-Thema, z. B. /*root*/mqtosfb/event/MQPlatformEvent1__e, ab. Die Bridge subskribiert das Thema, ruft Inhalt aus den Nachrichten ab und verwendet diesen zum Veröffentlichen von Ereignisnachrichten für ein Salesforce-Plattformereignis.

Sie müssen ein Plattformereignis in Salesforce erstellen und die Inhaltsfelder definieren, bevor Sie die Bridge verwenden können, um Ereignisnachrichten für dieses Plattformereignis zu erstellen. Der Plattformereignisname und sein Inhalt bestimmen, wie Sie die IBM MQ-Nachricht formatieren müssen, die von der Bridge verarbeitet wird. Wenn Ihr Salesforce Plattformereignis **Object name** beispielsweise *MQPlatformEvent1* ist und Ihre beiden benutzerdefinierten Felder Textfelder mit den **API name** *MeinText__c* und *Name__c* sind, muss Ihre IBM MQ-Nachricht, die im Thema /*root*/mqtosfb/event/MQPlatformEvent1__e veröffentlicht wird, wie folgt ein ordnungsgemäß formatiertes JSON sein:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

Die Nachrichten, die von der Bridge konsumiert und erstellt werden, sind Text-(MQSTR) Nachrichten im JSON-Format. Die Eingabenachricht ist eine einfache JSON-Datei und Programme können Zeichenfolgenverkettungen verwenden, um sie zu generieren.

### Fehlermeldungen

Fehler können von der Bridge erkannt werden, z. B. wenn sich die Nachricht nicht im Textformat befindet oder nicht von Salesforce stammt, z. B., wenn der Plattformereignisname nicht vorhanden ist. Tritt bei der Verarbeitung der Eingabenachricht ein Fehler auf, wird die Nachricht zusammen mit den Eigenschaften, die den Fehler beschreiben, in die Brückenfehlerwarteschlange verschoben. Der Fehler wird auch in den *stderr* -Datenstrom für die Brücke geschrieben.

Fehler, die von Salesforce generiert werden, haben das JSON-Format. Im Folgenden werden einige Fehler angezeigt, die durch falsch formatierte Nachrichten verursacht werden:

Ungültiger Plattformereignisinhalt, Status 400 Text

```
[{"message":"No such column 'Name__c' on sobject of type MQPlatformEvent2__e","errorCode":"INVA
LID_FIELD"}
```

Ungültiger Plattformereignisname, Status 404-Text

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

Fehlerhaftes JSON, Status 400

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

Nachricht ist nicht JSON, Status 400 Text

```
[{"message":
  "Unexpected character ('h' (code 104)): expected a valid value (number, String, array, ob
ject, 'true', 'false' or 'null') at [line:1, column:2]",
  "errorCode":"JSON_PARSER_ERROR"}
```

Keine Textnachricht (nicht an Salesforcegesendet)

```
Error: Publication on topic ' /sf/mqtosfb/event/MQPlatformEvent1' does not contain a text forma
tted message
```

## Linux Deprecated IBM MQ Bridge to Salesforce ausführen

Führen Sie die IBM MQ Bridge to Salesforce aus, um eine Verbindung zu Salesforce und IBM MQ herzustellen. Wenn eine Verbindung besteht, kann die Bridge Subskriptionen für Salesforce-Themen erstellen

und Nachrichten erneut im IBM MQ-Thema veröffentlichen. Die Bridge kann auch Ereignisnachrichten für Salesforce-Plattformereignisse erstellen.

## Vorbereitende Schritte

Sie haben Konfigurationsschritte in der Task abgeschlossen:

- „IBM MQ Bridge to Salesforce konfigurieren” auf Seite 925
- „Ereignisnachrichten für Salesforce-Plattformereignisse erstellen” auf Seite 933

.

## Informationen zu diesem Vorgang

Verwenden Sie die Konfigurationsdatei, die Sie in der vorherigen Task erstellt haben, um die IBM MQ Bridge to Salesforce auszuführen. Wenn Sie nicht alle erforderlichen Parameter in Ihre Konfigurationsdatei eingeschlossen haben, stellen Sie sicher, dass sie in die Befehlszeile eingeschlossen werden.

## Vorgehensweise

1. Definieren Sie die Push-Themen oder Plattformereignisse in Salesforce, für die Sie oder das Plattformereignis abonnieren möchten, für das Sie Ereignisnachrichten erstellen möchten.
2. Starten Sie die IBM MQ Bridge to Salesforce, um eine Verbindung zu Salesforce und Ihrem Warteschlangenmanager herzustellen. Wenn Sie die Bridge für die Subskription von Salesforce-Ereignissen ausführen, geben Sie den Namen des Push-Themas oder des Plattformereignisses ein, das Sie in Schritt 1 definiert haben.

   ```
   runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName
   ```

   Wenn die Brücke verbunden ist, werden die folgenden Nachrichten zurückgegeben:

   - Wenn Sie die Bridge verwenden, um Salesforce-Push-Themen und -Plattformereignisse zu subskribieren, gilt Folgendes:

     ```
     Successful connection to queue manager QM1
     Warning: Subscribing to MQ-created platform events is not enabled.
     Successful login to Salesforce at https://eu11.salesforce.com
     Ready to process events.
     ```

   - Wenn Sie die Bridge verwenden, um Ereignisnachrichten für Salesforce-Plattformereignisse zu erstellen, gilt Folgendes:

     ```
     Successful connection to queue manager QM1
     Successful login to Salesforce at https://eu11.salesforce.com
     Successful subscription to '/sf/mqtosfb/event/+' for MQ-created platform events
     Ready to process events.
     ```

3. Optional: Beheben Sie Fehler bei der Verbindung mit Ihrem Warteschlangenmanager und Salesforce, wenn die Nachrichten, die nach Ihrer Ausführung der Bridge zurückgegeben werden, darauf hinweisen, dass eine Verbindung nicht erfolgreich war.
   a) Geben Sie den Befehl im Debugmodus mit der Debugoption 1 aus.

      ```
      runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 1
      ```

      Die Brückenschritte über die Verbindung sind aufgebaut und zeigen die Verarbeitungsnachrichten im Modus 'terse' (terse) an.

   b) Geben Sie den Befehl im Debugmodus mit der Debugoption 2 aus.

      ```
      runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 2
      ```

      Die Brückenschritte über die Verbindung werden aufgebaut und die Verarbeitungsnachrichten werden im ausführlichen Modus angezeigt. Die vollständige Ausgabe wird in Ihre Protokolldatei geschrieben.

4. Generieren Sie Ereignisse mithilfe der Salesforce-Schnittstelle, um Datensätze in der Datenbank zu ändern.
5. Rufen Sie die IBM MQ Console auf, um Änderungen an Push-Themen in dem Widget anzuzeigen, das Sie in der vorherigen Task konfiguriert haben.

### Nächste Schritte

Verwenden Sie die *MQSFB_EXTRA_JAVA_OPTIONS*-Variable, um JVM-Eigenschaften zu übergeben, z. B. um die Tracefunktion von IBM MQ zu aktivieren. Weitere Informationen hierzu finden Sie im Abschnitt Traceverarbeitung für die IBM MQ Bridge to Salesforce.

**Zugehörige Tasks**

IBM MQ Bridge to Salesforce überwachen

**Zugehörige Verweise**

runmqsfb (IBM MQ Bridge to Salesforce ausführen)

## Linux z/OS MQ Adv. Deprecated IBM MQ für die Verwendung mit Blockchain konfigurieren

Richten Sie die IBM MQ Bridge to blockchain ein und führen Sie sie aus, um eine sichere Verbindung zu einem Warteschlangenmanager für IBM MQ Advanced oder IBM MQ Advanced for z/OS Value Unit Edition und IBM Blockchain herzustellen. Verwenden Sie die Bridge, um eine asynchrone Verbindung zu einer Ressource in Ihrer Blockchain herzustellen, nach ihr zu suchen und ihren Status zu aktualisieren, indem Sie eine Messaging-Anwendung verwenden, die eine Verbindung zu Ihrem Warteschlangenmanager für IBM MQ Advanced oder IBM MQ Advanced for z/OS VUE herstellt.

### Vorbereitende Schritte

**Anmerkungen:**

- Deprecated IBM MQ Bridge to blockchain ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-341). Die Blockchain -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.
- Removed V 9.3.2 Für Continuous Deliverywird die IBM MQ Bridge to blockchain unter IBM MQ 9.3.2aus dem Produkt entfernt.
- LTS IBM beabsichtigt, die Funktionalität von Long Term Support -Releases in zukünftigen Fixpacks zu entfernen. Wenn Sie über Anwendungen verfügen, die von dieser Änderung betroffen sind, wenden Sie sich an den IBM Support.

  **Achtung:** Die auf Hyperledger Composer erstellte IBM MQ Bridge to blockchain -Instanz wird nicht mehr unterstützt.

  Sie müssen IBM MQ 9.1.4 oder höher ausführen, um die IBM MQ Bridge to blockchain auf Hyperledger Fabricaufbauen zu können.
- IBM MQ Bridge to blockchain ist nur für die Verbindung zu den folgenden Warteschlangenmanagern verfügbar:
  - Linux IBM MQ Advanced, oder
  - z/OS IBM MQ Advanced for z/OS VUE
- Der Warteschlangenmanager muss sich auf der gleichen Befehlsebene wie die Bridge oder höher befinden, z. B. IBM MQ 9.3.0.
- IBM MQ Bridge to blockchain wird für die Verwendung mit Ihrem Blockchain-Netz unterstützt, das auf der Architektur von Hyperledger Fabric 1.4 basiert.

## Informationen zu diesem Vorgang

Blockchain ist ein gemeinsam genutzter, verteilter, digitaler Ledger, der aus einer Kette von Blöcken besteht, die sich auf Transaktionen zwischen Peers in einem Netzwerk geeinigt haben. Jeder Block in der Kette ist mit dem vorherigen Block verknüpft, und so weiter zurück zu der ersten Transaktion.

IBM Blockchain basiert auf Hyperledger Fabric. Sie können damit lokal mit Docker oder in einem Container-Cluster in IBM Cloud entwickeln. Sie können auch Ihr IBM Blockchain-Netz in der Produktion aktivieren und verwenden, um ein Geschäftsnetz mit einem hohen Maß an Sicherheit, Datenschutz und Leistung zu erstellen und zu steuern. Weitere Informationen finden Sie in der IBM Blockchain-Plattform .

Hyperledger Fabric ist ein Open-Source-Blockchain-Framework für Unternehmen, das von Mitgliedern der Hyperledger Projectgemeinsam entwickelt wird, einschließlich IBM als ursprünglicher Codeentwickler. Hyperledger Project oder Hyperledgerist eine globale, interaktive Linux Foundation Open-Source-Initiative zur Förderung branchenübergreifender Blockchain-Technologien. Weitere Informationen finden Sie unter IBM Blockchain, Hyperledger-Projekte und Hyperledger Fabric.

Wenn Sie IBM MQ Advanced oder IBM MQ Advanced for z/OS VUE und IBM Blockchain bereits verwenden, können Sie mit der IBM MQ Bridge to blockchain einfache Abfragen und Aktualisierungen senden und Antworten aus Ihrem Blockchain-Netz empfangen. Auf diese Weise können Sie Ihre lokale IBM-Software in einen Blockchain-Service der Cloud integrieren.

Sie können eine kurze Übersicht über den Bridge-Betriebsprozess in Abbildung 1 anzeigen. Eine Benutzeranwendung versetzt eine JSON-formatierte Nachricht in die Eingabe-/Anforderungswarteschlange auf dem IBM MQ Advanced- oder IBM MQ Advanced for z/OS VUE-Warteschlangenmanager. Die Brücke stellt eine Verbindung zum WS-Manager her, ruft die Nachricht aus der Eingabe-/Anforderungswarteschlange ab, prüft, ob das JSON korrekt formatiert ist, und gibt dann die Abfrage oder eine Aktualisierung der Blockkette aus. Die von der Blockchain zurückgegebenen Daten werden von der Bridge analysiert und in die Antwortwarteschlange gestellt, wie in der ursprünglichen IBM MQ-Anforderungsnachricht definiert. Die Benutzeranwendung kann eine Verbindung zum WS-Manager herstellen, die Antwortnachricht aus der Antwortwarteschlange abrufen und die Informationen verwenden.

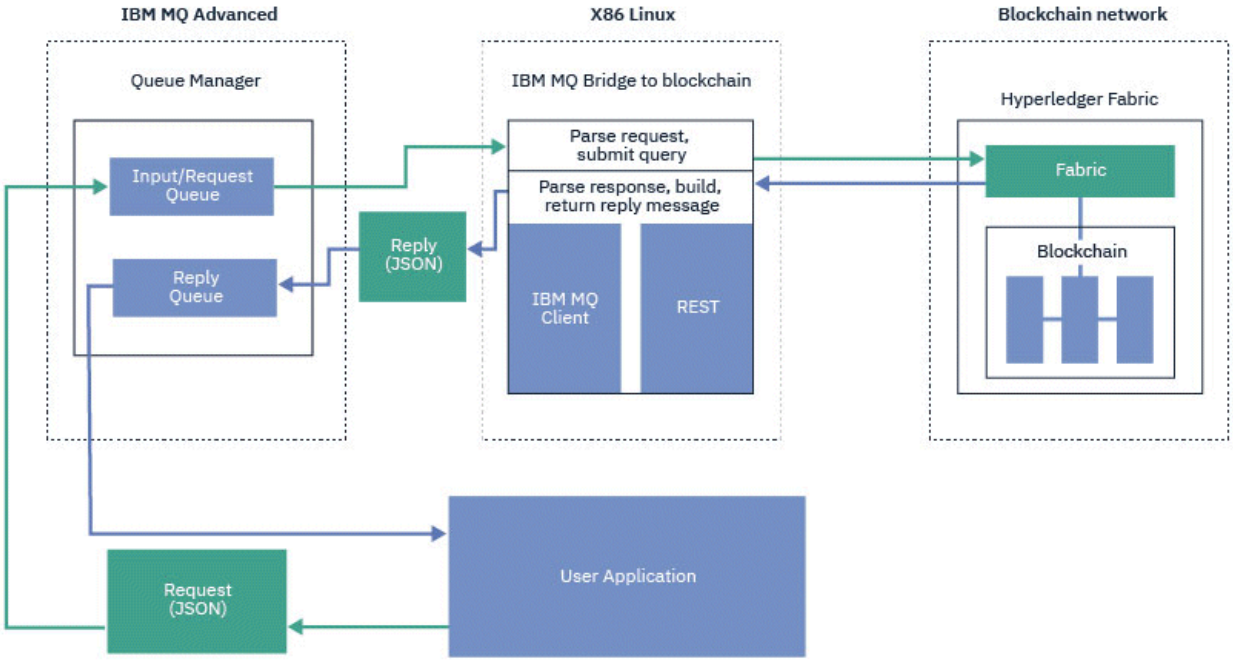



*Abbildung 98. IBM MQ Bridge to blockchain*

Sie können die IBM MQ Bridge to blockchain so konfigurieren, dass sie eine Verbindung zu einem Blockchain-Netz als Teilnehmer oder Peer herstellt. Wenn die Bridge ausgeführt wird, fordert eine Messaging-Anwendung die Brücke an, um die Chaincode-Routinen zu steuern, die den Status der Ressource abfragen oder aktualisieren und die Ergebnisse als Antwort an die Messaging-Anwendung zurückgeben.

## Vorgehensweise

1. Erstellen und starten Sie einen Warteschlangenmanager oder starten Sie einen vorhandenen Warteschlangenmanager, den Sie mit Ihrer IBM MQ Bridge to blockchain verwenden möchten.

   Warteschlangenmanager erstellen:

   ```
   crtmqm adv_qmgr_name
   ```

   Warteschlangenmanager starten:

   ```
   strmqm adv_qmgr_name
   ```

2. Erstellen Sie die Warteschlangen für die Bridge, die im Script **DefineQ.mqsc** definiert sind.

   Musterbrückenwarteschlangendefinitionen werden für die standardmäßigen benannten Warteschlangen bereitgestellt, die für Folgendes verwendet werden:

   - Benutzerberechtigungsnachweise, z. B. `SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE`
   - Nachrichteneingabe an die Brücke, z. B. `APPL1.BLOCKCHAIN.INPUT.QUEUE`
   - Antworten aus der Blockchain, z. B. `APPL1.BLOCKCHAIN.REPLY.QUEUE`

   Geben Sie den folgenden Befehl aus dem Verzeichnis `/opt/mqm/mqbc/samp` aus:

   ```
   runmqsc adv_qmgr_name < ./DefineQ.mqsc
   ```

   Unterschiedliche Anwendungen können die gleiche Eingabewarteschlange verwenden, aber Sie können mehrere Antwortwarteschlangen angeben, eine für jede Ihrer Anwendungen. Sie müssen definierte Antwortwarteschlangen nicht verwenden. Wenn Sie dynamische Warteschlangen für Antworten verwenden möchten, müssen Sie ihre Sicherheitskonfiguration in Betracht ziehen.

## Ergebnisse

Sie haben die Warteschlangen erstellt, die die Bridge für die Verarbeitung von Nachrichten aus IBM MQ und Ihrem Blockchain-Netz benötigt.

## Nächste Schritte

Verwenden Sie die Informationen zum Warteschlangenmanager für IBM MQ Advanced oder IBM MQ Advanced for z/OS VUE und die Berechtigungsnachweise aus Ihrem Blockchain-Netz, um eine Konfigurationsdatei für die IBM MQ Bridge to blockchain zu erstellen.

# Deprecated Konfigurationsdatei für den IBM MQ Bridge to blockchain erstellen

Geben Sie Ihren Warteschlangenmanager und Ihre Blockchain-Netzparameter ein, um die Konfigurationsdatei für den IBM MQ Bridge to blockchain zu erstellen, um eine Verbindung zu Ihren IBM MQ- und IBM Blockchain-Netzen herzustellen.

## Vorbereitende Schritte

- Sie haben Ihr Blockkettennetz erstellt und konfiguriert.
- Sie verfügen über die Berechtigungsnachweisdatei aus Ihrem Blockkettennetz.
- Sie haben die IBM MQ Bridge to blockchain in Ihrer x86-Linux-Umgebung installiert.

  Weitere Informationen finden Sie unter IBM MQ-Server unterLinux installieren und IBM MQ-RPM-Komponenten für Linux-Systeme.
- Sie haben Ihren IBM MQ Advanced-Warteschlangenmanager gestartet.

## Informationen zu diesem Vorgang

Diese Task führt Sie durch die minimale Konfiguration, die zum Erstellen der IBM MQ Bridge to blockchain-Konfigurationsdatei erforderlich ist, und eine erfolgreiche Verbindung zu Ihren IBM Blockchain- und IBM MQ-Netzen.

Sie können die Bridge verwenden, um eine Verbindung zu Blockchain-Netzen herzustellen, die auf Hyperledger Fabric 1.4 architecture basieren. Um die Brücke verwenden zu können, benötigen Sie Konfigurationsinformationen aus Ihrem Blockkettennetz. In jedem Schritt in dieser Task finden Sie Beispielkonfigurationsdetails, die auf zwei unterschiedlich konfigurierten Blockkettennetzen basieren:

- Hyperledger Fabric-Netz, das in Docker ausgeführt wird. Weitere Informationen hierzu finden Sie in den Abschnitten Erste Schritte mit Hyperledger Fabric, Schreiben der ersten Anwendung und „Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungsnachweisen” auf Seite 945.
- Hyperledger Fabric-Netz, das in einem Kubernetes-Cluster in IBM Cloud ausgeführt wird. Weitere Informationen finden Sie unter Entwickeln in einer Cloud-Sandbox auf IBM Blockchain Platform .

## Vorgehensweise

1. Führen Sie die Brücke aus, um eine Konfigurationsdatei zu erstellen.

   Sie benötigen die Parameter aus Ihrer Berechtigungsnachweisdatei für das Blockchain-Netz und von Ihrem IBM MQ Advanced-Warteschlangenmanager.

   ```
   runmqbcb -o config_file_name.cfg
   ```

   Wie das folgende Beispiel zeigt, werden die vorhandenen Werte in den eckigen Klammern angezeigt. Drücken Sie `Enter` , um vorhandene Werte zu akzeptieren, drücken Sie `Space` dann `Enter` , um die Werte zu löschen, und geben Sie in den Klammern `Enter` ein, um neue Werte hinzuzufügen. Sie können Listen von Werten (z. B. Peers) durch Kommas trennen oder indem Sie jeden Wert in einer neuen Zeile eingeben. Eine leere Zeile beendet die Liste.

   **Anmerkung:** Sie können die vorhandenen Werte nicht bearbeiten. Sie können sie behalten, ersetzen oder löschen.
2. Geben Sie Werte für die Verbindung zu Ihrem IBM MQ Advanced-Warteschlangenmanager ein.

   Die Mindestwerte, die für die Verbindung erforderlich sind, sind der Name des Warteschlangenmanagers, die Namen der von Ihnen definierten Brückeneingabe- und -Identitätswarteschlangen. Für Verbindungen zu fernen Warteschlangenmanagern benötigen Sie außerdem **`MQ Channel`** und **`MQ Conname`** (die Hostadresse und der Port, an dem bzw. der der Warteschlangenmanager ausgeführt wird). Wenn Sie TLS für die Verbindung zu IBM MQ in Schritt „4” auf Seite 944verwenden möchten, müssen Sie JNDI oder CCDT verwenden und **`MQ CCDT URL`** oder **`JNDI implementation class`** und **`JNDI provider URL`** entsprechend angeben.
3. 13. Geben Sie die Hyperledger Fabric-Serverberechtigungsnachweise für Ihr Netz ein.

   Im folgenden Code wird die erwartete Ausgabe gezeigt:

   ```
   Fabric Server
   -------------
   Network configuration file          : []connection-tls.json
   Wallet                              : []
   User Name                           : []User1
   Certificate                         : []<path_to_user_certificate>
   Private Key                         : []<path_to_private_key>/private_key.pem
   Organisation                        : []Org1MSP
   ```
4. Geben Sie Zertifikate für TLS-Verbindungen ein.

   Lassen Sie diesen Bereich leer, wenn kein Wert vorhanden ist.

   ```
   Certificate stores for MQ TLS connections
   -----------------------------------------
   Personal keystore                   : []
   Keystore password                   : []
   ```

```
Trusted store for signer certs       : []
Trusted store password               : []
```

5. Geben Sie den Pfad zu der Protokolldatei ein, in die die Bridge-Protokolle geschrieben werden sollen.

```
Behavior of bridge program
--------------------------
Runtime logfile for copy of stdout/stderr : []bridgelog.log
Number of logfiles                        : [3]
Maximum size of each logfile (bytes) : [2097152]
```

**Achtung:** Zuvor wurden Einzelheiten zu den Peers, zu Auftraggebern und der Zertifizierungsstelle in dieser Bridge-Konfiguration gespeichert. Jetzt werden diese Informationen allerdings in der *Netzkonfigurationsdatei* gespeichert, die mit dem Hyperledger Fabric-Serverabschnitt der Konfiguration verknüpft ist.

### Ergebnisse

Sie haben die Konfigurationsdatei erstellt, die der IBM MQ Bridge to blockchain verwendet, um eine Verbindung zu Ihrem IBM Blockchain-Netz und zu Ihrem IBM MQ Advanced-Warteschlangenmanager herzustellen.

### Nächste Schritte

Führen Sie die in „IBM MQ Bridge to blockchain ausführen” auf Seite 949 beschriebenen Schritte aus.

## Deprecated Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungsnachweisen

Inhalt der `.yml`-Datei aus Ihrem lokal instanziierten Hyperledger Fabric-Blockchain-Netzwerk, das in Docker ausgeführt wird, das Sie zum Konfigurieren Ihres IBM MQ Bridge to blockchains verwenden können.

IBM MQ Bridge to blockchain kann mit folgenden Komponenten verbunden werden:

- Linux IBM MQ Advanced, oder
- z/OS IBM MQ Advanced for z/OS VUE

(nur Warteschlangenmanager)

Nachdem Sie in den Erste Schritte mit Hyperledger Fabric-Lernprogrammen gearbeitet, Was hinter den Kulissen passiert verstanden und Ihr Netz mithilfe eines der Hyperledger Fabric-Beispiele gestartet haben, sollten Sie die folgende Konfigurationsdatei in Ihrem `/blockchain/fabric-samples/basic-network`-Ordner haben.

Wenn Sie eine Verbindung zu Ihrem Blockchain-Netz herstellen wollen, müssen Sie beim „Konfigurationsdatei für den IBM MQ Bridge to blockchain erstellen” auf Seite 943 die Konfigurationsdetails aus dieser Datei verwenden.

```
{
  "name": "basic-network",
  "version": "1.0.0",
  "client": {
    "organization": "Org1",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  },
  "channels": {
    "mychannel": {
```

```
      "orderers": [
        "orderer.example.com"
      ],
      "peers": {
        "peer0.org1.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": true,
          "ledgerQuery": true,
          "eventSource": true
        },
        "peer0.org2.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": false,
          "ledgerQuery": true,
          "eventSource": false
        }
      }
    }
  },
  "organizations": {
    "Org1": {
      "mspid": "Org1MSP",
      "peers": [
        "peer0.org1.example.com"
      ],
      "certificateAuthorities": [
        "ca-org1"
      ],
      "adminPrivateKeyPEM": {
        "path": "$<path_to_private_key>/admin_private_key"
      },
      "signedCertPEM": {
        "path": "<path_to_org_signed_cert>/Admin@org1.example.com-cert.pem"
      }
    },
    "Org2": {
      "mspid": "Org2MSP",
      "peers": [
        "peer0.org2.example.com"
      ],
      "certificateAuthorities": [
        "ca-org2"
      ]
    }
  },
  "orderers": {
    "orderer.example.com": {
      "url": "grpcs://localhost:7050",
      "mspid": "OrdererMSP",
      "grpcOptions": {
        "ssl-target-name-override": "orderer.example.com",
        "hostnameOverride": "orderer.example.com"
      },
      "tlsCACerts": {
        "path": "<path_to_orderer_cert>/ca.crt"
      },
      "adminPrivateKeyPEM": {
        "path": <path_to_orderers_private_key>/<private_key>"
      },
      "signedCertPEM": {
        "path": "<path_to_orderer_signed_cert>/Admin@example.com-cert.pem"
      }
    }
  },
  "peers": {
    "peer0.org1.example.com": {
      "url": "grpcs://localhost:7051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org1.example.com",
        "hostnameOverride": "peer0.org1.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": <path_to_peer_cert>/ca.crt"
      }
    },
    "peer0.org2.example.com": {
      "url": "grpcs://localhost:9051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org2.example.com",
        "hostnameOverride": "peer0.org2.example.com",
        "request-timeout": 120001
```

```
      },
      "tlsCACerts": {
        "path": "<path_to_peer_cert>/ca.crt"
      }
    }
  },
  "certificateAuthorities": {
    "ca-org1": {
      "url": "https://localhost:7054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org1.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    },
    "ca-org2": {
      "url": "https://localhost:8054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org2.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    }
  }
}
```

# Linux Deprecated Nachrichtenformate für die IBM MQ Bridge to blockchain ab IBM MQ 9.2.0

Informationen zur Formatierung der Nachrichten, die von der IBM MQ Bridge to blockchain gesendet und empfangen werden.

In einer Anwendung wird angefordert, dass die IBM MQ Bridge to blockchain den Hyperledger Fabric-Server steuert, um auf Informationen zu reagieren, die in der Blockchain enthalten sind. Die Anwendung führt dies durch die Anforderung einer Anforderungsnachricht in die Brückenanforderungswarteschlange durch. Die Ergebnisse der Anforderung werden von der Bridge in eine Antwortnachricht formatiert. Die Bridge verwendet Informationen, die in den Feldern **ReplyToQ** und **ReplyToQMgr** aus dem MQMD der Anforderungsnachricht als Ziel für die Antwortnachricht enthalten sind.

Die Anforderungs- und Antwortnachrichten sind Textnachrichten (MQSTR) im JSON-Format und enthalten vier Elemente.

## Anforderungsnachrichtenformat

Anforderungsnachrichten enthalten die folgenden Attribute:

**Betrieb**
Zeichenfolge - ohne Unterscheidung von Groß-/Kleinschreibung

`submit` für Aktualisierungen oder `evaluate` für Abfragen

**Netz**
Zeichenfolge-manchmal auch als `channel` in Hyperledger Fabric bezeichnet

**contract**
Zeichenfolge - Der Smart Contract oder das Chaincode-Paket, das aufgerufen werden soll

**args**
Array - normalerweise Zeichenfolgen, aber einige Elemente können verschachtelte JSON-Objekte sein.

Die tatsächlichen Argumente für **contract**, einschließlich Methodenname.

For example:

```
{
  "operation" : "Evaluate",
  "network"   : "mychannel",
  "contract"  : "marbles0",
  "args"      : [ "readMarble" , "marble1" ]
}
```

**Anmerkung:** Die Bridge überprüft die Inhalte nur darauf, ob diese Elemente vorhanden sind und ob es sich bei der Nachricht um eine gültige JSON-Nachricht handelt. Die Bridge setzt darauf, dass Hyperledger Fabric die Anforderung verarbeitet oder Fehler zurückgibt.

## Antwortnachrichtenformat

Antwortnachrichten haben ihre Korrelations-ID auf die Nachrichten-ID der eingehenden Nachricht gesetzt. Alle vom Benutzer definierten Eigenschaften werden aus der Anforderungsnachricht in die Antwortnachricht kopiert. Die Benutzer-ID in der Antwort wird auf die Benutzer-ID des Erstellers gesetzt.

Der **statusCode** ist ein HTTP-Statuscode. Wenn der Fehler von IBM MQ oder von der Brücke stammt, wird ein geeigneter **statusCode** verwendet.

**statusType** ist eine Zeichenfolge, entweder *SUCCESS* oder *FAILURE*.

Für erfolgreiche Anforderungen enthält das Element "**data**" in der Antwortnachricht die Antwort aus der aufgerufenen Hyperledger Composer-REST-API.

Ein Beispiel für eine erfolgreiche Verarbeitung:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

Alle Fehlerantworten haben die gleichen Felder, unabhängig davon, ob sie von der Brücke selbst generiert werden, von den Aufrufen an den Hyperledger Composer-REST-Server, von der Blockchain oder vom Chaincode-Aufruf. For example:

- Falsche JSON-Eingabenachricht

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method"."
}
```

- Anforderung, die vom Hyperledger Composer-REST-Server nicht verarbeitet werden konnte

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

Anwendungen können feststellen, ob die Anforderung erfolgreich war oder fehlgeschlagen ist, indem sie entweder die Zeichenfolge **statusType** oder das Vorhandensein des Datenfelds betrachtet. Wenn bei der Verarbeitung der Eingabenachricht ein Fehler auftritt und die Brücke sie nicht an die Blockkette sendet, ist der von der Brücke zurückgegebene Wert ein MQRC-Wert (normalerweise **MQRC_FORMAT_ERROR**).

# Deprecated IBM MQ Bridge to blockchain ausführen

Führen Sie die IBM MQ Bridge to blockchain aus, um eine Verbindung zu IBM Blockchain und IBM MQ herzustellen. Wenn sie verbunden ist, ist die Bridge bereit, Anforderungsnachrichten zu verarbeiten, sie an Ihr Hyperledger Composer-Blockkettennetz zu senden und die Antworten zu empfangen und zu verarbeiten.

## Informationen zu diesem Vorgang

Verwenden Sie die Konfigurationsdatei, die Sie in der vorherigen Task erstellt haben, um die IBM MQ Bridge to blockchain auszuführen.

## Vorgehensweise

1. Starten Sie den IBM MQ Advanced-Warteschlangenmanager, den Sie mit der Bridge verwenden möchten.
2. Starten Sie IBM MQ Bridge to blockchain, um eine Verbindung zu Ihrem Hyperledger Composer-REST-Server herzustellen, und starten Sie den IBM MQ Advanced-Warteschlangenmanager.

   Führen Sie den Befehl bridge aus.

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   ```

   Wenn die Brücke verbunden ist, wird die Ausgabe ähnlich der folgenden zurückgegeben:

   ```
   2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
   5724-H72 (C) Copyright IBM Corp. 2017, 2024.

   2018-05-17 14:28:19.331 BST Ready to process input messages.
   ```

3. Optional: Beheben Sie Fehler bei den Verbindungen zu Ihrem IBM MQ Advanced-Warteschlangenmanager und zu Ihrem Blockchain-Netz, wenn die Nachrichten, die nach der Ausführung der Bridge zurückgegeben werden, darauf hinweisen, dass eine Verbindung nicht erfolgreich ist.

   a) Geben Sie den Befehl im Debugmodus mit der Debugoption 1 aus.

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   -d 1
   ```

   Die Brückenschritte über die Verbindung sind aufgebaut und zeigen die Verarbeitungsnachrichten im Modus 'terse' (terse) an.

   b) Geben Sie den Befehl im Debugmodus mit der Debugoption 2 aus.

   ```
   runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   -d 2
   ```

Die Brückenschritte über die Verbindung werden aufgebaut und die Verarbeitungsnachrichten werden im ausführlichen Modus angezeigt. Die vollständige Ausgabe wird in Ihre Protokolldatei geschrieben.

### Ergebnisse

Sie haben die IBM MQ Bridge to blockchain gestartet und mit Ihrem Hyperledger Composer-REST-Server mit dem Warteschlangenmanager und dem Blockchain-Netz verbunden.

### Nächste Schritte

- Führen Sie die Schritte in „IBM MQ Bridge to blockchain -Clientbeispiel unter z/OS ausführen” auf Seite 952 aus, um eine Abfrage oder eine Aktualisierungsnachricht an Ihr Blockchain-Netz zu formatieren und zu senden.
- Verwenden Sie die *MQBCB_EXTRA_JAVA_OPTIONS*-Variable, um JVM-Eigenschaften zu übergeben, z. B. um die Tracefunktion von IBM MQ zu aktivieren. Weitere Informationen hierzu finden Sie im Abschnitt Traceverarbeitung für die IBM MQ Bridge to blockchain.

## z/OS Deprecated Nachrichtenformate für IBM MQ Bridge to blockchain vor IBM MQ 9.2.0 unter z/OS

Informationen zur Formatierung der Nachrichten, die von der IBM MQ Bridge to blockchain gesendet und empfangen werden.

**Achtung:** Das vorhandene Format für die Nachrichtenformate ist veraltet. Wenn Sie über ein Hyperledger Fabric-Netz verfügen, verwenden Sie das Format der in „Nachrichtenformate für die IBM MQ Bridge to blockchain ab IBM MQ 9.2.0” auf Seite 947 beschriebenen Nachrichten aus IBM MQ 9.2.0.

Eine Anwendung fordert, dass die IBM MQ Bridge to blockchain die über Hyperledger Composer definierte REST-API ansteuert, um Informationen zu bearbeiten, die in der Blockchain gespeichert sind. Die Anwendung führt dies durch die Anforderung einer Anforderungsnachricht in die Brückenanforderungswarteschlange durch. Die Ergebnisse der REST-Anforderung werden von der Bridge in eine Antwortnachricht formatiert. Die Bridge verwendet Informationen, die in den Feldern **ReplyToQ** und **ReplyToQMgr** aus dem MQMD der Anforderungsnachricht als Ziel für die Antwortnachricht enthalten sind.

Die Anforderungs- und Antwortnachrichten sind Textnachrichten (MQSTR) im JSON-Format.

### Anforderungsnachrichtenformat

Anforderungsnachrichten enthalten drei Attribute:

**Methode**
Das REST-Verb, das zum Aufrufen der REST-API von Hyperledger Composer verwendet wird, z. B. POST, DELETE oder GET

**path**
Der Pfad zur Hyperledger Composer-REST-API. Dies wird der Basis-Server-URL hinzugefügt. Der Pfad muss mit "`api/`" beginnen.

**Hauptteil**
Der methodenspezifische Inhalt. Dies ist häufig eine JSON-Struktur.

Im folgenden Beispiel wird die Methode POST zum Pfad `api/Trader`verwendet, um ein neues Trader-Objekt zu erstellen. Der Hauptteil gibt die Trader-Klasse an, wie sie vom Hyperledger Composer-Modell des Benutzers definiert ist, und gibt außerdem die zusätzlichen Werte an, die erforderlich sind, um ein neues Trader-Objekt innerhalb des Blockkettennetzwerks zu erstellen.

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class" : "org.example.trading",
```

```
    "tradeId" : "Trader2",
    "firstName": "Jane",
    "lastName" : "Doe"
```

## Antwortnachrichtenformat

Antwortnachrichten haben ihre Korrelations-ID auf die Nachrichten-ID der eingehenden Nachricht gesetzt. Alle vom Benutzer definierten Eigenschaften werden aus der Anforderungsnachricht in die Antwortnachricht kopiert. Die Benutzer-ID in der Antwort wird auf die Benutzer-ID des Erstellers gesetzt.

Der **statusCode** ist ein HTTP-Statuscode. Wenn der Fehler von IBM MQ oder von der Brücke stammt, wird ein geeigneter **statusCode** verwendet.

**statusType** ist eine Zeichenfolge, entweder *SUCCESS* oder *FAILURE*.

Für erfolgreiche Anforderungen enthält das Element "**data**" in der Antwortnachricht die Antwort aus der aufgerufenen Hyperledger Composer-REST-API.

Ein Beispiel für eine erfolgreiche Verarbeitung:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

Alle Fehlerantworten haben die gleichen Felder, unabhängig davon, ob sie von der Brücke selbst generiert werden, von den Aufrufen an den Hyperledger Composer-REST-Server, von der Blockchain oder vom Chaincode-Aufruf. For example:

- Falsche JSON-Eingabenachricht

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method"."
}
```

- Anforderung, die vom Hyperledger Composer-REST-Server nicht verarbeitet werden konnte

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

Anwendungen können feststellen, ob die Anforderung erfolgreich war oder fehlgeschlagen ist, indem sie entweder die Zeichenfolge **statusType** oder das Vorhandensein des Datenfelds betrachtet. Wenn bei der Verarbeitung der Eingabenachricht ein Fehler auftritt und die Brücke sie nicht an die Blockkette sendet, ist der von der Brücke zurückgegebene Wert ein MQRC-Wert (normalerweise **MQRC_FORMAT_ERROR**).

# z/OS Deprecated IBM MQ Bridge to blockchain -Clientbeispiel unter z/OS ausführen

Sie können das JMS-Clientbeispiel verwenden, das mit IBM MQ Bridge to blockchain bereitgestellt wird, um eine Nachricht in die Eingabewarteschlange zu stellen, die von der Blockchain-Bridge überprüft wird, und die empfangene Antwort anzuzeigen. Dieses Beispiel basiert auf der Verwendung der IBM MQ Bridge to blockchain-Integration in das Hyperledger Composer Trader-Netzwerk-Beispiel.

## Vorbereitende Schritte

Weitere Informationen finden Sie unter /trade_network .

Ihr IBM MQ Bridge to blockchain wird ausgeführt und ist mit Ihrem IBM MQ Advanced-oder IBM MQ Advanced for z/OS VUE-Warteschlangenmanager sowie mit Ihrem Blockchain-Netzwerk verbunden.

## Informationen zu diesem Vorgang

Suchen Sie die JMS-Beispielanwendung (`ComposerBCBSamp.java`) im Verzeichnis `samp` der IBM MQ Bridge to blockchain.

Beispiel: `<MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java`, wobei `<MQ_INSTALL_ROOT>` die folgende ist:

- Linux Das Verzeichnis, in dem IBM MQ installiert ist.
- z/OS Das z/OS UNIX System Services-Verzeichnis, in dem Sie z/OS UNIX-Komponenten von IBM MQ installiert sind.

.

## Vorgehensweise

1. Bearbeiten Sie die Java-Quellendatei des Clients.

   Befolgen Sie die Anweisungen im Beispiel, um ihn so zu konfigurieren, dass er mit Ihrer IBM MQ-Umgebung und Ihrem Blockchain-Netz übereinstimmt.

   Der folgende Code aus dem Beispiel definiert drei JSON-Anforderungsnachrichten, die an die Bridge gesendet werden sollen:

   a. Erstens, um ein vorhandenes '`commodity`' zu entfernen

   b. Zweitens, um eine neue '`commodity`', '`owner`' und zugehörige Werte zu erstellen,

   c. Schließlich werden die neuen Informationen zum '`commodity`' nach den vorherigen beiden Anforderungsnachrichten angezeigt.

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
```

```
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
  }
```

2. Kompilieren Sie das Beispiel.

   Verweisen Sie auf die IBM MQ-Clientklassen und die JSON4J.jar-Datei, die im Bridge-Verzeichnis geliefert werden.

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar Compo
serBCBClient.java
```

3. Führen Sie die kompilierte Klasse aus.

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:. Compo
serBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
  {"method":"DELETE"," path ":"api\/Commodity\/BC"}
Response text:
{
   "statusCode": 204,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": ""
}
SUCCESS
Sent message:
{"body":{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"descripti
on":"Blockchain Sample Description","mainExchange":"My Exchange","tradingSymbol":"BC"},"ope
ration":"POST","url":"Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "Me",
      "quantity": 100,
      "tradingSymbol": "BC"
   }
}
SUCCESS
Sent message:
  {"method":"GET","path":"api\/Commodity"}
Response text:
```

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": [
      {
         "$class": "org.example.trading.Commodity",
         "description": "Blockchain Sample Description",
         "mainExchange": "My Exchange",
         "owner": "resource:org.example.trading.Trader#Me",
         "quantity": 100,
         "tradingSymbol": "BC"
      }
   ]
}
SUCCESS
```

Das Feld **message** enthält entweder "OK" für eine erfolgreich verarbeitete Nachricht oder im Falle einer fehlgeschlagenen Anforderung Informationen zur Ursache des Fehlers.

Wenn der Client eine Zeitlimitüberschreitung beim Warten auf die Antwort empfängt, überprüfen Sie, ob die Brücke aktiv ist.

# Linux Deprecated Zusätzliche Konfigurationsoptionen für IBM MQ Bridge to blockchain

Ab IBM MQ 9.2.0gibt es eine Änderung an der Art und Weise, wie Traceerstellung und Protokollierung unter IBM MQ Bridge to blockchainfunktionieren.

## Änderungen ab IBM MQ 9.1.0 IBM MQ Bridge to blockchain

Mit Ausnahme des Rotierens von Protokolldateien gibt es ab der IBM MQ 9.1.0-Bridge standardmäßig keine Änderungen am Verhalten. Weitere Informationen finden Sie unter „Protokollrotation” auf Seite 954.

## Interaktion von Trace und Debug

Ab IBM MQ 9.2.0 wird das Debug-Flag wie zuvor für IBM MQ 9.1.0 ausgeführt. Das bedeutet, dass *-d1* Debuginformationen übermittelt und *-d2* die Debugprotokollierung für die vorausgesetzten Komponenten aktiviert. Wenn Sie allerdings den IBM MQ-Trace beim Start der Bridge aktiviert haben, ist die Berichterstellung auf Ebene *-d2* automatisch aktiviert.

## Protokollrotation

Ab IBM MQ 9.2.0 werden als Standardverhalten für die Protokolldatei drei Protokolldateien mit einer Größe von jeweils 2 MB verwendet. Sie können diese Werte mithilfe zusätzlicher Konfigurationseigenschaften überschreiben. Das vorhandene Konfigurationsattribut oder der Befehlszeilenparameter für die Protokolldatei wird als Basisname für die Protokolle verwendet, wobei noch ein Index hinzugefügt wird.

Wenn die konfigurierte Protokolldatei

- keinen Dateityp hat, wird der Index am Ende des Dateinamens hinzugefügt.

  Wenn die Protokolldatei auf `abc` gesetzt ist, werden die Protokolle `abc.0`, `abc.1` usw. erzeugt.
- einen Dateityp hat, wird der Index vor dem Dateityp hinzugefügt.

  Wenn die Protokolldatei auf `abc.log` gesetzt ist, werden die Protokolle `abc.0.log`, `abc.1.log` usw. erzeugt.

**Anmerkungen:**

1. Da die Bridges mit einer beliebigen Benutzerberechtigung ausgeführt werden können, ist es nicht möglich, für die Protokolle ein bestimmtes Verzeichnis, z. B. `/var/mqm/qmgrs/<qm>/errors`, zu erzwingen.
2. Die gleichen Informationen werden weiterhin in die Datenströme *stdout* und *stderr* geschrieben.

3. Sobald eine einzelne Protokolldatei erneut geöffnet wird, werden die Informationen zu Basiskonfiguration erneut gedruckt. Die Informationen sind immer verfügbar, statt nur ein einziges Mal beim Start des Programms gedruckt zu werden.

# z/OS Deprecated MQ Adv. VUE IBM MQ Advanced for z/OS VUE für die Verwendung mit Blockchain konfigurieren

Konfigurieren Sie die IBM MQ Bridge to blockchain und führen Sie sie aus, um eine sichere Verbindung zwischen einem IBM MQ auf dem z/OS-Warteschlangenmanager und IBM Blockchain herzustellen. Verwenden Sie die Bridge, um eine asynchrone Verbindung zu einer Ressource in Ihrer Blockchain herzustellen, nach ihr zu suchen und ihren Status zu aktualisieren, indem Sie eine Messaging-Anwendung verwenden, die eine Verbindung zu Ihrem IBM MQ Advanced for z/OS VUE-Warteschlangenmanager herstellt.

## Vorbereitende Schritte

**Anmerkungen:**

- Deprecated IBM MQ Bridge to blockchain ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-341). Blockchain -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.
- Removed V 9.3.2 Für Continuous Deliverywird die IBM MQ Bridge to blockchain unter IBM MQ 9.3.2aus dem Produkt entfernt.
- IBM MQ Bridge to blockchain ist als Teil eines Connector-Packs in IBM MQ Advanced for z/OS Value Unit Edition 9.1.0 verfügbar. Sie können eine Verbindung zu IBM MQ Advanced for z/OS VUE-Warteschlangenmanagern herstellen, die auf der gleichen Befehlsebene oder höher ausgeführt werden.
- IBM MQ Bridge to blockchain wird für die Verwendung mit Ihrem Blockchain-Netz unterstützt, das auf Hyperledger Composer basiert, die auf Hyperledger Fabric erstellt wurden.
- Der IBM MQ Bridge to blockchain muss in einer z/OS UNIX System Services-Umgebung installiert sein und benötigt Version 8 des Java runtime environment von IBM.

## Informationen zu diesem Vorgang

Blockchain ist ein gemeinsam genutzter, verteilter, digitaler Ledger, der aus einer Kette von Blöcken besteht, die sich auf Transaktionen zwischen Peers in einem Netzwerk geeinigt haben. Jeder Block in der Kette ist mit dem vorherigen Block verknüpft, und so weiter zurück zu der ersten Transaktion.

IBM Blockchain basiert auf Hyperledger Fabric und Hyperledger Composer. Sie können damit lokal unter Verwendung von Docker oder in einem Container-Cluster in IBM Cloud entwickeln. Sie können auch Ihr IBM Blockchain-Netz in der Produktion aktivieren und verwenden, um ein Geschäftsnetz mit einem hohen Maß an Sicherheit, Datenschutz und Leistung zu erstellen und zu steuern. Weitere Informationen finden Sie in der IBM Blockchain-Plattform .

Hyperledger Fabric und Hyperledger Composer sind ein Open-Source-Blockchain-Framework für Unternehmen, das von Mitgliedern der Hyperledger Projectgemeinsam entwickelt wird, einschließlich IBM als ursprünglicher Codebeitrag. Hyperledger Project oder Hyperledgerist eine globale, interaktive Linux Foundation Open-Source-Initiative zur Förderung branchenübergreifender Blockchain-Technologien. Weitere Informationen finden Sie unter IBM Blockchain, Hyperledger Projects, Hyperledger Fabric, und Hyperledger Composer.

Wenn Sie bereits IBM MQ Advanced for z/OS VUE und IBM Blockchain verwenden, können Sie IBM MQ Bridge to blockchain verwenden, um Ihr Hyperledger Composer-Geschäftsmodell über die Hyperledger Composer-REST-Schnittstelle zu steuern, sodass Sie den Status in Ihrer Blockchain aktualisieren oder abfragen und Antworten aus Ihrem Blockchain-Netz empfangen können. Auf diese Weise können Sie Ihre lokale IBM-Software mit einem Cloud-Blockchain-Service oder einer lokal verwalteten On-Premise-Lösung integrieren.

Eine kurze Übersicht über den Prozess der Bridge-Bedienung kann in Abbildung 1 angezeigt werden. Eine Benutzeranwendung versetzt eine JSON-formatierte Nachricht in die Eingabe-/Anforderungswarteschlange auf dem z/OS-Warteschlangenmanager. Über den Hyperledger Composer-REST-Server stellt die Bridge eine Verbindung zum Warteschlangenmanager her, ruft die Nachricht aus der Eingabe-/Anforderungswarteschlange ab, prüft, ob das JSON-Format korrekt eingehalten ist, und gibt anschließend die REST-Anforderung an die Blockchain aus. Die von der Blockchain zurückgegebenen Daten werden von der Bridge analysiert und in die Antwortwarteschlange gestellt, wie in der ursprünglichen IBM MQ-Anforderungsnachricht definiert. Die Benutzeranwendung kann eine Verbindung zum WS-Manager herstellen, die Antwortnachricht aus der Antwortwarteschlange abrufen und die Informationen verwenden.

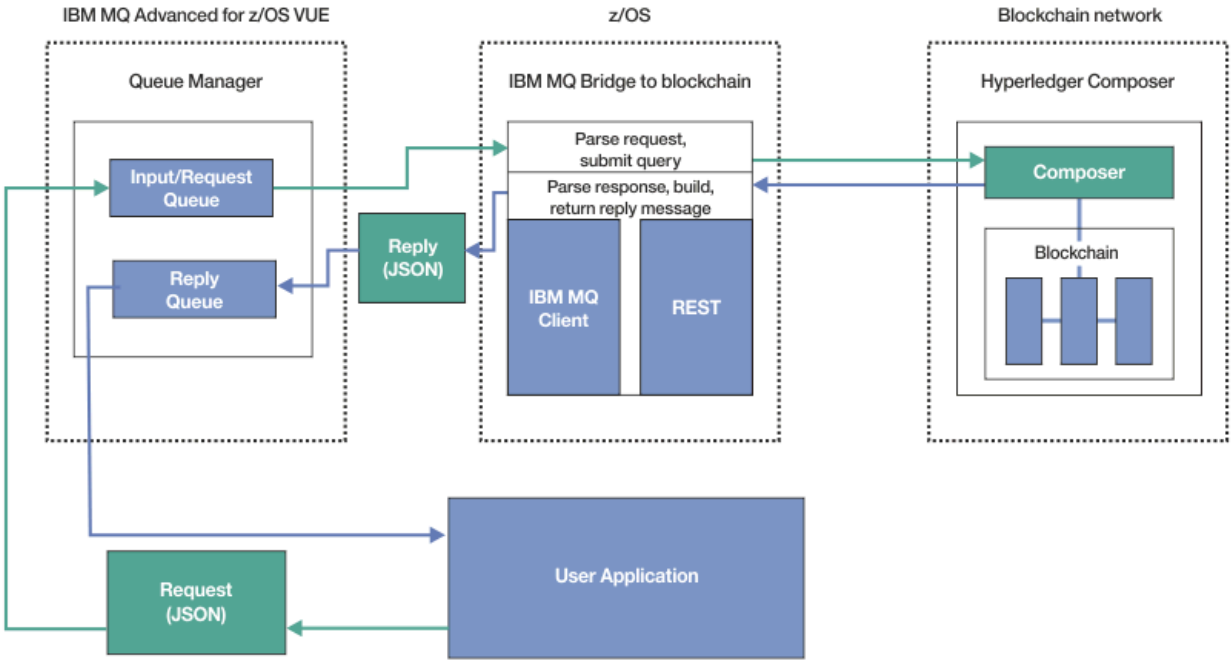



*Abbildung 99. IBM MQ Bridge to blockchain*

Sie müssen die IBM MQ Bridge to blockchain so konfigurieren, dass sie eine Verbindung zu einem Hyperledger Composer-REST-Server und nicht direkt mit der zugrunde liegenden Hyperledger Fabric-Ebene herstellt. Wenn die Bridge aktiv ist, fordert eine Messaging-Anwendung die Bridge auf, die Hyperledger Composer-REST-API auf der Basis des benutzerdefinierten Geschäftsnetzmodells zu steuern, wodurch wiederum die zugrunde liegenden Chaincoderoutinen gesteuert werden, die den Status der Ressource abfragen oder aktualisieren können und die Ergebnisse als Antwort über den Hyperledger Composer-REST-Server an die Messaging-Anwendung zurückgeben.

## Vorgehensweise

Erstellen Sie die Warteschlangen für die Bridge, indem Sie die Beispiel-JCL in `thlqual.SCSQPROC(CSQ4BCBQ)` anpassen und übergeben.

Musterbrückenwarteschlangendefinitionen werden für die standardmäßigen benannten Warteschlangen bereitgestellt, die für Folgendes verwendet werden:

- Nachrichteneingabe an die Bridge: `SYSTEM.BLOCKCHAIN.INPUT.QUEUE` und `APPL1.BLOCKCHAIN.INPUT.QUEUE`
- Antworten aus der Blockchain: `APPL1.BLOCKCHAIN.REPLY.QUEUE`

Unterschiedliche Anwendungen können die gleiche Eingabewarteschlange verwenden, aber Sie können mehrere Antwortwarteschlangen angeben, eine für jede Ihrer Anwendungen. Sie müssen definierte Antwortwarteschlangen nicht verwenden. Wenn Sie dynamische Warteschlangen für Antworten verwenden möchten, müssen Sie ihre Sicherheitskonfiguration in Betracht ziehen.

### Ergebnisse

Sie haben die Warteschlangen erstellt, die die Bridge für die Verarbeitung von Nachrichten aus IBM MQ und Ihrem Blockchain-Netz benötigt.

### Nächste Schritte

Erstellen Sie anhand der Informationen zu Ihrem Warteschlangenmanager und der Berechtigungsnachweise Ihres Blockchain-Netzes eine Konfigurationsdatei für die IBM MQ Bridge to blockchain.

## z/OS Deprecated MQ Adv. VUE Konfigurationsdatei für IBM MQ Bridge to blockchain unter z/OS erstellen

Geben Sie Ihren Warteschlangenmanager und Ihre Blockchain-Netzparameter ein, um die Konfigurationsdatei für den IBM MQ Bridge to blockchain zu erstellen, um eine Verbindung zu Ihren IBM MQ- und IBM Blockchain-Netzen herzustellen.

### Vorbereitende Schritte

**Anmerkungen:**

- Deprecated IBM MQ Bridge to blockchain ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-341). Blockchain -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.
- Removed V 9.3.2 Für Continuous Deliverywird die IBM MQ Bridge to blockchain unter IBM MQ 9.3.2aus dem Produkt entfernt.
- LTS IBM beabsichtigt, die Funktionalität von Long Term Support -Releases in zukünftigen Fixpacks zu entfernen. Wenn Sie über Anwendungen verfügen, die von dieser Änderung betroffen sind, wenden Sie sich an den IBM Support.
- Sie haben Ihr Hyperledger Composer-Blockchain-Netz erstellt und konfiguriert.
- Sie haben die IBM MQ Bridge to blockchain in Ihrer z/OS-Umgebung installiert.
- Sie haben Ihren IBM MQ Advanced for z/OS VUE-Warteschlangenmanager gestartet.

### Informationen zu diesem Vorgang

Diese Task führt Sie durch die minimale Konfiguration, die zum Erstellen der IBM MQ Bridge to blockchain-Konfigurationsdatei erforderlich ist, und eine erfolgreiche Verbindung zu Ihren IBM Blockchain- und IBM MQ-Netzen.

Sie können die Bridge verwenden, um eine Verbindung zu Blockchain-Netzen herzustellen, die auf Hyperledger Composer basieren. Um die Brücke verwenden zu können, benötigen Sie Konfigurationsinformationen aus Ihrem Blockkettennetz. In jedem Schritt in dieser Task finden Sie Beispielkonfigurationsdetails, die auf zwei unterschiedlich konfigurierten Blockkettennetzen basieren:

- Hyperledger Composer-Netz, das in Docker ausgeführt wird. Weitere Informationen finden Sie unter Installieren von Hyperledger Composer und Generieren einer REST-API.
- Hyperledger Composer-Netz, das in einem Kubernetes-Cluster in IBM Cloud ausgeführt wird. Weitere Informationen finden Sie unter Entwickeln in einer Cloud-Sandbox auf IBM Blockchain Platform .

### Vorgehensweise

1. Führen Sie die Bridge in Ihrer z/OS UNIX System Services (z/OS UNIX)-Umgebung aus, um eine Konfigurationsdatei zu erstellen.

   Sie benötigen die Parameter aus Ihren Hyperledger Composer-Sicherheitsinformationen und aus Ihrem IBM MQ Advanced for z/OS VUE-Warteschlangenmanager.

Führen Sie das Bridge-Script aus dem Verzeichnis `mqbc/bin` der Position in z/OS UNIX aus, in der IBM MQ installiert ist.

```
./runmqbcb -o config_file_name.cfg
```

Wie das folgende Beispiel zeigt, werden die vorhandenen Werte in den eckigen Klammern angezeigt. Drücken Sie `Enter` , um vorhandene Werte zu akzeptieren, drücken Sie `Space` dann `Enter` , um die Werte zu löschen, und geben Sie in den Klammern `Enter` ein, um neue Werte hinzuzufügen. Sie können Listen von Werten (z. B. Peers) durch Kommas trennen oder indem Sie jeden Wert in einer neuen Zeile eingeben. Eine leere Zeile beendet die Liste.

**Anmerkung:** Sie können die vorhandenen Werte nicht bearbeiten. Sie können sie behalten, ersetzen oder löschen.

2. Geben Sie Werte für die Verbindung zu Ihrem IBM MQ Advanced for z/OS VUE-Warteschlangenmanager ein.

   Die Mindestwerte, die für die Verbindung benötigt werden, sind der Name des Warteschlangenmanagers und die Namen der von Ihnen definierten Brückeneingabewarteschlangen. Für Verbindungen zu fernen IBM MQ Advanced for z/OS VUE -Warteschlangenmanagern benötigen Sie auch **MQ Channel** und **MQ Conname** (Hostadresse und Port, unter denen der Warteschlangenmanager ausgeführt wird).

   Wenn Sie TLS für die Verbindung zu IBM MQ in Schritt „5” auf Seite 959verwenden möchten, müssen Sie JNDI oder CCDT verwenden und **MQ CCDT URL** oder **JNDI implementation class** und **JNDI provider URL** entsprechend angeben.

   **Anmerkung:** Die Werte für **MQ CCDT** oder **JNDI** haben Vorrang vor der Konfigurationsdatei, in der sich die Werte überschneiden.

```
Connection to Queue Manager
---------------------------
Queue Manager                    : [z/OS_ADV_VUE_qmgr_name]
Bridge Input Queue               : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
MQ Channel                       : []
MQ Conname                       : []
MQ CCDT URL                      : []
JNDI implementation class        : []
JNDI provider URL                : []
MQ Userid                        : []
MQ Password                      : []
```

3. Geben Sie die Berechtigungsnachweise für den Hyperledger Composer-REST-Server ein, der Ihrem Blockchain-Netz zugeordnet ist (falls konfiguriert).

   Im folgenden Beispiel wurde der REST-Server Hyperledger Composer mithilfe des Moduls **passport-ldapauth NodeJS** mit einem LDAP-Speicher für Berechtigungsnachweise konfiguriert. Beachten Sie, dass Sie alle **passport-*** -Module verwenden können, die Basisbenutzer-und Kennwortberechtigungsnachweise auf diese Weise zur Verfügung stellen. Weitere Informationen hierzu finden Sie im Abschnitt Authentifizierung für den REST-Server aktivieren .

```
User Identification
-------------------------------
Userid                          : []admin
Password                        : []******
API path for Login              : auth/ldap
```

4. Geben Sie die Adresse für den Hyperledger Composer-REST-Server ein.

   Beachten Sie, dass in diesem Attribut kein Protokoll, das heißt `http` oder `https`, benötigt wird und dass die Portnummer obligatorisch ist. Ob das HTTP-oder HTTPS-Protokoll verwendet wird, hängt von der Sicherheitskonfiguration des REST-Servers ab. Wenn ein Zertifikat und ein privates Schlüsselpaar für den REST-Server bereitgestellt werden, wird HTTPS verwendet. HTTPS wird verwendet. Andern-

falls wird HTTP verwendet. Informationen zum Angeben des Zertifikats und des privaten Schlüsselpaars finden Sie in Schritt „5” auf Seite 959.

```
REST Server
-------------------------
Address for Composer REST server     : [composer-rest-server-ip-address:3000]
```

5. Geben Sie Zertifikate für TLS-Verbindungen ein.

   Die Brücke fungiert als IBM MQ JMS-Client, der eine Verbindung zu einem Warteschlangenmanager herstellt. Dies bedeutet, dass er so konfiguriert werden kann, dass er die TLS-Sicherheit verwendet, um eine sichere Verbindung herzustellen, wie alle anderen IBM MQ JMS-Clients. Die Konfiguration von TLS-Verbindungsdetails wird nur dann bereitgestellt, wenn Sie in Schritt „2” auf Seite 958 JNDI- oder CCDT-Informationen angegeben haben.

   Die Zertifikatsspeicher werden für Hyperledger Composer und für Ihren IBM MQ Advanced for z/OS VUE-Warteschlangenmanager verwendet. Wenn Zertifikatsspeicher angegeben sind, versucht die Bridge stets über HTTPS eine Verbindung zum Hyperledger-REST-Server herzustellen. TLS kann jedoch mit der folgenden Option für IBM MQ-Verbindungen inaktiviert werden, während TLS für Hyperledger Composer weiterhin verwendet wird.

```
Certificate stores for TLS connections
--------------------------------------
Personal keystore                    : []
Keystore password                    : []
Trusted store for signer certs       : []
Trusted store password               : []
Use TLS for MQ connection            : [N]
Timeout for Blockchain operations    : [12]
```

   Weitere Informationen finden Sie im Abschnitt REST-Server mit HTTPS und TLS sichern .

6. Optional: Geben Sie die Position für die Protokolldatei für den IBM MQ Bridge to blockchain ein.

   Sie können den Namen und die Position der Protokolldatei in der Konfigurationsdatei oder in der Befehlszeile angeben.

```
Behavior of bridge program
--------------------------
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

### Ergebnisse

Sie haben die Konfigurationsdatei erstellt, die der IBM MQ Bridge to blockchain verwendet, um eine Verbindung zu Ihrem IBM Blockchain-Netz und zu Ihrem IBM MQ Advanced for z/OS VUE-Warteschlangenmanager herzustellen.

### Nächste Schritte

Führen Sie die in „IBM MQ Bridge to blockchain unter z/OS ausführen” auf Seite 960 beschriebenen Schritte aus.

## z/OS Deprecated MQ Adv. VUE IBM MQ -Sicherheitskonfiguration für IBM MQ Bridge to blockchain unter z/OS

Hinweise zur Konfiguration der IBM MQ-Sicherheit mit der IBM MQ Bridge to blockchain.

**Anmerkungen:**

- Deprecated IBM MQ Bridge to blockchain ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-341). Blockchain -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.

- Removed V 9.3.2 Für Continuous Deliverywird die IBM MQ Bridge to blockchain unter IBM MQ 9.3.2aus dem Produkt entfernt.

Die folgenden Beispiele zeigen RACF-Definitionen, die verwendet werden können, um der IBM MQ Bridge to blockchain Zugriff auf die von ihr benötigten Warteschlangen zu ermöglichen. Die Definitionen gehen davon aus, dass die Brücke unter der MQBCBUSR-Benutzer-ID ausgeführt wird.

Darüber hinaus muss der IBM MQ Bridge to blockchain mit einer der folgenden Methoden Zugriffsrecht für die Verbindung mit dem Warteschlangenmanager erteilt werden:

- Direkte Verwendung des Bindungsmodus; siehe Verbindungssicherheitsprofile für Stapelverbindungen oder
- Verwenden eines Clientmodus über CHINIT; siehe Client-MQI-Anforderungen

### Berechtigung für IBM MQ Bridge to blockchain-Anforderungswarteschlange

Setzen Sie die folgenden RACF-Befehle ab, um den Zugriff der MQBCBUSR-Benutzer-ID auf den Empfang von Nachrichten aus der Standardanforderungswarteschlange SYSTEM.BLOCKCHAIN.INPUT.QUEUE zu erteilen:

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
```

### Berechtigung für IBM MQ Bridge to blockchain-Antwortwarteschlange

Setzen Sie die folgenden RACF-Befehle ab, um den Zugriff der MQBCBUSR-Benutzer-ID auf das Senden von Nachrichten an die Datei APPL1.BLOCKCHAIN.REPLY.QUEUE zu erteilen. Dieser Warteschlangenname wird in der Antwort auf den Warteschlangennamen in der Anforderungsnachricht angegeben:

```
RDEFINE MQQUEUE APPL1.BLOCKCHAIN.REPLY.QUEUE UACC(NONE)
PERMIT APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
PERMIT CONTEXT.APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQADMIN) ID(MQBCBUSR) ACCESS(UPDATE)
```

**Zugehörige Konzepte**

Profile für die Warteschlangensicherheit

**Zugehörige Tasks**

„IBM MQ Bridge to blockchain -Clientbeispiel unter z/OS ausführen” auf Seite 952

Sie können das JMS-Clientbeispiel verwenden, das mit IBM MQ Bridge to blockchain bereitgestellt wird, um eine Nachricht in die Eingabewarteschlange zu stellen, die von der Blockchain-Bridge überprüft wird, und die empfangene Antwort anzuzeigen. Dieses Beispiel basiert auf der Verwendung der IBM MQ Bridge to blockchain-Integration in das Hyperledger Composer Trader-Netzwerk-Beispiel.

**Zugehörige Verweise**

API-Ressourcenzugriffsschutz-Kurzreferenz

## z/OS Deprecated MQ Adv. VUE IBM MQ Bridge to blockchain unter z/OS ausführen

Führen Sie die IBM MQ Bridge to blockchain aus, um eine Verbindung zu IBM Blockchain und IBM MQ herzustellen. Wenn sie verbunden ist, ist die Bridge bereit, Anforderungsnachrichten zu verarbeiten, sie an Ihr Hyperledger Composer-Blockkettennetz zu senden und die Antworten zu empfangen und zu verarbeiten.

### Vorbereitende Schritte

**Anmerkungen:**

- Deprecated IBM MQ Bridge to blockchain ist in allen Releases ab 22. November 2022 veraltet (siehe US-Ankündigungsschreiben 222-341). Blockchain -Konnektivität kann mit IBM App Connect oder über App Connect -Funktionen erreicht werden, die mit IBM Cloud Pak for Integrationverfügbar sind.
- Removed V 9.3.2 Für Continuous Deliverywird die IBM MQ Bridge to blockchain unter IBM MQ 9.3.2aus dem Produkt entfernt.

## Informationen zu diesem Vorgang

Verwenden Sie die Konfigurationsdatei, die Sie in der vorherigen Task erstellt haben, um die IBM MQ Bridge to blockchain auszuführen.

## Vorgehensweise

1. Starten Sie den IBM MQ Advanced for z/OS VUE-Warteschlangenmanager, den Sie mit der Bridge verwenden möchten.
2. Starten Sie IBM MQ Bridge to blockchain, um eine Verbindung zu Ihrem Blockchain-Netz herzustellen, und starten Sie den IBM MQ Advanced for z/OS VUE-Warteschlangenmanager.

   Entweder:

   a) Führen Sie die Bridge direkt in z/OS UNIX System Services (z/OS UNIX) aus dem Verzeichnis `mqbc/bin` in der z/OS UNIX-Position aus, in der IBM MQ installiert ist.

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
   ```

   oder

   b) b. Führen Sie die Bridge auf Ihrem z/OS-System aus, indem Sie die in `thlqual.SCSQPROC(CSQ4BCB)` bereitgestellte Beispiel-JCL verwenden.

   Sie müssen eine Reihe von Aktualisierungen für die JCL vornehmen, die speziell für Ihre Umgebung gilt:

   - Ersetzen Sie ++THLQUAL++ durch das übergeordnete Qualifikationsmerkmal der IBM MQ-Zielbibliotheksdateien.
   - Ersetzen Sie ++LANGLETTER++ durch den Buchstaben für die Sprache, in der Nachrichten angezeigt werden sollen.
   - Ersetzen Sie ++PATHPREFIX++ durch den Installationspfad der z/OS UNIX-Komponenten.
   - Ersetzen Sie ++CONFIGFILE++ durch den Pfad zu einer Konfigurationsdatei, die mit dem Befehl `runmqbc -o <file>` im Verzeichnis z/OS UNIX erstellt wurde.
   - Ersetzen Sie ++JAVAHOME++ durch die Position einer 64-Bit Java Virtual Machine (JVM), die unter Java 8 oder höher ausgeführt wird.

   Wenn die Brücke verbunden ist, wird die Ausgabe ähnlich der folgenden zurückgegeben:

   ```
   2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
   5724-H72 (C) Copyright IBM Corp. 2017, 2024.

   2018-05-17 14:28:19.331 BST Ready to process input messages.
   ```

3. Optional: Beheben Sie Fehler bei den Verbindungen zu Ihrem IBM MQ Advanced for z/OS VUE-Warteschlangenmanager und zu Ihrem Blockchain-Netz, wenn die Nachrichten, die nach der Ausführung der Bridge zurückgegeben werden, darauf hinweisen, dass eine Verbindung nicht erfolgreich hergestellt wurde.

   a) Geben Sie den Befehl im Debugmodus mit der Debugoption 1 aus.

   ```
   ./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFi
   le.log -d 1
   ```

Die Brückenschritte über die Verbindung sind aufgebaut und zeigen die Verarbeitungsnachrichten im Modus 'terse' (terse) an.

b) Geben Sie den Befehl im Debugmodus mit der Debugoption 2 aus.

```
./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFi
le.log -d 2
```

Die Brückenschritte über die Verbindung werden aufgebaut und die Verarbeitungsnachrichten werden im ausführlichen Modus angezeigt. Die vollständige Ausgabe wird in Ihre Protokolldatei geschrieben.

Beachten Sie, dass Sie optional auch die Debugmodusoptionen in der JCL angeben können, indem Sie '`-d 0`' in '`-d 1`' oder '`-d 2`' ändern.

## Ergebnisse

Sie haben die IBM MQ Bridge to blockchain gestartet und sich mit Ihrem Warteschlangenmanager und dem Blockchain-Netz verbunden.

## Nächste Schritte

- Führen Sie die Schritte in „IBM MQ Bridge to blockchain -Clientbeispiel unter z/OS ausführen” auf Seite 952 aus, um eine Abfrage oder eine Aktualisierungsnachricht an Ihr Blockchain-Netz zu formatieren und zu senden.
- Verwenden Sie die *MQBCB_EXTRA_JAVA_OPTIONS*-Variable, um JVM-Eigenschaften zu übergeben, z. B. um die Tracefunktion von IBM MQ zu aktivieren. Weitere Informationen hierzu finden Sie im Abschnitt Traceverarbeitung für die IBM MQ Bridge to blockchain.

# z/OS Deprecated Nachrichtenformate für IBM MQ Bridge to blockchain vor IBM MQ 9.2.0 unter z/OS

Informationen zur Formatierung der Nachrichten, die von der IBM MQ Bridge to blockchain gesendet und empfangen werden.

**Achtung:** Das vorhandene Format für die Nachrichtenformate ist veraltet. Wenn Sie über ein Hyperledger Fabric-Netz verfügen, verwenden Sie das Format der in „Nachrichtenformate für die IBM MQ Bridge to blockchain ab IBM MQ 9.2.0” auf Seite 947 beschriebenen Nachrichten aus IBM MQ 9.2.0.

Eine Anwendung fordert, dass die IBM MQ Bridge to blockchain die über Hyperledger Composer definierte REST-API ansteuert, um Informationen zu bearbeiten, die in der Blockchain gespeichert sind. Die Anwendung führt dies durch die Anforderung einer Anforderungsnachricht in die Brückenanforderungswarteschlange durch. Die Ergebnisse der REST-Anforderung werden von der Bridge in eine Antwortnachricht formatiert. Die Bridge verwendet Informationen, die in den Feldern **ReplyToQ** und **ReplyToQMgr** aus dem MQMD der Anforderungsnachricht als Ziel für die Antwortnachricht enthalten sind.

Die Anforderungs- und Antwortnachrichten sind Textnachrichten (MQSTR) im JSON-Format.

## Anforderungsnachrichtenformat

Anforderungsnachrichten enthalten drei Attribute:

**Methode**
Das REST-Verb, das zum Aufrufen der REST-API von Hyperledger Composer verwendet wird, z. B. POST, DELETE oder GET

**path**
Der Pfad zur Hyperledger Composer-REST-API. Dies wird der Basis-Server-URL hinzugefügt. Der Pfad muss mit "`api/`" beginnen.

**Hauptteil**
Der methodenspezifische Inhalt. Dies ist häufig eine JSON-Struktur.

Im folgenden Beispiel wird die Methode POST zum Pfad `api/Trader`verwendet, um ein neues Trader-Objekt zu erstellen. Der Hauptteil gibt die Trader-Klasse an, wie sie vom Hyperledger Composer-Modell des Benutzers definiert ist, und gibt außerdem die zusätzlichen Werte an, die erforderlich sind, um ein neues Trader-Objekt innerhalb des Blockkettennetzwerks zu erstellen.

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class" : "org.example.trading",
    "tradeId" : "Trader2",
    "firstName": "Jane",
    "lastName" : "Doe"
```

## Antwortnachrichtenformat

Antwortnachrichten haben ihre Korrelations-ID auf die Nachrichten-ID der eingehenden Nachricht gesetzt. Alle vom Benutzer definierten Eigenschaften werden aus der Anforderungsnachricht in die Antwortnachricht kopiert. Die Benutzer-ID in der Antwort wird auf die Benutzer-ID des Erstellers gesetzt.

Der **statusCode** ist ein HTTP-Statuscode. Wenn der Fehler von IBM MQ oder von der Brücke stammt, wird ein geeigneter **statusCode** verwendet.

**statusType** ist eine Zeichenfolge, entweder *SUCCESS* oder *FAILURE*.

Für erfolgreiche Anforderungen enthält das Element "**data**" in der Antwortnachricht die Antwort aus der aufgerufenen Hyperledger Composer-REST-API.

Ein Beispiel für eine erfolgreiche Verarbeitung:

```
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "data": [
      {
         "$class": "org.example.trading",
         "firstName": "John",
         "lastName": "Doe",
         "tradeId": "Trader1"
      },
      {
         "$class": "org.example.trading",
         "firstName": "Jane",
         "lastName": "Doe",
         "tradeId": "Trader2"
      }
   ]
}
```

Alle Fehlerantworten haben die gleichen Felder, unabhängig davon, ob sie von der Brücke selbst generiert werden, von den Aufrufen an den Hyperledger Composer-REST-Server, von der Blockchain oder vom Chaincode-Aufruf. For example:

- Falsche JSON-Eingabenachricht

```
{
   "statusCode": 400,
   "statusType": "FAILURE",
   "message": "[AMQBC021E] Error: Cannot parse input message or there are
    missing fields in the message. Missing fields appear to be: "method"."
}
```

- Anforderung, die vom Hyperledger Composer-REST-Server nicht verarbeitet werden konnte

```
{
   "statusCode": 500,
   "statusType": "FAILURE",
   "message": "Error trying to invoke business network. Error: No valid responses
```

```
    from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
    error (status: 500, message: Error: Failed to add object with ID 'Trader1'
    as the object already exists)"
}
```

Anwendungen können feststellen, ob die Anforderung erfolgreich war oder fehlgeschlagen ist, indem sie entweder die Zeichenfolge **statusType** oder das Vorhandensein des Datenfelds betrachtet. Wenn bei der Verarbeitung der Eingabenachricht ein Fehler auftritt und die Brücke sie nicht an die Blockkette sendet, ist der von der Brücke zurückgegebene Wert ein MQRC-Wert (normalerweise **MQRC_FORMAT_ERROR**).

## z/OS Deprecated IBM MQ Bridge to blockchain -Clientbeispiel unter z/OS ausführen

Sie können das JMS-Clientbeispiel verwenden, das mit IBM MQ Bridge to blockchain bereitgestellt wird, um eine Nachricht in die Eingabewarteschlange zu stellen, die von der Blockchain-Bridge überprüft wird, und die empfangene Antwort anzuzeigen. Dieses Beispiel basiert auf der Verwendung der IBM MQ Bridge to blockchain-Integration in das Hyperledger Composer Trader-Netzwerk-Beispiel.

### Vorbereitende Schritte

Weitere Informationen finden Sie unter /trade_network .

Ihr IBM MQ Bridge to blockchain wird ausgeführt und ist mit Ihrem IBM MQ Advanced-oder IBM MQ Advanced for z/OS VUE-Warteschlangenmanager sowie mit Ihrem Blockchain-Netzwerk verbunden.

### Informationen zu diesem Vorgang

Suchen Sie die JMS-Beispielanwendung (`ComposerBCBSamp.java`) im Verzeichnis `samp` der IBM MQ Bridge to blockchain.

Beispiel: `<MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java`, wobei `<MQ_INSTALL_ROOT>` die folgende ist:

- Linux Das Verzeichnis, in dem IBM MQ installiert ist.
- z/OS Das z/OS UNIX System Services-Verzeichnis, in dem Sie z/OS UNIX-Komponenten von IBM MQ installiert sind.

.

### Vorgehensweise

1. Bearbeiten Sie die Java-Quellendatei des Clients.

   Befolgen Sie die Anweisungen im Beispiel, um ihn so zu konfigurieren, dass er mit Ihrer IBM MQ-Umgebung und Ihrem Blockchain-Netz übereinstimmt.

   Der folgende Code aus dem Beispiel definiert drei JSON-Anforderungsnachrichten, die an die Bridge gesendet werden sollen:

   a. Erstens, um ein vorhandenes '`commodity`' zu entfernen
   b. Zweitens, um eine neue '`commodity`', '`owner`' und zugehörige Werte zu erstellen,
   c. Schließlich werden die neuen Informationen zum '`commodity`' nach den vorherigen beiden Anforderungsnachrichten angezeigt.

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
```

```
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
  }
```

2. Kompilieren Sie das Beispiel.

   Verweisen Sie auf die IBM MQ-Clientklassen und die JSON4J.jar-Datei, die im Bridge-Verzeichnis geliefert werden.

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar Compo
serBCBClient.java
```

3. Führen Sie die kompilierte Klasse aus.

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:. Compo
serBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
  {"method":"DELETE"," path ":"api\/Commodity\/BC"}
Response text:
{
   "statusCode": 204,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": ""
}
SUCCESS
Sent message:
{"body":{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"descripti
on":"Blockchain Sample Description","mainExchange":"My Exchange","tradingSymbol":"BC"},"ope
ration":"POST","url":"Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": {
```

```
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "Me",
      "quantity": 100,
      "tradingSymbol": "BC"
   }
}
SUCCESS
Sent message:
  {"method":"GET","path":"api\/Commodity"}
Response text:
{
   "statusCode": 200,
   "statusType": "SUCCESS",
   "message": "OK",
   "data": [
      {
         "$class": "org.example.trading.Commodity",
         "description": "Blockchain Sample Description",
         "mainExchange": "My Exchange",
         "owner": "resource:org.example.trading.Trader#Me",
         "quantity": 100,
         "tradingSymbol": "BC"
      }
   ]
}
SUCCESS
```

Das Feld **message** enthält entweder "0K" für eine erfolgreich verarbeitete Nachricht oder im Falle einer fehlgeschlagenen Anforderung Informationen zur Ursache des Fehlers.

Wenn der Client eine Zeitlimitüberschreitung beim Warten auf die Antwort empfängt, überprüfen Sie, ob die Brücke aktiv ist.

# z/OS Warteschlangenmanager unter z/OS erstellen

Verwenden Sie diese Anweisungen zum Konfigurieren von Warteschlangenmanagern unter IBM MQ for z/OS.

## Vorbereitende Schritte

Lesen Sie vor der Konfiguration von IBM MQ for z/OSdie folgenden Informationen:

- IBM MQ for z/OS -Konzepte
- IBM MQ -Umgebung unter z/OSplanen

## Informationen zu diesem Vorgang

Nach der Installation von IBM MQ müssen Sie eine Reihe von Tasks ausführen, bevor Sie das Programm den Benutzern zur Verfügung stellen können.

## Prozedur

- Beachten Sie die Informationen in den folgenden Unterabschnitten zur Konfiguration von Warteschlangenmanagern unter IBM MQ for z/OS.

**Zugehörige Konzepte**

z/OS Quellen, aus denen Sie MQSC-und PCF-Befehle unter IBM MQ for z/OS ausgeben können

**Zugehörige Tasks**

„Warteschlangenmanager auf Multiplatforms erstellen” auf Seite 7
Bevor Sie Nachrichten und Warteschlangen verwenden können, müssen Sie mindestens einen WS-Manager und die zugehörigen Objekte erstellen und starten. Ein Warteschlangenmanager verwaltet die Ressourcen, die ihm zugeordnet sind, insbesondere die Warteschlangen, die er besitzt. Er stellt Warteschlangenservices für Anwendungen für MQI-Aufrufe (Message Queuing Interface) und Befehle zum Erstellen, Ändern, Anzeigen und Löschen von IBM MQ-Objekten bereit.

Sicherung

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

„Verbindungen zwischen Client und Server konfigurieren” auf Seite 15
Um die Kommunikationsverbindungen zwischen IBM MQ MQI clients und den Servern zu konfigurieren, müssen Sie das Kommunikationsprotokoll festlegen, die Verbindungen an beiden Enden der Verbindung definieren, einen Listener starten und Kanäle definieren.

z/OS IBM MQ for z/OS verwalten

Planung

**Zugehörige Verweise**

z/OS IBM MQ for z/OS-Dienstprogramme verwenden

# z/OS Vorbereiten der Anpassung von Warteschlangenmanagern unter z/OS

Verwenden Sie dieses Thema, wenn Sie Ihre Warteschlangenmanager mit Details zu installierbaren Features, landessprachlichen Features und Informationen zu Tests anpassen und die Sicherheit konfigurieren.

## Anpassung vorbereiten

Das Programmverzeichnis listet den Inhalt des IBM MQ-Installationsbands, die Programm- und Service-Level-Informationen für IBM MQ auf und beschreibt, wie IBM MQ for z/OS mit Hilfe von System Modification Program Extended (SMP/E) installiert werden kann. Download-Links für die Programmverzeichnisse finden Sie unter IBM MQ for z/OS Program Directory PDF files.

Nach der Installation von IBM MQ müssen Sie einige Tasks ausführen, bevor Sie das Programm den Benutzern zur Verfügung stellen können. Eine Beschreibung zu diesen Tasks finden Sie in den folgenden Abschnitten:

- „IBM MQ for z/OS einrichten” auf Seite 972
- „Warteschlangenmanager auf z/OS testen” auf Seite 1043
- Sicherheit unter z/OS einrichten

Wenn Sie von einer früheren Version von IBM MQ for z/OS migrieren, müssen Sie fast alle Anpassungstasks ausführen. Weitere Informationen zu den Tasks, die Sie ausführen müssen, finden Sie unter Verwalten und Migrieren .

### Installierbare Funktionen von IBM MQ for z/OS

IBM MQ for z/OS umfasst die folgenden Funktionen:

**Basis**
Dies ist erforderlich; es umfasst alle Hauptfunktionen, einschließlich der folgenden:

- Verwaltung und Dienstprogramme
- Unterstützung für CICS, IMS und Stapelanwendungen, die die IBM MQ-Anwendungsprogrammierschnittstelle oder C++ verwenden.
- Verteilte Warteschlangenfunktion (Unterstützung für TCP/IP-und APPC-Kommunikation)

**Funktionen in der Landessprache**
Diese enthalten Fehlermeldungen und Anzeigen in allen unterstützten Landessprachen. Jedem Sprache ist ein Sprachbrief zugeordnet. Die Sprachen und Buchstaben sind:

**C**
Vereinfachtes Chinesisch

**E**
U.S. Englisch (Groß-/Kleinschreibung)

**F**
Französisch

**K**
Japanisch

**U**
U.S. Englisch (Großschreibung)

Sie müssen die Option 'US English (mixed case)' installieren. Sie können auch eine oder mehrere andere Sprachen installieren. (Der Installationsprozess für andere Sprachen setzt voraus, dass amerikanisches Englisch (Groß-/Kleinschreibung) installiert ist, selbst wenn Sie amerikanisches Englisch (Groß-/Kleinschreibung) nicht verwenden.)

**IBM MQ for z/OS UNIX System Services Components**
Diese Funktion ist optional. Wählen Sie diese Funktion aus, wenn Sie Java -Anwendungen, die Jakarta Messaging 3.0 oder Java Message Service 2.0 verwenden, erstellen und ausführen möchten, um eine Verbindung zu IBM MQ for z/OSherzustellen.

Informationen zur Installation von IBM MQ for z/OS UNIX System Services Componentsfinden Sie unter PDF-Dateien für dasIBM MQ for z/OS -Programmverzeichnis .

**IBM MQ for z/OS UNIX System Services Web Components**
Diese Funktion ist optional.

Wählen Sie diese Funktion aus, wenn Sie die IBM MQ Console oder die REST API verwenden möchten.

Sie müssen die IBM MQ for z/OS UNIX System Services Components-Komponente installieren, um diese Funktion zu installieren.

**IBM MQ for z/OS Managed File Transfer**
Diese Funktion ist optional und sollte nur installiert werden, wenn Sie über eine Berechtigung für IBM MQ Advanced for z/OS, IBM MQ for z/OS Value Unit Edition (VUE) oder IBM MQ for z/OS Managed File Transfer verfügen.

Wählen Sie diese Funktion aus, wenn Sie die Managed File Transfer-Funktionen von IBM MQ for z/OS verwenden möchten.

Sie müssen die IBM MQ for z/OS UNIX System Services Components-Komponente installieren, um diese Funktion zu installieren.

**Bibliotheken, die nach der Installation vorhanden sind**

IBM MQ enthält eine Reihe separater Ladebibliotheken. Tabelle 53 auf Seite 968 zeigt die Bibliotheken, die nach der Installation von IBM MQ möglicherweise vorhanden sind.

*Tabelle 53. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind*

| Name | Beschreibung |
|---|---|
| thlqual.SCSQANLC | Enthält die Lademodule für die Version von IBM MQ in vereinfachtem Chinesisch. |
| thlqual.SCSQANLE | Enthält die Lademodule für die USA. Englische Version (Groß-/Kleinschreibung) von IBM MQ. |
| thlqual.SCSQANLF | Enthält die Lademodule für die Version von IBM MQ in Französisch. |
| thlqual.SCSQANLK | Enthält die Lademodule für die Version von IBM MQ in Japanisch. |
| thlqual.SCSQANLU | Enthält die Lademodule für die USA. Englische Version (Großschreibung) von IBM MQ. |

*Tabelle 53. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind (Forts.)*

| Name | Beschreibung |
|---|---|
| thlqual.SCSQASMS | Enthält Quelle für Assembler-Beispielprogramme. |
| thlqual.SCSQAUTH | Haupt-Repository für alle IBM MQ-Produktlademodule. Es enthält das Standardparametermodul CSQZPARM. Diese Bibliothek muss APF-berechtigt und im PDS-E-Format sein. |
| thlqual.SCSQCICS | Enthält zusätzliche Lademodule, die in die CICS-Programmbibliotheksverkettung miteinbezogen werden müssen. Diese Bibliothek muss APF-berechtigt und im PDS-E-Format sein. |
| thlqual.SCSQCLST | Enthält die von den Musterprogrammen verwendeten CLISTs. |
| thlqual.SCSQCOBC | Enthält COBOL-Copybooks, einschließlich Copybooks, die für die Beispielprogramme erforderlich sind. |
| thlqual.SCSQCOBS | Enthält Quelle für COBOL-Beispielprogramme. |
| thlqual.SCSQCPPS | Enthält Quelle für C + + -Musterprogramme. |
| thlqual.SCSQC37S | Enthält Quelle für C-Beispielprogramme. |
| thlqual.SCSQC370 | Enthält C-Header, einschließlich Header, die für die Beispielprogramme erforderlich sind. |
| thlqual.SCSQDEFS | Enthält die Nebendefinitionen für C++ und die Db2-DBRMs für gemeinsam genutzte Warteschlangen. |
| thlqual.SCSQEXEC | Enthält ausführbare REXX-Dateien, die in die SYSEXEC-oder SYSPROC-Verkettung eingeschlossen werden sollen, wenn Sie die für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQFCMD | Enthält Vorlagen für Jobs, mit denen Managed File Transfer-Tasks erstellt und ausgeführt werden können. |
| thlqual.SCSQHPPS | Enthält Headerdateien für C + +. |
| thlqual.SCSQINST | Enthält JCL für Installationsjobs. |
| thlqual.SCSQLINK | Frühe Code-Bibliothek. Enthält die Lademodule, die beim Systemeinleitungsprogramm geladen werden (IPL). Die Bibliothek muss APF-berechtigt sein. |
| thlqual.SCSQLOAD | Bibliothek laden. Enthält Lademodule für Nicht-APF-Code, Benutzerexits, Dienstprogramme, Beispiele, Installationsprüfprogramme und Adapterstubs. Die Bibliothek muss nicht APF-autorisiert sein und muss nicht in der Linkliste enthalten sein. Diese Bibliothek muss sich im PDS-E-Format befinden. |
| thlqual.SCSQMACS | Enthält Assemblermakros mit folgenden Makros: Beispielmakros, Produktmakros und Systemparametermakros. |
| thlqual.SCSQMAPS | Enthält CICS-Kartengruppen, die von den Beispielprogrammen verwendet werden. |
| thlqual.SCSQMSGC | Enthält ISPF-Nachrichten, die in die ISPMLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Vereinfachtes Chinesisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |

*Tabelle 53. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind (Forts.)*

| Name | Beschreibung |
|---|---|
| thlqual.SCSQMSGE | Enthält ISPF-Nachrichten, die in die ISPMLIB-Verkettung eingeschlossen werden sollen, wenn Sie die USA-Version verwenden. Sprachkomponente Englisch (Groß-/Kleinschreibung) für die IBM MQ-Operationen und -Steuerkonsolen. |
| thlqual.SCSQMSGF | Enthält ISPF-Nachrichten, die in die ISPMLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Französisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQMSGK | Enthält ISPF-Nachrichten, die in die ISPMLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Japanisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQMSGU | Enthält ISPF-Nachrichten, die in die ISPMLIB-Verkettung eingeschlossen werden sollen, wenn Sie die USA-Version verwenden. Die englische (Großschreibung) Sprachfunktion für die IBM MQ-Operationen und -Steuerkonsolen. |
| thlqual.SCSQMVR1 | Enthält die Lademodule für verteilte Warteschlangensteuerung. Diese Bibliothek muss APF-berechtigt und im PDS-E-Format sein. |
| thlqual.SCSQPLIC | Enthält PL/I-Dateien. |
| thlqual.SCSQPLIS | Enthält Quelle für PL/I-Beispielprogramme. |
| thlqual.SCSQPNLA | Enthält IPCS-Anzeigen für das Formatierungsprogramm für Speicherauszüge, die in die ISPPLIB-Verkettung eingeschlossen werden sollen. Enthält zudem Fenster für IBM MQ-Beispielprogramme. |
| thlqual.SCSQPNLC | Enthält ISPF-Fenster, die in die ISPPLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Vereinfachtes Chinesisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQPNLE | Enthält ISPF-Anzeigen, die in die ISPPLIB-Verkettung eingeschlossen werden sollen, wenn Sie die USA-Version verwenden. Sprachkomponente Englisch (Groß-/Kleinschreibung) für die IBM MQ-Operationen und -Steuerkonsolen. |
| thlqual.SCSQPNLF | Enthält ISPF-Fenster, die in die ISPPLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Französisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQPNLK | Enthält ISPF-Fenster, die in die ISPPLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Japanisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQPNLU | Enthält ISPF-Anzeigen, die in die ISPPLIB-Verkettung eingeschlossen werden sollen, wenn Sie die USA-Version verwenden. Die englische (Großschreibung) Sprachfunktion für die IBM MQ-Operationen und -Steuerkonsolen. |
| thlqual.SCSQPROC | Enthält Beispiele für JCL-und Standardsysteminitialisierungsdatensätze. |

| *Tabelle 53. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind (Forts.)* | |
|---|---|
| **Name** | **Beschreibung** |
| thlqual.SCSQSNLC | Enthält die Lademodule für die Module von IBM MQ in vereinfachtem Chinesisch, die für eine Sonderfunktion erforderlich sind (z. B. den frühen Code). |
| thlqual.SCSQSNLE | Enthält die Lademodule für die USA. Englische (Groß-/Kleinschreibung) Versionen der IBM MQ-Module, die für eine spezielle Funktion erforderlich sind (z. B. der vorzeitige Code). |
| thlqual.SCSQSNLF | Enthält die Lademodule für die Module von IBM MQ in Französisch, die für eine Sonderfunktion erforderlich sind (z. B. den frühen Code). |
| thlqual.SCSQSNLK | Enthält die Lademodule für die Module von IBM MQ in Japanisch, die für eine Sonderfunktion erforderlich sind (z. B. den frühen Code). |
| thlqual.SCSQSNLU | Enthält die Lademodule für die USA. Englische (Großschreibung) Versionen der IBM MQ-Module, die für eine spezielle Funktion erforderlich sind (z. B. der frühe Code). |
| thlqual.SCSQTBLC | Enthält ISPF-Tabellen, die in die ISPTLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Vereinfachtes Chinesisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQTBLE | Enthält ISPF-Tabellen, die in die ISPTLIB-Verkettung eingeschlossen werden sollen, wenn Sie die USA-Version verwenden. Sprachkomponente Englisch (Groß-/Kleinschreibung) für die IBM MQ-Operationen und -Steuerkonsolen. |
| thlqual.SCSQTBLF | Enthält ISPF-Tabellen, die in die ISPTLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Französisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQTBLK | Enthält ISPF-Tabellen, die in die ISPTLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Japanisch' für die Operationen und Bedienfelder von IBM MQ verwenden. |
| thlqual.SCSQTBLU | Enthält ISPF-Tabellen, die in die ISPTLIB-Verkettung eingeschlossen werden sollen, wenn Sie die USA-Version verwenden. Die englische (Großschreibung) Sprachfunktion für die IBM MQ-Operationen und -Steuerkonsolen. |

**Anmerkung:** Ändern Sie keine dieser Bibliotheken oder passen Sie sie nicht an. Wenn Sie Änderungen vornehmen möchten, kopieren Sie die Bibliotheken und nehmen Sie die Änderungen an den Kopien vor.

**Zugehörige Konzepte**

IBM MQ for z/OS - Konzepte

„IBM MQ mit IMS verwenden” auf Seite 1084

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die es IBM MQ ermöglichen, mit IMS zu interagieren.

„IBM MQ mit CICS verwenden” auf Seite 1093

Um IBM MQ mit CICS verwenden zu können, müssen Sie den IBM MQ CICS-Adapter und optional die Komponenten der IBM MQ CICS bridge konfigurieren.

„OTMA-Exits in IMS verwenden” auf Seite 1096

Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

**Zugehörige Tasks**

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten" auf Seite 1052
In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

IBM MQ for z/OS verwalten

**Zugehörige Verweise**

„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen" auf Seite 1094
Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

# z/OS IBM MQ for z/OS einrichten

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

Am besten können Sie einen WS-Manager konfigurieren, indem Sie die folgenden Schritte in der angegebenen Reihenfolge ausführen:

1. Konfigurieren Sie den Basiswarteschlangenmanager.
2. Konfigurieren Sie den Kanalinitiator, der Warteschlangenmanager für die Kommunikation zwischen den WS-Managern und die Kommunikation mit der fernen Clientanwendung ausführt.
3. Wenn Sie Nachrichten verschlüsseln oder schützen möchten, konfigurieren Sie Advanced Message Security for z/OS.
4. Wenn Sie IBM MQ zum Übertragen von Dateien verwenden möchten, konfigurieren Sie Managed File Transfer for z/OS.
5. Wenn Sie die Verwaltungs- oder Messaging-REST API oder die IBM MQ Console verwenden möchten, um IBM MQ über einen Web-Browser zu verwalten, konfigurieren Sie den mqweb-Server.

Dieses Thema führt Sie durch die verschiedenen Phasen der Konfiguration von IBM MQ nach der erfolgreichen Installation. Der Installationsprozess wird im Programmverzeichnis beschrieben. Download-Links für die Programmverzeichnisse finden Sie unter IBM MQ for z/OS Program Directory PDF files.

Zusammen mit IBM MQ wird Mustercode bereitgestellt, der die Anpassung erleichtern soll. Die Beispieldateinamen haben Namen, die mit den vier Zeichen CSQ4 beginnen und sich in der Bibliothek thlqual.SCSQPROC befinden.

Vor Ausführung der in diesem Abschnitt beschriebenen Anpassungstasks sind noch eine Reihe von Konfigurationseinstellungen zu überprüfen, die Einfluss auf die Leistung und die Ressourcenvoraussetzungen von IBM MQ for z/OS haben. Sie müssen z. B. entscheiden, welche Globalisierungsbibliotheken Sie verwenden möchten.

Wenn Sie einige der Anpassungsschritte automatisieren möchten, lesen Sie den Abschnitt „Verwendung von IBM z/OSMF zur Automatisierung von IBM MQ" auf Seite 1100.

## Konfigurationsoptionen

Weitere Informationen zu diesen Optionen finden Sie unter Planung für z/OS.

Die Beschreibung der einzelnen Tasks in diesem Abschnitt gibt an, ob:

- Die Task ist Teil des Prozesses zum Konfigurieren von IBM MQ. Dies bedeutet, dass Sie die Task einmal ausführen, wenn Sie IBM MQ auf dem z/OS-System anpassen. (In einem parallelen Sysplex müssen Sie die Task für jedes z/OS-System im Sysplex ausführen und sicherstellen, dass die einzelnen z/OS-Systeme identisch konfiguriert sind.)
- Die Task ist Teil des Hinzufügung eines Warteschlangenmanagers. Dies bedeutet, dass Sie die Task für jeden WS-Manager einmal ausführen, wenn Sie diesen Warteschlangenmanager hinzufügen.

Keine der Tasks erfordert ein IPL Ihres z/OS -Systems, wenn Sie Befehle zum Ändern der verschiedenen z/OS -Systemparameter verwenden und „SYS1.PARMLIB-Teildateien aktualisieren” auf Seite 988 wie empfohlen ausführen.

Um den Betrieb zu vereinfachen und die Problembestimmung zu erleichtern, müssen Sie sicherstellen, dass alle z/OS-Systeme identisch konfiguriert sind, damit Warteschlangenmanager im Notfall rasch auf einem System erstellt werden können.

Zur Vereinfachung der Wartung sollten Sie die Definition von Aliasnamen für Ihre IBM MQ-Bibliotheken in Betracht ziehen. Weitere Informationen hierzu finden Sie im Abschnitt Aliasnamen verwenden, um auf eine IBM MQ-Bibliothek zu verweisen.

**Zugehörige Konzepte**

IBM MQ for z/OS - Konzepte

„IBM MQ mit IMS verwenden” auf Seite 1084
Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die es IBM MQ ermöglichen, mit IMS zu interagieren.

„IBM MQ mit CICS verwenden” auf Seite 1093
Um IBM MQ mit CICS verwenden zu können, müssen Sie den IBM MQ CICS-Adapter und optional die Komponenten der IBM MQ CICS bridge konfigurieren.

„OTMA-Exits in IMS verwenden” auf Seite 1096
Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

**Zugehörige Tasks**

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten” auf Seite 1052
In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

IBM MQ for z/OS verwalten

**Zugehörige Verweise**

„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen” auf Seite 1094
Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

## z/OS z/OS-System für IBM MQ konfigurieren

In diesen Abschnitten finden Sie eine schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

### z/OS *Angeben der z/OS-Systemparameter*

Einige der Tasks beziehen sich auf die Aktualisierung der z/OS-Systemparameter. Sie müssen wissen, welche angegeben wurden, als das System-IPL ausgeführt wurde.

- *Diese Task muss einmal für jedes z/OS-System durchgeführt werden, auf dem IBM MQ ausgeführt werden soll.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*

SYS1.PARMLIB(IEASYSpp) enthält eine Liste der Parameter, die auf andere Member von SYS1.PARMLIB zeigen (wobei pp die Systemparameterliste des z/OS-Systems darstellt, die für die Durchführung eines IPL des Systems verwendet wurde).

Zu den Einträgen, die Sie finden müssen:

**Für „APF-Autorisierung der IBM MQ-Ladebibliotheken” auf Seite 974:**
PROG=xx oder APF=aa verweist auf die Liste der berechtigten APF-Bibliotheken (Teildatei PROGxx oder IEFAPFaa)

**Für „Aktualisieren der z/OS-Linkliste und LPA” auf Seite 975:**
LNK=kk zeigt auf die Linkliste (Mitglied LNKLSTkk) LPA=mm verweist auf die LPA-Liste (Member LPALSTmm)

**Für „Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften” auf Seite 979:**
SCH=xx zeigt auf die Tabelle mit den Programmeigenschaften (PPT) (Member SCHEDxx)

**Für „IBM MQ-Subsystem für z/OS definieren” auf Seite 980:**
SSN=ss verweist auf die definierte Subsystemliste (Member IEFSSNss)

## z/OS *APF-Autorisierung der IBM MQ-Ladebibliotheken*

APF-Berechtigen Sie verschiedene Bibliotheken. Einige Lademodule sind möglicherweise bereits berechtigt.

- *Diese Task muss einmal für jedes z/OS-System durchgeführt werden, auf dem IBM MQ ausgeführt werden soll.*
- *Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie sicherstellen, dass die Einstellungen für IBM MQ auf allen z/OS-Systemen in dem Sysplex identisch sind.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*
- *Verwendung der Bibliotheksauslegung (LLA):*
  - *Einige IBM MQ-Anwendungen können zu einer hohen Ein-/Ausgabe führen, um Module aus Bibliotheken zu laden. Diese Ein-/Ausgabe kann mit der LLA-Funktion des Betriebssystems reduziert werden.*
  - *Diese hohe Ein-/Ausgabe kann während der folgenden Schritte auftreten:*
    - *Anwendungen mit einer hohen MQCONN/MQDISC-Rate, z. B. in einer WLM-gespeicherten Prozedur.*
    - *Kanalexits werden geladen. Wenn Sie über Kanäle verfügen, die häufig gestartet und gestoppt werden, und Kanalexits verwenden.*
  - *Das Member CSVLLAxx in SYS1.PARMLIB gibt die LLA-Konfiguration an. Die Angabe eines Bibliotheksnamens in der Anweisung LIBRARIES bedeutet, dass eine Programmkopie immer aus VLF (Virtual Lookaside Facility) übernommen wird und daher in der Regel keine Ein-/Ausgabe erforderlich macht, wenn sie stark genutzt wird.*

    *Inklusion in der Anweisung FREEZE bedeutet, dass es keine E/A gibt, um die relevanten Verkettungsverzeichnisse der Datendefinitionsanweisung abzurufen (dies kann oft mehr E/A sein als die Programmlast selbst).*

    *Betriebssystembefehl verwenden " F LLA, REFRESH " nach Änderungen an einer dieser Bibliotheken.*

Die IBM MQ-Ladebibliotheken thlqual.SCSQAUTH und thlqual.SCSQLINK müssen APF-autorisiert sein. Sie müssen auch die Bibliotheken für Ihre Landessprachenfunktion (thlqual.SCSQANLx und thlqual.SCSQSNLx) und für die verteilte Warteschlangenfunktion (thlqual.SCSQMVR1) mit APF berechtigen.

Alle Lademodule im LPA werden jedoch automatisch mit APF-Berechtigung autorisiert. Dies gilt auch für alle Mitglieder der Linkliste, wenn das SYS1.PARMLIB-Mitglied IEASYSpp die Anweisung enthält:

```
LNKAUTH=LNKLST
```

LNKAUTH=LNKLST ist der Standardwert, wenn LNKAUTH nicht angegeben ist.

Abhängig von den Elementen, die Sie dem LPA oder der Linkliste hinzufügen (siehe „Aktualisieren der z/OS-Linkliste und LPA” auf Seite 975), müssen Sie die Bibliotheken unter Umständen nicht zur APF-Linkliste hinzufügen.

**Anmerkung:** Alle Bibliotheken, die in die STEPLIB von IBM MQ eingefügt werden, müssen für APF autorisiert werden. Wenn Sie eine Bibliothek, die in der STEPLIB nicht APF-autorisiert ist, in eine Bibliothek stellen, verliert die gesamte Bibliothekenverknüpfung ihre APF-Berechtigung.

Die APF-Listen befinden sich im SYS1.PARMLIB-Member PROGxx oder IEAAPFaa. Diese Listen enthalten die Namen der APF-autorisierten z/OS-Bibliotheken. Die Reihenfolge der Einträge in den Listen ist nicht signifikant. Informationen zu APF-Listen finden Sie unter APF-autorisierte Bibliotheksliste .

Weitere Informationen zum Optimieren des Systems finden Sie unter SupportPac MP16.

Wenn Sie PROGxx-Member mit dynamischem Format verwenden, müssen Sie nur den z/OS -Befehl `SETPROG APF,ADD,DSNAME=hlq.SCSQ XXXX,VOLUME= YYYYYY` ausgeben, damit die Änderungen wirksam werden: Dabei ist *XXXX* abhängig vom Bibliotheksnamen und wobei *YYYYY* der Datenträger ist. Andernfalls, wenn Sie ein statisches Format oder IEAAPFaa-Member verwenden, müssen Sie ein IPL auf Ihrem System ausführen.

Beachten Sie, dass Sie den tatsächlichen Namen der Bibliothek in der APF-Liste verwenden müssen. Wenn Sie versuchen, den Datenbankaliasnamen der Bibliothek zu verwenden, schlägt die Autorisierung fehl.

**Zugehörige Konzepte**

„Aktualisieren der z/OS-Linkliste und LPA” auf Seite 975
Aktualisieren Sie die LPA-Bibliotheken mit der neuen Version der Early-Code-Bibliotheken. Der andere Code kann in der Linkliste oder im LPA enthalten sein.

„Vorbereiten der Anpassung von Warteschlangenmanagern unter z/OS” auf Seite 967
Verwenden Sie dieses Thema, wenn Sie Ihre Warteschlangenmanager mit Details zu installierbaren Features, landessprachlichen Features und Informationen zu Tests anpassen und die Sicherheit konfigurieren.

### z/OS *Aktualisieren der z/OS-Linkliste und LPA*

Aktualisieren Sie die LPA-Bibliotheken mit der neuen Version der Early-Code-Bibliotheken. Der andere Code kann in der Linkliste oder im LPA enthalten sein.

- Sie müssen diese Task einmal für jedes z/OS-System ausführen, auf dem IBM MQ ausgeführt werden soll.
- Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, sollten Sie den Early Code in jedem Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange auf die IBM MQ 9.3.0-Stufe aktualisieren, bevor Sie einen der Warteschlangenmanager in IBM MQ 9.3.0 migrieren.

  Installieren Sie den aktuellsten Early Code in jeder LPAR und aktualisieren Sie anschließend die Warteschlangenmanager nacheinander zu einem beliebigen Zeitpunkt vor der Migration. Sie müssen nicht alle Warteschlangenmanager gleichzeitig migrieren.
- Möglicherweise müssen Sie diese Task ausführen, wenn Sie eine Migration von einer früheren Version durchführen. Weitere Informationen finden Sie im Programmverzeichnis. Download-Links für die Programmverzeichnisse finden Sie unter IBM MQ for z/OS Program Directory PDF files.

**Anmerkung:** Der für LPA festgelegte Datensatz ist versionsspezifisch. Wenn Sie einen vorhandenen LPA im System verwenden, wenden Sie sich an Ihren Systemadministrator, um zu entscheiden, welcher LPA verwendet werden soll.

## Vorzeigecode

Einige IBM MQ-Lademodule müssen MVS für IBM MQ hinzugefügt werden, damit sie als Subsystem fungieren können. Diese Module werden als Early-Code bezeichnet und können auch dann ausgeführt werden, wenn ein WS-Manager nicht aktiv ist. Wenn beispielsweise ein Bedienerbefehl auf der Konsole mit einem IBM MQ-Befehlspräfix abgesetzt wird, erhält dieser Vorzeigecode die Kontrolle und überprüft, ob er einen Warteschlangenmanager starten oder die Anforderung an einen aktiven Warteschlangenmanager übergeben muss. Dieser Code wird in den Link Pack Area (LPA) geladen. Es gibt eine Reihe von Vorabmodulen, die für alle Warteschlangenmanager verwendet werden; diese Vorabmodule müssen die höchste Version von IBM MQ aufweisen. Vorabcode aus einer höheren Version von IBM MQ kann für einen Warteschlangenmanager mit einer älteren Version von IBM MQ verwendet werden, umgekehrt ist dies jedoch nicht möglich.

Der frühe Code besteht aus den folgenden Lademodulen:

- CSQ3INI und CSQ3EPX in der Bibliothek thqual.SCSQLINK
- CSQ3ECMX in der Bibliothek thqual.SCSQSNL *x*, wobei *x* Ihr Sprachenbuchstabe ist:
  - thlqual.SCSQSNL E, für amerikanisches Englisch in Groß-/Kleinschreibung
  - thlqual.SCSQSNLU, für amerikanisches Englisch in Großschreibung
  - thlqual.SCSQSNL K, für Japanisch
  - thlqual.SCSQSNL F, für Französisch
  - thlqual.SCSQSNL C, für Chinesisch

IBM MQ enthält eine Benutzermodifikation, die den Inhalt der Bibliothek 'thqual.SCSQSNL *i* in den Wert 'thqual.SCSQLINK' verschiebt und SMP/E informiert. Diese Benutzermodifikation heisst CSQ8UERL und ist im *Programmverzeichnis für IBM MQ for z/OS* beschrieben, entweder für Long Term Support oder für Continuous Delivery. Download-Links für die Programmverzeichnisse finden Sie unter IBM MQ for z/OS Program Directory PDF files.

Nach der Aktualisierung des Vorabcodes in den LPA-Bibliotheken steht dieser Code ab dem nächsten z/OS-Start über IPL (mit der CLPA-Option) für alle WS-Manager-Subsysteme zur Verfügung, die beim IPL aus den Definitionen in den IEFSSNss-Membern in SYS1.PARMLIB hinzugefügt wurden.

Sie können sie sofort ohne IPL verfügbar machen, wenn ein neues Warteschlangenmanager-Subsystem später hinzugefügt wird (wie in „IBM MQ-Subsystem für z/OS definieren” auf Seite 980 beschrieben), indem Sie es wie folgt zum LPA hinzufügen:

- Wenn Sie die Benutzermodifikation CSQ8UERL nicht verwendet haben, geben Sie die folgenden z/OS-Befehle aus:

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thqual.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqual.SCSQSNL x
```

- Wenn CSQ8UERL verwendet wurde, können Sie den Vorabcode mit dem folgenden z/OS-Befehl in den Link-Pack-Bereich laden:

```
SETPROG LPA,ADD,MASK=*,DSNAME=thqual.SCSQLINK
```

- Wenn Sie Advanced Message Security verwenden, müssen Sie außerdem folgenden z/OS-Befehl ausgeben, um ein zusätzliches Modul in den Link-Pack-Bereich einzuschließen:

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thqual.SCSQLINK
```

Wenn Sie Wartungspakete angewendet haben oder einen Warteschlangenmanager mit einer neueren Version oder einem höheren Release von IBM MQerneut starten wollen, können Sie den Vorabcode mit den folgenden Schritten für vorhandene Warteschlangenmanager verfügbar machen. Warteschlangenmanager, für die Sie diese Schritte nicht ausführen, verwenden weiterhin die Version des Vorabcodes, die sie bereits verwenden. Es ist nicht erforderlich, diese Schritte für alle Warteschlangenmanager in einer LPAR auszuführen, es sei denn, Sie versuchen ausdrücklich, eine Wartung auf alle Warteschlangenmanager anzuwenden oder alle auf eine neuere Version oder ein aktuelleres Release von IBM MQzu aktualisieren.

1. Fügen Sie den Code (wie zu Beginn dieses Abschnitts beschrieben) mit den SETPROG-Befehlen von z/OS dem Link-Pack-Bereich hinzu.
2. Stoppen Sie den Warteschlangenmanager mit dem IBM MQ-Befehl STOP QMGR.
3. Stellen Sie sicher, dass das Sicherheitsprofil 'qmgr.REFRESH.QMGR' konfiguriert ist. Siehe MQSC-Befehle, -Profile und deren Zugriffsebenen .
4. Aktualisieren Sie den Vorabcode für den Warteschlangenmanager mit dem IBM MQ-Befehl REFRESH QMGR TYPE(EARLY).
5. Starten Sie den Warteschlangenmanager mit dem IBM MQ-Befehl START QMGR erneut.

Die IBM MQ-Befehle STOP QMGR, REFRESH QMGR und START QMGR werden im Abschnitt MQSC-Befehle beschrieben.

## Anderer Code

Alle von IBM MQ bereitgestellten Lademodule in den folgenden Bibliotheken sind wiedereintrittsfähig und können in den Link-Pack-Bereich eingefügt werden:

- SCSQAUTH
- SCSQANL *x*, wobei *x* Ihr Sprachenbuchstabe ist
- SCSQMVR1

**Wichtig:** Wenn Sie jedoch die Bibliotheken in den LPA stellen, müssen Sie, wenn Sie die Wartung anwenden, alle geänderten Module manuell in den LPA kopieren. Daher empfiehlt es sich, die IBM MQ -Ladebibliotheken in die Linkliste zu stellen, die nach der Wartung durch Absetzen des z/OS -Befehls MODIFY LLA REFRESH aktualisiert werden kann.

Weitere Informationen finden Sie unter Inhalt von LNKLST-Datasets ändern und unter Dynamische LNKLST-Funktion sicher und ordnungsgemäß verwenden.

Dies wird insbesondere für SCSQAUTH empfohlen, so dass Sie es nicht in mehrere STEPLIBs aufnehmen müssen. Nur eine Sprachbibliothek, SCSQANL *x* , sollte in die LPA-oder Linkliste gestellt werden. Die Linklistenbibliotheken sind in einem LNKLSTkk-Member von SYS1.PARMLIB angegeben.

Die verteilte Warteschlangenfunktion und CICS bridge (aber nicht der Warteschlangenmanager selbst) benötigen Zugriff auf die LE-Laufzeitbibliothek (Language Environment) SCEERUN. Wenn Sie eine dieser Funktionen verwenden, müssen Sie SCEERUN in die Linkliste aufnehmen.

V 9.3.2 Einige Module werden beim Start des Warteschlangenmanagers in ECSA geladen. In eingeschränkten ECSA-Umgebungen ist es möglich, diese Module stattdessen in den Link-Pack-Bereich zu stellen. Weitere Informationen finden Sie unter „Globale IBM MQ -Module in den Link-Pack-Bereich stellen” auf Seite 977.

**Wichtig:** LTS Wenn Sie diese Funktion unter IBM MQ 9.3 verwenden möchten, müssen Sie APAR PH52358anwenden.

**Zugehörige Konzepte**

„Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften” auf Seite 979
Für den IBM MQ-Warteschlangenmanager sind einige zusätzliche Einträge in der Programmeigenschaftentabelle (Program Properties Table, PPT) erforderlich.

z/OS V 9.3.2 *Globale IBM MQ -Module in den Link-Pack-Bereich stellen*

Wenn ein IBM MQ for z/OS -Warteschlangenmanager gestartet wird, lädt er einige seiner Lademodule (globalen Module) in den erweiterten allgemeinen Servicebereich (ECSA). Beim Beenden des Warteschlangenmanagers wird der ECSA freigegeben.

Es gibt 19 globale Module, die unter IBM MQ 9.3ungefähr 1.2 MB ECSA für jeden aktiven Warteschlangenmanager belegen.

**Anmerkung:** Obwohl CSQ7GPLM ein globales Modul ist, sollte es nicht zum Link-Pack-Bereich hinzugefügt werden.

In Umgebungen, in denen mehrere Warteschlangenmanager für jede LPAR ausgeführt werden und die eine Reduzierung der ECSA-Nutzung aufgrund von ECSA oder hohen privaten Einschränkungen erfordern, ist es möglich, die globalen Module im LPA zu platzieren. Die Platzierung der globalen Module von IBM MQ im Link-Pack-Bereich ist ein manueller Prozess, bei dem Vorsicht geboten ist. Daher sollten Sie diese Prozedur nur ausführen, wenn ECSA oder hohe private Einschränkungen beachtet werden müssen.

**Wichtig:** LTS Wenn Sie diese Funktion unter IBM MQ 9.3 verwenden möchten, müssen Sie APAR PH52358anwenden.

Wenn der Warteschlangenmanager in seiner STEPLIB kein globales Modul findet und feststellt, dass sich das Modul im LPA befindet, verwendet er die LPA-Kopie direkt, anstatt eine Kopie des Moduls in ECSA zu laden. Wenn der Warteschlangenmanager-Code normalerweise aus der Linkliste geladen wird, haben alle globalen Module im Link-Pack-Bereich Vorrang vor allen globalen Modulen in der Linkliste.

Die z/OS -Funktion zur Verfolgung des gemeinsamen Speichers (siehe Überwachungsfunktion des gemeinsamen Speichers verwenden) verfolgt den Speicher unter dem MSTR-Adressraum jedes Warteschlangenmanagers und kann verwendet werden, um festzustellen, wie viel Speicherplatz von den globalen Modulen belegt wird.

Standardmäßig befinden sich die globalen Module in der Ladebibliothek SCSQAUTH. Wenn der MSTR-Adressraum eines Warteschlangenmanagers SCSQAUTH über die STEPLIB-Verkettung lokalisiert, werden die globalen Module von dort bevorzugt im LPA verwendet und in ECSA geladen.

Die globalen Module sind:

CSQ0GPLM, CSQ3AMGP, CSQ3SSGP, CSQ9PREP,
CSQ9SCNB, CSQGGPLM, CSQMCGLM, CSQMGPLM, CSQRGLM1,
CSQSLD1, CSQVGEPL, CSQVSRX, CSQWDLD2, CSQWDLD3,
CSQWVZSA, CSQWZDG0, CSQWVZPS, CSQWVGTM, CSQZTDDM

**Wichtig:**

- Der Name der globalen Module für IBM MQ bleibt in verschiedenen IBM MQ -Versionen konstant. Wenn Sie globale Module in den Link-Pack-Bereich laden, sollten sie daher aus einer einzigen IBM MQ -Version stammen und nur von Warteschlangenmanagern mit derselben IBM MQ -Version verwendet werden.
- Wenn mehrere Versionen von IBM MQ in derselben LPAR ausgeführt werden, kann jeweils nur eines dieser Module seine globalen Module im LPA haben.
- Wenn eine Wartung auf eine IBM MQ -Installation angewendet wird, bei der globale Module in den Link-Pack-Bereich geladen sind und diese Wartung eines der globalen Module aktualisiert, sollten Sie die im folgenden Text beschriebene Prozedur erneut ausführen.

## Verfahren

Führen Sie die folgenden Schritte aus, um die globalen Module aus einer Version von IBM MQ in den Link-Pack-Bereich zu stellen:

1. Erstellen Sie eine Kopie der `thlqual.SCSQAUTH` -Ladebibliothek und ihren Inhalt, z. B. `thlqual.LOCAL.SCSQAUTH`. Stellen Sie sicher, dass diese Ladebibliothek mit dem externen Sicherheitsmanager (ESM) vor unbefugtem Zugriff geschützt ist.
2. Autorisieren Sie die `thlqual.LOCAL.SCSQAUTH` -Ladebibliothek mit APF (siehe „APF-Autorisierung der IBM MQ-Ladebibliotheken” auf Seite 974).
3. Erstellen Sie eine neue`hlqual.GLOBAL.SCSQAUTH` -Ladebibliothek mit denselben Attributen wie `thlqual.LOCAL.SCSQAUTH`.

   **Anmerkung:** Diese Ladebibliothek muss nicht APF-berechtigt sein. Stellen Sie sicher, dass diese Ladebibliothek vor unbefugtem Zugriff mit ESM geschützt ist.
4. Kopieren Sie die 19 globalen Module aus `thlqual.LOCAL.SCSQAUTH` in `thlqual.GLOBAL.SCSQAUTH`.
5. Löschen Sie die 19 globalen Module aus `thlqual.LOCAL.SCSQAUTH`.
6. Platzieren Sie die 19 globalen Module aus `thlqual.GLOBAL.SCSQAUTH` wie folgt in den Link-Pack-Bereich (LPA):

   a. a. Hinzufügen von `thlqual.GLOBAL.SCSQAUTH` zu einem `LPALSTxx` -Member von SYS1.PARMLIB Anschließend muss für das System ein IPL mit der Option CLPA durchgeführt werden, um sicherzustellen, dass der Bibliotheksinhalt in den PLPA geladen wird.

   b. b. Fügen Sie die Module mit dem folgenden Befehl dynamisch zum LPA hinzu:

```
SETPROG
LPA,ADD,MODNAME=(CSQ0GPLM,CSQ3AMGP,CSQ3SSGP,CSQ9PREP,CSQ9SCNB,CSQGGPLM,
CSQMCGLM,CSQMGPLM,CSQRGLM1,CSQSLD1,CSQVGEPL,CSQVSRX,CSQWDLD2,CSQWDLD3,
CSQWVZSA,CSQWZDG0,CSQWVZPS,CSQWVGTM,CSQZTDDM),DSNAME= thlqual.GLOBAL.SCSQAUTH
```

**Anmerkung:** LPALSTxx ist die bevorzugte langfristige Möglichkeit, Module im Link-Pack-Bereich zu platzieren.

7. Überprüfen Sie, ob sich die Module im Link-Pack-Bereich (LPA) befinden, indem Sie den folgenden Befehl ausgeben:

```
D PROG,LPA,MODNAME=CSQMCGLM
```

Die Ausgabe des Befehls sollte die Eingangs-und Ladepunkte des Moduls angeben, wenn es erfolgreich in den Link-Pack-Bereich geladen wurde.

Für jeden Warteschlangenmanager, der die globalen Module aus dem Link-Pack-Bereich verwenden muss, wenn Sie normalerweise Folgendes angeben:

1. `thlqual.SCSQAUTH` in der Linkliste stoppen und starten Sie Ihren Warteschlangenmanager. Die globalen Module werden aus dem LPA und die lokalen Module aus der Linkliste geladen.
2. `thlqual.SCSQAUTH` in der MSTR-JCL STEPLIB ändern Sie die JCL so, dass die STEPLIB `thlqual.LOCAL.SCSQAUTH` anstelle von `thlqual.SCSQAUTH`verwendet. Stoppen und starten Sie den Warteschlangenmanager. Die globalen Module werden aus dem Link-Pack-Bereich und die lokalen Module aus der STEPLIB geladen.

Die CHIN-und AMSM-JCL kann weiterhin `thlqual.SCSQAUTH` wie alle anderen IBM MQ -Anwendungen verwenden.

Führen Sie die folgenden Schritte aus, um den Warteschlangenmanager auf das Laden der globalen Module in ECSA zurückzusetzen:

1. Warteschlangenmanager stoppen
2. Entfernen Sie die globalen Module aus dem Link-Pack-Bereich beim nächsten IPL, indem Sie die LPALSTxx -Definitionen entfernen oder den folgenden Befehl verwenden:

```
SETPROG LPA,DELETE,MODNAME=(xxx) FORCE=YES
```

3. Wenn `thlqual.LOCAL.SCSQAUTH` in der STEPLIB des Warteschlangenmanagers enthalten ist, ersetzen Sie es durch `thlqual.SCSQAUTH`.
4. Starten Sie die Warteschlangenmanager neu.

**Zugehörige Konzepte**

„Aktualisieren der z/OS-Linkliste und LPA” auf Seite 975
Aktualisieren Sie die LPA-Bibliotheken mit der neuen Version der Early-Code-Bibliotheken. Der andere Code kann in der Linkliste oder im LPA enthalten sein.

## z/OS *Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften*

Für den IBM MQ-Warteschlangenmanager sind einige zusätzliche Einträge in der Programmeigenschaftentabelle (Program Properties Table, PPT) erforderlich.

- *Diese Task muss für jedes z/OS-System ausgeführt werden, auf dem IBM MQ eingesetzt werden soll.*
- *Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie sicherstellen, dass die Einstellungen für IBM MQ auf allen z/OS-Systemen in dem Sysplex identisch sind.*
- *Bei der Migration von einer früheren Version muss diese Task nicht ausgeführt werden.*
- *Sie müssen den CSQ0DSRV-Teil dieser Task ausführen, wenn Sie Advanced Message Security benötigen.*

In thlqual.SCSQPROC (CSQ4SCHD) wird ein Beispiel bereitgestellt, das alle erforderlichen PPT-Einträge enthält. Stellen Sie sicher, dass die erforderlichen Einträge zu der PPT hinzugefügt werden, die Sie in SYS1.PARMLIB (SCHEDxx) finden.

In z/OSist CSQYASCP bereits für das Betriebssystem mit den Attributen definiert und muss nicht mehr in ein SCHEDxx-Member von PARMLIB eingeschlossen werden.

Der IBM MQ-Warteschlangenmanager steuert selbst die Auslagerungsfunktion. Falls Ihr IBM MQ-Netz jedoch stark ausgelastet und die Antwortzeit kritisch ist, kann es von Vorteil sein, die Auslagerungsfähigkeit

des IBM MQ-Kanalinitiators zu unterbinden, indem der PPT-Eintrag CSQXJST hinzugefügt wird. Dies kann jedoch die Leistung des übrigen z/OS-Systems beeinträchtigen.

Wenn Sie Advanced Message Security benötigen, fügen Sie den Eintrag CSQ0DSRV PPT hinzu.

Setzen Sie den z/OS -Befehl **SET SCH=xx**ab, wobei *xx* das Suffix des PARMLIB-Members SCHEDxx ist, damit die Änderungen wirksam werden.

**Zugehörige Konzepte**

„IBM MQ-Subsystem für z/OS definieren" auf Seite 980
Aktualisieren Sie die Tabelle mit dem Subsystemnamen und entscheiden Sie sich für eine Konvention für Befehlspräfixzeichenfolgen.

## z/OS Warteschlangenmanager und Kanalinitiator konfigurieren

Verwenden Sie diese Themen als Schritt-Anleitung für die Konfiguration des Warteschlangenmanagers und des Kanalinitiators.

### z/OS *IBM MQ-Subsystem für z/OS definieren*

Aktualisieren Sie die Tabelle mit dem Subsystemnamen und entscheiden Sie sich für eine Konvention für Befehlspräfixzeichenfolgen.

Wiederholen Sie diese Task für jeden IBM MQ-Warteschlangenmanager. Sie müssen diese Task bei der Migration von einer früheren Version nicht ausführen.

**Zugehörige Konzepte**

„Prozeduren für den IBM MQ-Warteschlangenmanager erstellen" auf Seite 984
Für jedes IBM MQ-Subsystem ist eine katalogisierte Prozedur zum Starten des Warteschlangenmanagers erforderlich. Sie können eigene Prozeduren erstellen oder die von IBM bereitgestellte Prozedurbibliothek verwenden.

#### z/OS *Aktualisieren der Subsystemnamentabelle*

Wenn Sie das IBM MQ-Subsystem definieren, müssen Sie einen Eintrag zur Subsystemnamentabelle hinzufügen.

Die Subsystemnamentabelle von z/OS, die anfänglich aus dem Member IEFSSNss von SYS1.PARMLIB entnommen wird, enthält die Definitionen von formal definierten z/OS-Subsystemen. Um die einzelnen IBM MQ-Subsysteme zu definieren, müssen Sie einen Eintrag zu dieser Tabelle hinzufügen, indem Sie entweder das IEFSSNss-Member von SYS1.PARMLIB ändern oder, vorzugsweise, den z/OS-Befehl SETSSI verwenden.

Die IBM MQ-Subsysteminitialisierung unterstützt die parallele Verarbeitung. Deshalb können IBM MQ-Subsystemdefinitionsanweisungen sowohl oberhalb als auch unterhalb des Schlüsselworts BEGINPARALLEL in der Tabelle IEFSSNss hinzugefügt werden, die unter z/OS V1.12 und höher verfügbar ist.

Wenn Sie den Befehl SETSSI verwenden, wird die Änderung sofort wirksam, und es ist nicht erforderlich, ein IPL für das System auszuführen. Stellen Sie sicher, dass Sie auch SYS1.PARMLIB wie im Abschnitt „SYS1.PARMLIB-Teildateien aktualisieren" auf Seite 988 beschrieben aktualisieren, damit die Änderungen nach nachfolgenden IPLs wirksam bleiben.

Der Befehl SETSSI zum dynamischen Definieren eines IBM MQ-Subsystems ist:

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

Die entsprechenden Informationen in IEFSSNss können auf eine der beiden folgenden Arten angegeben werden:

- Das Schlüsselwortparameterformat der IBM MQ-Subsystemdefinition in IEFSSNss. Dies ist die empfohlene Methode.

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- Das positionsgebundene Parameterformat der IBM MQ-Subsystemdefinition.

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

Mischen Sie die beiden Formulare nicht in einem IEFSSNss-Member. Wenn verschiedene Formulare erforderlich sind, verwenden Sie für jeden Typ ein separates IEFSSNss-Member, und fügen Sie den Operanden SSN des neuen Members zum Member IEASYSpp SYS1.PARMLIB hinzu. Um mehrere SSN anzugeben, verwenden Sie SSN = (aa, bb, ...) in IEASYSpp.

In den Beispielen

**ssid**
: Die Subsystemkennung. Er kann bis zu vier Zeichen lang sein. Alle Zeichen müssen alphanumerisch sein (Großschreibung A bis Z, 0 bis 9), sie muss mit einem alphabetischen Zeichen beginnen. Der Warteschlangenmanager hat denselben Namen wie das Subsystem. Daher können nur Zeichen verwendet werden, die sowohl für z/OS-Subsystemnamen als auch für IBM MQ-Objektnamen zulässig sind.

**cpf**
: Die Befehlspräfixzeichenfolge (Informationen zu Befehlspräfixzeichenfolgen finden Sie im Abschnitt „Befehlspräfixzeichenfolgen definieren (CPFs)” auf Seite 982).

**scope**
: Der Systembereich, der verwendet wird, wenn Sie in einem z/OS-Sysplex ausgeführt werden (Informationen zum Systembereich finden Sie im Abschnitt „CPFs in einer Sysplex-Umgebung” auf Seite 983).

Im Abschnitt Abbildung 100 auf Seite 981 finden Sie verschiedene Beispiele für IEFSSNss-Anweisungen.

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

*Abbildung 100. Beispiele für IEFSSNss-Anweisungen zum Definieren von Subsystemen*

**Anmerkung:** Wenn Sie Objekte in einem Subsystem erstellt haben, können Sie den Subsystemnamen nicht ändern oder die Seitengruppen von einem Subsystem in einem anderen Subsystem verwenden. Dazu müssen Sie alle Objekte und Nachrichten aus einem Subsystem entladen und in ein anderes Subsystem erneut laden.

Im Abschnitt Tabelle 54 auf Seite 981 finden Sie eine Reihe von Beispielen, in denen die Zuordnungen von Subsystemnamen und Befehlspräfixzeichenfolgen, wie in den Anweisungen im Abschnitt Abbildung 100 auf Seite 981 definiert, dargestellt werden.

*Tabelle 54. Subsystemname zu CPF-Zuordnungen*

| **IBM MQ-Subsystemname** | **CPF** |
|---|---|
| CSQ1 | +mqs1cpf |
| CSQ2 | +mqs2cpf |
| CSQ3 | ++ |

**Anmerkung:** Die Funktionen ACTIVATE und DEACTIVATE des z/OS-Befehls SETSSI werden von IBM MQ nicht unterstützt.

Setzen Sie den folgenden Befehl in SDSFab, um den Status der Änderungen zu überprüfen: `/D SSI,L`. Es werden die neuen Subsysteme angezeigt, die mit dem Status AKTIV erstellt wurden.

z/OS *Befehlspräfixzeichenfolgen definieren (CPFs)*

Jede Subsysteminstanz von IBM MQ kann über eine Befehlspräfixzeichenfolge verfügen, um dieses Subsystem zu identifizieren.

Sie können eine systemweite Konvention für Ihre CPFs für alle Subsysteme verwenden, um Konflikte zu vermeiden. Hier finden Sie die folgenden Richtlinien:

- Definieren Sie ein CPF als Zeichenfolge mit bis zu acht Zeichen.
- Verwenden Sie kein CPF, die bereits von einem anderen Subsystem verwendet wird, und vermeiden Sie die Verwendung des auf Ihrem System definierten JES-Rückspeicherzeichens als erstes Zeichen Ihrer Zeichenfolge.
- Definieren Sie Ihr CPF mit Zeichen aus der Gruppe der gültigen Zeichen, die in Tabelle 56 auf Seite 983 aufgelistet sind.
- Verwenden Sie kein CPF, die eine Abkürzung für einen bereits definierten Prozess darstellt oder die mit der Befehlssyntax verwechselt werden kann. Beispiel: Ein CPF wie z. B. 'D' steht in Konflikt mit z/OS-Befehlen wie DISPLAY. Um dies zu vermeiden, verwenden Sie eines der Sonderzeichen (in Tabelle 56 auf Seite 983 dargestellt) als erstes oder einziges Zeichen in Ihrer CPF-Zeichenfolge.
- Definieren Sie kein CPF, das entweder eine Untergruppe oder ein Superset eines vorhandenen CPF ist. Ein Beispiel hierzu finden Sie im Thema Tabelle 55 auf Seite 982.

*Tabelle 55. Beispiel für die CPF-Subset-und Superset-Regeln*

| Subsystemname | CPF definiert | Weitergeleitete Befehle |
|---|---|---|
| MQA | !A | MQA |
| MQB | !B | MQB |
| MQC1 | !C1 | MQC1 |
| MQC2 | !C2 | MQC2 |
| MQB1 | !B1 | MQB |

Befehle für Subsystem MQB1 (mit CPF!B1) werden an Subsystem MQB weitergeleitet, da der CPF für dieses Subsystem ist!B, ein Teil von!B1. Wenn Sie z. B. den folgenden Befehl eingegeben haben:

```
!B1  START  QMGR
```

Subsystem MQB empfängt den folgenden Befehl:

```
1 START QMGR
```

(die in diesem Fall nicht behandelt werden können).

Sie können sehen, welche Präfixe vorhanden sind, indem Sie den z/OS-Befehl DISPLAY OPDATA ausgeben.

Wenn Sie in einem Sysplex arbeiten, diagnostiziert z/OS alle Konflikte dieses Typs zum Zeitpunkt der CPF-Registrierung (siehe „CPFs in einer Sysplex-Umgebung” auf Seite 983 zu Informationen zur CPF-Registrierung).

In Tabelle 56 auf Seite 983 sind die Zeichen aufgeführt, die Sie bei der Definition von CPF-Zeichenfolgen verwenden können:

| *Tabelle 56. Gültiger Zeichensatz für CPF-Zeichenfolgen* | |
|---|---|
| **Zeichensatz** | **Inhalt** |
| Alphabetisch | Großschreibung A bis Z, Kleinbuchstaben a bis z |
| Numerisch | 0 bis 9 |
| National (siehe Anmerkung) | @ $# (Zeichen, die als Hexadezimalwerte dargestellt werden können) |
| Spezial | . □ ( ) * & + - = ¢ < \| ! ; % _ ? : > |

**Anmerkung:**

Das System erkennt die folgenden hexadezimalen Darstellungen der nationalen Zeichen: @ als X'7C ', $als X'5B', und # als X'7B '. In anderen Ländern als den U.S., Die auf Terminaltastaturen dargestellten U.S. nationalen Sonderzeichen können eine andere hexadezimale Darstellung generieren und einen Fehler verursachen. In einigen Ländern kann das $-Zeichen beispielsweise einen X'4A ' generieren.

Das Semikolon (;) ist als CPF gültig, aber auf den meisten Systemen ist dieses Zeichen der Befehlsbegrenzer.

z/OS *CPFs in einer Sysplex-Umgebung*

In diesem Thema wird erläutert, wie Sie CPFs im Geltungsbereich eines Sysplex verwenden können.

Bei Verwendung in einer Sysplex-Umgebung registriert IBM MQ Ihre CPFs, um Ihnen die Eingabe eines Befehls von einer beliebigen Konsole im Sysplex und das Leiten dieses Befehls an das entsprechende System zwecks Ausführung zu ermöglichen. Die Befehlsantworten werden an die ursprüngliche Konsole zurückgegeben.

## Geltungsbereich für Sysplex-Operation definieren

Der Geltungsbereich wird verwendet, um den Typ der CPF-Registrierung zu ermitteln, die vom IBM MQ-Subsystem ausgeführt wird, wenn Sie IBM MQ in einer Sysplex-Umgebung ausführen.

Gültige Werte für den Geltungsbereich sind:

**M**

Systembereich.

Die CPF wird von IBM MQ beim einleitenden Programmladen des Systems bei z/OS registriert und bleibt für die gesamte Zeit registriert, in der das z/OS -System aktiv ist.

IBM MQ-Befehle müssen an einer Konsole eingegeben werden, die mit dem z/OS-Image verbunden ist, auf dem das Zielsubsystem ausgeführt wird, oder Sie müssen ROUTE-Befehle verwenden, um den Befehl an dieses Image zu leiten.

Verwenden Sie diese Option, wenn Sie nicht in einem Sysplex ausgeführt werden.

**S**

Der Bereich Sysplex wurde gestartet.

Die CPF wird bei z/OS registriert, wenn das Subsystem IBM MQ gestartet wird, und bleibt aktiv, bis das Subsystem IBM MQ beendet wird.

Sie müssen ROUTE-Befehle verwenden, um den ursprünglichen Befehl START QMGR auf das Zielsystem zu übertragen, aber alle weiteren IBM MQ-Befehle können an jeder Konsole eingegeben werden, die mit dem Sysplex verbunden ist, und werden automatisch an das Zielsystem weitergeleitet.

Nach der Beendigung von IBM MQ müssen Sie die ROUTE-Befehle verwenden, um nachfolgende START-Befehle an das IBM MQ-Zielsubsystem zu leiten.

**X**
Sysplex-IPL-Bereich.

Die CPF wird von IBM MQ beim einleitenden Programmladen des Systems bei z/OS registriert und bleibt für die gesamte Zeit registriert, in der das z/OS -System aktiv ist.

IBM MQ -Befehle können an jeder Konsole eingegeben werden, die mit dem Sysplex verbunden ist, und werden an das Image weitergeleitet, in dem das Zielsystem automatisch ausgeführt wird.

Ein IBM MQ-Subsystem mit einem CPF mit dem Geltungsbereich S kann auf einem oder mehreren z/OS-Images in einem Sysplex definiert werden, sodass diese Images eine einzelne Subsystemnamentabelle gemeinsam nutzen können. Sie müssen jedoch sicherstellen, dass der Anfangsbefehl START auf dem z/OS-Image, auf dem das IBM MQ-Subsystem ausgeführt werden soll, ausgegeben wird (oder an das Image weitergeleitet wird). Wenn Sie diese Option verwenden, können Sie das IBM MQ-Subsystem stoppen und auf einem anderen z/OS-Image innerhalb des Sysplex erneut starten, ohne die Subsystemnamentabelle ändern zu müssen, oder ein IPL eines z/OS-Systems durchführen zu müssen.

Ein IBM MQ-Subsystem mit einem CPF mit dem Geltungsbereich "X" kann nur auf einem z/OS-Image in einem Sysplex definiert werden. Wenn Sie diese Option verwenden, müssen Sie für jedes z/OS-Image eine eindeutige Subsystemnamentabelle definieren, die IBM MQ-Subsysteme mit CPFs des Geltungsbereichs X erfordert.

Wenn Sie den z/OS Automatic Restart Manager (ARM) verwenden möchten, um Warteschlangenmanager in verschiedenen z/OS-Images automatisch erneut zu starten, muss jeder WS-Manager in jedem z/OS-Image definiert werden, in dem dieser Warteschlangenmanager erneut gestartet werden kann. Jeder WS-Manager muss mit einem systemspezifischen, eindeutigen, aus vier Zeichen umfassenden Subsystemnamen mit einem CPF-Geltungsbereich von S definiert werden.

## z/OS *Prozeduren für den IBM MQ-Warteschlangenmanager erstellen*

Für jedes IBM MQ-Subsystem ist eine katalogisierte Prozedur zum Starten des Warteschlangenmanagers erforderlich. Sie können eigene Prozeduren erstellen oder die von IBM bereitgestellte Prozedurbibliothek verwenden.

- Wiederholen Sie diese Task für jeden IBM MQ-Warteschlangenmanager.
- Möglicherweise müssen Sie die katalogisierte Prozedur ändern, wenn Sie eine Migration von einer früheren Version durchführen.

Erstellen Sie in einer Prozedurbibliothek für jedes in der Tabelle mit den Subsystemnamen definierte IBM MQ-Subsystem eine katalogisierte Prozedur zum Starten des Warteschlangenmanagers. Die IBM-Prozedurbibliothek hat den Namen SYS1.PROCLIB, aber Ihre Installation kann eine eigene Namenskonvention verwenden.

Der Name der gestarteten Taskprozedur des WS-Managers wird durch Verkettung des Subsystemnamens mit den Zeichen MSTR gebildet. Das Subsystem CSQ1 hat beispielsweise den Prozedurnamen CSQ1MSTR. Sie benötigen eine Prozedur für jedes Subsystem, das Sie definieren.

Sie müssen die Bibliothek, die Nachrichten enthält, in die ausgewählte Sprache aufnehmen:

- thlqual.SCSQSNL E, für amerikanisches Englisch in Groß-/Kleinschreibung
- thlqual.SCSQSNLU, für amerikanisches Englisch in Großschreibung
- thlqual.SCSQSNL K, für Japanisch
- thlqual.SCSQSNL F, für Französisch
- thlqual.SCSQSNL C, für Chinesisch

Bei zahlreichen Beispielen und Anweisungen in dieser Produktdokumentation wird davon ausgegangen, dass Sie über ein Subsystem mit dem Namen CSQ1 verfügen. Sie können diese Beispiele einfacher verwenden, wenn zunächst für Installationsprüfungs- und Testzwecke ein Subsystem mit dem Namen CSQ1 erstellt wird.

In thlqual.SCSQPROC werden zwei Musterprozeduren für gestartete Tasks bereitgestellt. Das Member CSQ4MSTR verwendet für jede Nachrichtenklasse eine Seitengruppe. Das Member CSQ4MSRR verwendet für die Hauptklassen der Nachricht mehrere Seitengruppen. Kopieren Sie diese Prozeduren in das Member 'xxxxMSTR' (dabei ist 'xxxx' der Name des IBM MQ-Subsystems) von SYS1.PROCLIB oder (falls Sie nicht SYS1.PROCLIB verwenden) Ihrer Prozedurbibliothek. Kopieren Sie die Musterprozedur für jedes von Ihnen definierte IBM MQ-Subsystem in eine Teildatei in Ihrer Prozedurbibliothek.

Wenn Sie die Member kopiert haben, können Sie sie mit den Anweisungen in der Teildatei an die Anforderungen der einzelnen Subsysteme anpassen. Informationen zur Angabe von Speicherbegrenzungen, die vom Warteschlangenmanager verwendet werden, finden Sie im Abschnitt Speicherkonfiguration. Sie können auch symbolische Parameter in der JCL verwenden, um die Prozedur zu ändern, wenn sie gestartet wird. Sind mehrere IBM MQ-Subsysteme vorhanden, ist es unter Umständen sinnvoller, für die allgemeinen Abschnitte der Prozedur JCL-Include-Gruppen zu verwenden, um künftige Wartungen zu erleichtern.

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, muss die STEPLIB-Verkettung die Db2-Laufzeitzielbibliothek SDSNLOAD enthalten, und die APF-Berechtigung muss vorhanden sein. Diese Bibliothek ist nur in der STEPLIB-Verkettung erforderlich, wenn sie über die Linkliste oder den LPA nicht zugänglich ist.

**Anmerkungen:**

1. Sie können die Namen Ihrer Bootstrap-Dateigruppe (BSDS), Protokolle und Seitengruppen zur Verwendung in JCL notieren und diese Sätze dann zu einem späteren Schritt in dem Prozess definieren.
2. Die Beispielprozeduren CSQ4MSTR und CSQ4MSRR für gestartete Tasks wurden aktualisiert und enthalten jetzt die CSQMINI DD-Karte (die allerdings auf Kommentar gesetzt ist), mit der ein QMINI-Dataset definiert werden kann, das die Transportsicherheit, also SSL- oder TLS-Eigenschaften, enthält.

   Mit „Dataset QMINI” auf Seite 992 können Sie die TLS 1.3-Unterstützung aktivieren oder inaktivieren und/oder eine benutzerdefinierte Liste der CipherSpecs definieren, die von Kanälen verwendet werden können.

**Zugehörige Konzepte**

„Prozeduren für den Kanalinitiator erstellen” auf Seite 985
Passen Sie für jedes IBM MQ-Subsystem eine Kopie von CSQ4CHIN an. Abhängig von den anderen Produkten, die Sie verwenden, müssen Sie möglicherweise den Zugriff auf andere Dateien zulassen.

## z/OS *Prozeduren für den Kanalinitiator erstellen*

Passen Sie für jedes IBM MQ-Subsystem eine Kopie von CSQ4CHIN an. Abhängig von den anderen Produkten, die Sie verwenden, müssen Sie möglicherweise den Zugriff auf andere Dateien zulassen.

- Wiederholen Sie diese Task für jeden IBM MQ-Warteschlangenmanager.
- Möglicherweise müssen Sie die katalogisierte Prozedur ändern, wenn Sie eine Migration von einer früheren Version durchführen.

Für jedes IBM MQ-Subsystem, das die verteilte Steuerung von Warteschlangen verwenden soll, muss eine Prozedur für gestartete Tasks für den Kanalinitiator erstellt werden.

Gehen Sie dazu wie folgt vor:

1. Kopieren Sie die gestartete Taskprozedurprozedur thlqual.SCSQPROC (CSQ4CHIN) in die Prozedurbibliothek. Benennen Sie die Prozedur *xxxx* CHIN, wobei *xxxx* der Name Ihres IBM MQ -Subsystems ist ( CSQ1CHIN wäre beispielsweise die Prozedur der gestarteten Kanalinitiatortask für Warteschlangenmanager CSQ1).
2. Erstellen Sie eine Kopie für jedes IBM MQ-Subsystem, das verwendet werden soll.
3. Passen Sie die Prozeduren gemäß den Anweisungen in der Beispielprozedur CSQ4CHIN an Ihre Anforderungen an. Sie können auch symbolische Parameter in der JCL verwenden, um die Prozedur zu ändern, wenn sie gestartet wird. Dies wird mit den Startoptionen im Abschnitt IBM MQ for z/OS verwalten beschrieben.

   Katalogisieren Sie die verteilte Warteschlangenbibliothek thlqual.SCSQMVR1.

Der Zugriff auf die LE-Laufzeitbibliothek SCEERUN ist erforderlich. Ist dies nicht in der Linkliste (SYS1.PARMLIB (LNKLSTkk)), verketten Sie sie in der Datendefinitionsanweisung STEPLIB.

V 9.3.1 Sie können den Parameter MEMLIMIT mithilfe der Informationen in Speicherkonfiguration anpassen.

4. Autorisieren Sie die Prozeduren, die unter Ihrem externen Sicherheitsmanager ausgeführt werden sollen.
5. Sie müssen die Bibliothek, die Nachrichten enthält, in die ausgewählte Sprache aufnehmen:
   - thlqual.SCSQSNL E, für amerikanisches Englisch in Groß-/Kleinschreibung
   - thlqual.SCSQSNLU, für amerikanisches Englisch in Großschreibung
   - thlqual.SCSQSNL K, für Japanisch
   - thlqual.SCSQSNL F, für Französisch
   - thlqual.SCSQSNL C, für Chinesisch

Der Kanalinitiator ist ein lang laufender Adressraum. Um die Beendigung zu verhindern, nachdem eine eingeschränkte CPU-Menge verbraucht wurde, bestätigen Sie, dass entweder:

- Der Standardwert für gestartete Tasks in Ihrem z/OS-System ist unbegrenzte CPU-Auslastung; dies wird mit einer JES2-Konfigurationsanweisung für JOBCLASS (STC) mit TIME=(1440,00) erreicht, oder
- Fügen Sie der EXEC-Anweisung für CSQXJST explizit eine TIME=1440-oder TIME=NOLIMIT-Parameter hinzu.

Sie können die Exitbibliothek (CSQXLIB) später zu dieser Prozedur hinzufügen, wenn Sie Kanalexits verwenden wollen. Sie müssen den Kanalinitiator stoppen und erneut starten, um dies zu tun.

Wenn Sie TLS verwenden, ist der Zugriff auf die TLS-Laufzeitbibliothek des Systems erforderlich. Diese Bibliothek heißt SIEALNKE. Die Bibliothek muss APF-autorisiert sein.

Wenn Sie TCP/IP verwenden, muss der Adressraum des Kanalinitiators in der Lage sein, auf die Datei TCPIP.DATA zuzugreifen, die TCP/IP-Systemparameter enthält. Die Art und Weise, in der die Datei konfiguriert werden muss, hängt davon ab, welches TCP/IP-Produkt und welche Schnittstelle Sie verwenden. Dazu gehören:

- Umgebungsvariable, RESOLVER_CONFIG
- /etc/resolv.conf im Dateisystem
- // DD-Anweisung SYSTCPD
- // DD-Anweisung SYSTCPDD
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

Einige davon wirken sich auf die JCL der gestarteten Taskprozedur aus. Weitere Informationen finden Sie unter z/OS Communications Server: IP-Konfigurationshandbuch.

**Zugehörige Konzepte**

„IBM MQ-Subsystem für eine z/OS-WLM-Serviceklasse definieren” auf Seite 986

Um IBM MQ die entsprechende Leistungspriorität im z/OS-System zu erteilen, müssen Sie den Adressräumen des Warteschlangenmanagers und der Kanalinitiatoradresse eine entsprechende WLM-Serviceklasse (z/OS) zuordnen. Wenn Sie dies nicht explizit tun, können unzulässige Standardwerte gelten.

## z/OS *IBM MQ-Subsystem für eine z/OS-WLM-Serviceklasse definieren*

Um IBM MQ die entsprechende Leistungspriorität im z/OS-System zu erteilen, müssen Sie den Adressräumen des Warteschlangenmanagers und der Kanalinitiatoradresse eine entsprechende WLM-Serviceklasse (z/OS) zuordnen. Wenn Sie dies nicht explizit tun, können unzulässige Standardwerte gelten.

- *Wiederholen Sie diese Tasks für jeden IBM MQ-Warteschlangenmanager.*

- *Bei der Migration von einer früheren Version muss diese Task nicht ausgeführt werden.*

Verwenden Sie den ISPF-Dialog, der im Lieferumfang von WLM enthalten ist, um die folgenden Tasks auszuführen:

- Extrahieren Sie die WLM-Richtliniendefinition von z/OS aus der WLM-Koppeldatei
- Aktualisieren Sie diese Richtliniendefinition, indem Sie der ausgewählten Serviceklasse die Namen von Taskprozedurnamen für gestartete Tasks des WS-Managers und des Kanalinitiators
- Die geänderte Richtlinie in der WLM-Koppeldatei installieren

Aktivieren Sie diese Richtlinie dann mit dem Befehl z/OS

```
V WLM,POLICY=policyname,REFRESH
```

Weitere Informationen zur Festlegung von Leistungsoptionen finden Sie im Abschnitt IBM MQ-Umgebung unter z/OS planen.

**Zugehörige Konzepte**

„Konfiguration der Db2-Umgebung” auf Seite 1027
Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie die erforderlichen Db2-Objekte erstellen, indem Sie eine Reihe von Beispieljobs anpassen und ausführen.

### z/OS *Implementieren Sie Ihre ESM-Sicherheitskontrollen.*

Implementieren Sie die Sicherheitssteuerungen für Warteschlangenmanager und den Kanalinitiator.

- *Wiederholen Sie diese Tasks für jeden IBM MQ-Warteschlangenmanager.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*

Wenn Sie RACF als externen Sicherheitsmanager verwenden, lesen Sie die Informationen im Abschnitt Sicherheit auf z/OS einrichten, in der die Implementierung dieser Sicherheitskontrollen beschrieben wird.

Wenn Sie den Kanalinitiator verwenden, müssen Sie außerdem die folgenden Schritte ausführen:

- Wenn für Ihr Subsystem die Verbindungssicherheit aktiv ist, definieren Sie ein Verbindungssicherheitsprofil ssid.CHIN für Ihren externen Sicherheitsmanager (siehe Verbindungssicherheitsprofile für den Kanalinitiator ).
- Wenn Sie TLS (Transport Layer Security) oder eine Socketschnittstelle verwenden, müssen Sie sicherstellen, dass die Benutzer-ID, unter deren Berechtigung der Kanalinitiator ausgeführt wird, für die Verwendung von z/OS UNIX System Serviceskonfiguriert ist, wie in der Dokumentation *z/OS UNIX System Services Planung* beschrieben.
- Wenn Sie TLS verwenden, stellen Sie sicher, dass die Benutzer-ID, unter deren Berechtigung der Kanalinitiator ausgeführt wird, für den Zugriff auf den Schlüsselring konfiguriert ist, der im Parameter SSLKEYR des Befehls ALTER QMGR angegeben ist.

Richten Sie vor dem Start des Warteschlangenmanagers wie folgt die Datei- und Systemsicherheit für IBM MQ ein:

- Autorisieren Sie die Ausführung der Taskprozedur für den Warteschlangenmanager, die unter Ihrem externen Sicherheitsmanager ausgeführt werden soll.
- Autorisieren des Zugriffs auf die WS-Manager-Dateien.
- Konfigurieren Sie bei Bedarf die Verschlüsselung von z/OS-Datasets.

  Weitere Informationen finden Sie im Abschnitt zur Vertraulichkeit für ruhende Daten in IBM MQ for z/OS mit der Dataset-Verschlüsselung. weitere Informationen hierzu.

Weitere Informationen zu diesem Thema finden Sie unter Tasks zur Sicherheitsinstallation für z/OS.

Wenn Sie RACF verwenden, müssen Sie kein IPL (einleitendes Programmladen) für Ihr System ausführen (siehe Abschnitt RACF-Berechtigung für gestartete Taskprozeduren), vorausgesetzt, Sie verwenden die RACF-Klasse STARTED.

**Zugehörige Konzepte**

„SYS1.PARMLIB-Teildateien aktualisieren” auf Seite 988
Um sicherzustellen, dass die Änderungen nach einem IPL wirksam bleiben, müssen Sie einige Member von SYS1.PARMLIB aktualisieren.

„Implementieren Sie die ESM-Sicherheitssteuerelemente für die Gruppe mit gemeinsamer Warteschlange.” auf Seite 1031
Implementieren Sie Sicherheitsmaßnahmen für alle Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange für den Zugriff auf Db2 und die Listenstrukturen der Coupling-Facility.

## z/OS *SYS1.PARMLIB-Teildateien aktualisieren*

Um sicherzustellen, dass die Änderungen nach einem IPL wirksam bleiben, müssen Sie einige Member von SYS1.PARMLIB aktualisieren.

- *Diese Task muss einmal für jedes z/OS-System durchgeführt werden, auf dem IBM MQ ausgeführt werden soll.*
- *Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie sicherstellen, dass die Einstellungen für IBM MQ auf allen z/OS-Systemen in dem Sysplex identisch sind.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*

Aktualisieren Sie die Member SYS1.PARMLIB wie folgt:

1. Aktualisieren Sie das Member IEFSSNss, wie im Abschnitt „IBM MQ-Subsystem für z/OS definieren” auf Seite 980 beschrieben.
2. Ändern Sie IEASYSpp so, dass bei einem IPL die folgenden Member verwendet werden:
   - die Member PROGxx oder IEAAPFaa, die im Abschnitt „APF-Autorisierung der IBM MQ-Ladebibliotheken” auf Seite 974 verwendet werden
   - die Member LNKLSTkk und LPALSTmm, die im Abschnitt „Aktualisieren der z/OS-Linkliste und LPA” auf Seite 975 verwendet werden
   - das Member SCHEDxx, das im Abschnitt „Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften” auf Seite 979 verwendet wird
   - das Member IEFSSNss, das im Abschnitt „IBM MQ-Subsystem für z/OS definieren” auf Seite 980 verwendet wird

**Zugehörige Konzepte**

„Passen Sie die Initialisierungseingabedatensätze an.” auf Seite 988
Erstellen Sie Arbeitskopien der Eingabedatensätze für die Beispielinitialisierung und passen Sie sie an Ihre Systemanforderungen an.

## z/OS *Passen Sie die Initialisierungseingabedatensätze an.*

Erstellen Sie Arbeitskopien der Eingabedatensätze für die Beispielinitialisierung und passen Sie sie an Ihre Systemanforderungen an.

- *Wiederholen Sie diese Tasks für jeden IBM MQ-Warteschlangenmanager.*
- *Sie müssen diese Task ausführen, wenn Sie eine Migration von einer früheren Version durchführen.*

Jeder IBM MQ-Warteschlangenmanager erhält seine Anfangsdefinitionen über eine Reihe von Befehlen, die in den IBM MQ*Initialisierungseingabedateien* enthalten sind. Auf diese Dateien wird durch die Datendefinitionsnamen CSQINP1, CSQINP2und CSQINPT, die in der Prozedur der gestarteten Task des Warteschlangenmanagers definiert sind, verwiesen.

Antworten auf diese Befehle werden in die Initialisierungsausgabedatengruppen geschrieben, auf die durch die DD-Namen CSQOUT1, CSQOUT2 und CSQOUTT verwiesen wird.

Um die Originale zu erhalten, erstellen Sie Arbeitskopien der einzelnen Muster. Anschließend können Sie die Befehle in diesen Arbeitskopien an Ihre Systemanforderungen anpassen.

Wenn Sie über mehrere IBM MQ-Subsysteme verfügen, schließen Sie den Subsystemnamen in das übergeordnete Qualifikationsmerkmal des Initialisierungseingabedateinamens ein, damit Sie das IBM MQ-Subsystem, das jeder Datei zugeordnet ist, einfacher erkennen können.

Weitere Informationen zu den Beispielen finden Sie in den folgenden Abschnitten:

- Initialisierungsdateiformate
- Beispiel CSQINP1 verwenden
- Beispiel CSQINP2 verwenden
- Beispiel CSQINPX verwenden
- Beispiel CSQINPT verwenden

## Formate für Initialisierungsdateien

Bei den Initialisierungseingabedateien kann es sich um partitionierte Dateien (PDS) oder sequenzielle Dateigruppen handeln. Es kann sich um eine verkettete Reihe von Datensätzen handeln. Definieren Sie sie mit einer Satzlänge von 80 Byte. Dabei gilt Folgendes:

- Nur die Spalten 1 bis 72 sind von Bedeutung. Die Spalten 73 bis 80 werden ignoriert.
- Datensätze mit einem Stern (*) in Spalte 1 werden als Kommentare interpretiert und werden ignoriert.
- Leere Sätze werden ignoriert.
- Jeder Befehl muss in einem neuen Datensatz beginnen.
- Ein abschließendes Minuszeichen bedeutet Fortsetzung ab Spalte 1 des nächsten Datensatzes.
- Ein abschließendes Pluszeichen bedeutet Fortsetzung ab der ersten belegten Spalte des nächsten Datensatzes.
- Die maximal zulässige Anzahl an Zeichen in einem Befehl beträgt 32.762.

Bei den Initialisierungsausgabedatengruppen handelt es sich um sequenzielle Datensätze mit einer Satzlänge von 125, einem Satzformat von VBA und einer Blockgröße von 629.

## Beispiel 'CSQINP1' verwenden

Die Datei `thlqual.SCSQPROC` enthält zwei Member, die Definitionen von Pufferpools, Seitengruppen für Pufferpoolzuordnungen und einen ALTER SECURITY-Befehl enthalten.

Member CSQ4INP1 verwendet eine Seitengruppe für jede Nachrichtenklasse. Die Nachrichten werden in die folgenden Klassen unterteilt:

- Systembezogene Nachrichten.
- Wichtige Nachrichten von langer Dauer.
- Nachrichten von kurzer Dauer.
- Sonstige Nachrichten.

Member CSQ4INPR verwendet mehrere Seitengruppen für jede Hauptklasse der Nachrichten und eine Gruppe für jede andere Klasse. Bei den folgenden Klassen handelt es sich um die Hauptklassen der Nachrichten:

- Wichtige Nachrichten von langer Dauer.
- Nachrichten von kurzer Dauer.

Nehmen Sie die entsprechende Stichprobe in die CSQINP1-Verkettung Ihrer gestarteten Taskprozedur Ihres WS-Managers auf.

**Anmerkungen:**

1. IBM MQ unterstützt bis zu 100 Pufferpools im Bereich von 0 bis 99. Der Befehl DEFINE BUFFPOOL kann nur von einer Initialisierungsdatei CSQINP1 ausgegeben werden. Die Definitionen in dem Beispiel geben vier Pufferpools an.

2. Jede Seitengruppe, die vom Warteschlangenmanager verwendet wird, muss in der Initialisierungsdatei CSQINP1 definiert werden, indem der Befehl DEFINE PSID verwendet wird. Die Seitensatzdefinition ordnet eine Pufferpool-ID einer Seitengruppe zu. Wenn kein Pufferpool angegeben ist, wird standardmäßig der Pufferpool null verwendet.

   Seitengruppe Null (00) muss definiert sein. Sie enthält alle Objektdefinitionen. Sie können bis zu 100 Seitengruppen für jeden WS-Manager definieren.
3. Der Befehl ALTER SECURITY kann verwendet werden, um die Sicherheitsattribute TIMEOUT und INTERVAL zu ändern. In CSQ4INP1 werden die Standardwerte als 54 für TIMEOUT und 12 für INTERVAL definiert.

Informationen zum Organisieren von Pufferpools und Seitengruppen finden Sie unter Seitengruppen und Pufferpools planen .

Wenn Sie den Pufferpool und die Seitensattendefinitionen während der Ausführung des Warteschlangenmanagers dynamisch ändern, sollten Sie auch die Definitionen von CSQINP1 aktualisieren. Ist das Attribut REPLACE nicht in der Pufferpooldefinition enthalten, bleiben die Änderungen nur bei einem Kaltstart von IBM MQ erhalten.

## Verwenden der CSQINP2-Beispiele

In dieser Tabelle werden die Member von `thlqual.SCSQPROC` aufgelistet, die in die CSQINP2 -Verkettung Ihrer gestarteten Taskprozedur für den Warteschlangenmanager eingeschlossen werden können, mit einer Beschreibung ihrer Funktion. Die Namenskonvention ist CSQ4IN*. Members von CSQ4INY* sollten für Ihre Konfiguration geändert werden. Sie sollten die CSQINS* -Member nicht ändern, da Sie Änderungen erneut anwenden müssen, wenn Sie auf das nächste Release migrieren. Stattdessen können Sie DEFINE-oder ALTER-Befehle in CSQ4INY * -Teildateien eingeben.

*Tabelle 57. Mitglieder von `thlqual.SCSQPROC`*

| **Mitgliedsname** | **Beschreibung** |
|---|---|
| CSQ4INSG | Systemobjektdefinitionen. |
| CSQ4INSA | Systemobjekt und Standardregeln für die Kanalauthentifizierung. |
| CSQ4INSX | Systemobjektdefinitionen. |
| CSQ4INSS | Passen Sie diese Teildatei an und schließen Sie sie ein, wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden. |
| CSQ4INSJ | Passen Sie dieses Member an und schließen Sie es ein, wenn Sie Publish/Subscribe mit JMS verwenden. |
| CSQ4INSM | Systemobjektdefinitionen für Advanced Message Security. |
| CSQ4INSR | Passen Sie dieses Member an und schließen Sie es ein, wenn Sie WebSphere Application Serveroder die eingereihte Publish/Subscribe-Schnittstelle verwenden, die vom eingereihten Publish/Subscribe-Dämon in IBM MQunterstützt wird. |
| CSQ4DISP | CSQINP2-Beispiel für das Anzeigen von Objektdefinitionen. |
| CSQ4INYC | Clustering-Definitionen. |
| CSQ4INYD | Definitionen verteilter Warteschlangen. |
| CSQ4INYG | Allgemeine Definitionen. |
| CSQ4INYR | Speicherklassendefinitionen, wobei mehrere Seitengruppen für die Hauptklassen der Nachricht verwendet werden. |
| CSQ4INYS | Speicherklassendefinitionen, wobei für jede Nachrichtenklasse eine Seitengruppe verwendet wird. |

Sie müssen nur einmal Objekte definieren, nicht jedes Mal, wenn Sie einen Warteschlangenmanager starten, so dass es nicht erforderlich ist, diese Definitionen jedes Mal in CSQINP2 einzuschließen. Wenn Sie sie jedes Mal einschließen, versuchen Sie, Objekte zu definieren, die bereits vorhanden sind, und Sie erhalten Nachrichten ähnlich der folgenden:

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL' ABNORMAL COMPLETION
```

Die Objekte werden durch diesen Fehler nicht beschädigt. Wenn Sie die Datei SYSTEM-Definitionen in der CSQINP2-Verkettung verlassen wollen, können Sie die Fehlernachrichten vermeiden, indem Sie das Attribut REPLACE für jedes Objekt angeben.

## Verwenden des Beispiels 'CSQINPX'

Beispiel `thlqual.SCSQPROC(CSQ4INPX)` enthält eine Gruppe von Befehlen, die Sie bei jedem Start des Kanalinitiators ausführen können. Dies sind in der Regel kanalbezogene Befehle, wie z. B. START LISTENER, die bei jedem Start des Kanalinitiators erforderlich sind, und nicht jedes Mal, wenn der Warteschlangenmanager gestartet wird und die in den Eingabedatengruppen CSQINP1 oder CSQINP2 nicht zulässig sind. Sie müssen dieses Muster vor der Verwendung anpassen; Sie können es dann in den CSQINPX-Datensatz für den Kanalinitiator aufnehmen.

Die in der Datei enthaltenen IBM MQ-Befehle werden am Ende der Initialisierung des Kanalinitiators ausgeführt; die Ausgabe wird in die in der Anweisung CSQOUTX DD angegebene Datei geschrieben. Die Ausgabe ist ähnlich der Ausgabe, die von der COMMAND-Funktion des IBM MQ-Dienstprogramms (CSQUTIL) generiert wird. Siehe Dienstprogramm CSQUTIL für IBM MQ for z/OSverwenden.

Sie können neben den Kanalbefehlen auch alle IBM MQ-Befehle verwenden, die von CSQUTIL ausgegeben werden können. Sie können Befehle aus anderen Quellen eingeben, während CSQINPX verarbeitet wird. Alle Befehle werden unabhängig vom Erfolg des vorherigen Befehls in der Reihenfolge ausgegeben.

Wenn Sie eine Befehlsantwortzeit angeben möchten, können Sie den Befehl COMMAND für Pseudobefehle als ersten Befehl in der Datei verwenden. Dies erfordert ein einzelnes optionales Schlüsselwort RESPTIME ( *nnn* ), wobei *nnn* die Zeit in Sekunden ist, die auf die Antwort auf die einzelnen Befehle gewartet wird. Dieser Wert liegt im Bereich von 5 bis 999; der Standardwert ist 30.

Stellt IBM MQ fest, dass die Antwort auf vier Befehle zu lange gedauert hat, wird die Verarbeitung von CSQINPX gestoppt und es werden keine weiteren Befehle ausgegeben. Der Kanalinitiator wird nicht gestoppt, aber die Nachricht CSQU052E wird in die CSQOUTX-Datei geschrieben, und die Nachricht CSQU013E wird an die Konsole gesendet.

Wenn IBM MQ die Verarbeitung von CSQINPX erfolgreich abgeschlossen hat, wird die Nachricht CSQU012I an die Konsole gesendet.

## Beispiel 'CSQINPT' verwenden

In dieser Tabelle werden die Mitglieder von `thlqual.SCSQPROC` aufgelistet, die in die CSQINPT-Verkettung Ihrer gestarteten Taskprozedur für den Warteschlangenmanager eingeschlossen werden können, mit einer Beschreibung ihrer Funktion.

*Tabelle 58. Mitglieder von `thlqual.SCSQPROC`*

| Mitgliedsname | Beschreibung |
|---|---|
| CSQ4INST | Standardsubskriptionsdefinition des Systems. |
| CSQ4INYT | Publish/Subscribe-Definitionen. |

Die in der Datei enthaltenen IBM MQ-Befehle werden bei Abschluss der Initialisierung von Publish/Subscribe ausgeführt; die Ausgabe wird in die in der Anweisung CSQOUTT DD angegebene Datei geschrieben. Die Ausgabe ist ähnlich der Ausgabe, die von der COMMAND-Funktion des IBM MQ-Dienstprogramms (CSQUTIL) generiert wird. Siehe Dienstprogramm CSQUTIL für IBM MQ for z/OSverwenden.

**Zugehörige Konzepte**

„Erstellen Sie die Bootstrap-und Protokolldatengruppen.” auf Seite 993
Verwenden Sie das mitgelieferte Programm CSQJU003, um die Bootstrap-Dateigruppen (BSDSs) und die Protokolldatengruppen vorzubereiten.

z/OS *Dataset QMINI*

Mit dem Dataset QMINI können Sie Eigenschaften angeben, die während der Initialisierung des Warteschlangenmanagers gelesen und verarbeitet werden sollen.

## Merkmale des Datasets QMINI

Bei dem Dataset QMINI handelt es sich um ein sequenzielles Dataset mit einer maximalen Satzlänge von 80 Byte (72 Byte für Daten und 8 Byte für die Zeilennummer).

Im folgenden Beispiel werden die Eigenschaften für ein sequenzielles QMINI-Dataset gezeigt. Einige Eigenschaften basieren natürlich auf Ihrer Umgebung.

```
Data Set Name . . . . : QM01.QMINI
General Data                          Current Allocation
Management class . .  : STANDARD      Allocated tracks  . : 1
Storage class  . . .  : STANDARD      Allocated extents . : 1
Volume serial . . .   : P5P21E
Device type . . . .   : 3390
Data class . . . . .  : **None**
Organization  . . .   : PS            Current Utilization
Record format . . .   : FB            Used tracks . . . . : 0
Record length . . .   : 80            Used extents  . . . : 0
Block size  . . . .   : 3120
1st extent tracks .   : 1
Secondary tracks  .   : 1             Dates
Data set name type    :               Creation date . . . : 2020/08/11
Data set encryption   : NO            Referenced date . . : ***None***
                                      Expiration date . . : ***None***
SMS Compressible  .   : NO
```

`thlqual.SCSQPROC`, enthält:

- Die Beispielinhalte eines QMINI-Datasets in CSQ4QMIN.
- Ein Beispiel für die Angabe des QMINI-Datasets mit der //CSQMINI DD-Karte in der Start-JCL des Warteschlangenmanager in den gestarteten Taskprozeduren CSQ4MSTR und CSQ4MSRR.

**Anmerkungen:**

- Der Code, mit dem das Dataset analysiert wird, wertet nur die ersten 72 Byte jedes Datensatzes aus.
- Zeilennummern werden ignoriert, deshalb müssen keine Zeilennummern angegeben werden.
- Wenn eine Zeile mit einem Stern (*) beginnt, wird die Zeile als Kommentar behandelt.
- Die Inhalte des Datasets QMINI werden während des Warteschlangenmanagerstarts analysiert. Wenn die Inhalte erfolgreich analysiert werden, wird die Nachricht CSQM578I in das Jobprotokoll des Warteschlangenmanagers ausgegeben. Wenn während der Analyse Fehler auftreten, werden in das Jobprotokoll des Warteschlangenmanagers Fehlernachrichten wie beispielsweise CSQM573E ausgegeben, aber der Warteschlangenmanager wird trotzdem gestartet.

  Überprüfen Sie, ob Fehlernachrichten vorhanden sind, und beheben Sie mögliche Fehler im QMINI-Dataset.

  Wenn der Warteschlangenmanager das QMINI-Dataset nicht analysieren kann, können Sie den Kanalinitiator starten, aber Sie können keine Kanäle starten, die für die Verwendung von SSL oder TLS konfiguriert sind, da die Einstellungen für die Sicherheitskonfigurationseinstellung unbekannt sind.
- Wenn Sie nach dem Start des Warteschlangenmanager Aktualisierungen am Dataset vornehmen, müssen Sie den Warteschlangenmanager erneut starten, damit die Änderungen übernommen werden.

### Zeilengruppe 'TransportSecurity'

Ab IBM MQ for z/OS 9.2.0 unterstützt das QMINI-Dataset die Zeilengruppe `TransportSecurity`. Diese Zeilengruppe stellt ähnliche Funktionen bereit wie die, die von der SSL-Zeilengruppe in der Datei 'qm.ini' in IBM MQ for Multiplatforms bereitgestellt werden.

Die Zeilengruppe `TransportSecurity` unterstützt die folgenden Eigenschaften:

**AllowTLSV13**
Gibt an, ob ein Warteschlangenmanager die TLS 1.3-CipherSpecs verwenden kann; gültige Werte sind *TRUE/T/YES/Y* oder *FALSE/F/NO/N*.

TLS 1.3 ist für migrierte Warteschlangenmanager nicht standardmäßig aktiviert. Sie können TLS 1.3 aktivieren, indem Sie ein QMINI-Dataset mit der Zeilengruppe `TransportSecurity` definieren und **`AllowTLSV13`**=TRUE festlegen.

Für neu erstellte Warteschlangenmanager ist TLS 1.3 standardmäßig aktiviert.

**AllowedCipherSpecs**
Gibt eine benutzerdefinierte Liste der aktivierten CipherSpecs an.

Weitere Informationen zu dieser Eigenschaft finden Sie im Abschnitt Benutzerdefinierte Liste mit bestellten und aktivieren CipherSpecs auf IBM MQ for z/OS bereitstellen.

Doppelte CipherSpec-Namen in der Liste werden ignoriert.

V 9.3.0 **OutboundSNI**
Gibt an, ob die SNI (Server Name Indication) beim Einleiten einer TLS-Verbindung auf den IBM MQ-Zielkanalnamen des fernen Systems oder auf den Hostnamen gesetzt wird; gültige Werte sind `CHANNEL` oder `HOSTNAME`.

Wenn der Zielkanal mit einer Zertifikatsbezeichnung im Feld CERTLABL des Kanalobjekts konfiguriert ist, müssen Sie CERTLABL auf den Kanalwert setzen. Wenn eine Verbindung mit der Einstellung `HOSTNAME` zu einem Kanal mit einer CERTLABL-Einstellung hergestellt wird, schlägt die Verbindung fehl und eine Nachricht AMQ9673 wird in den Fehlerprotokollen des fernen Warteschlangenmanagers ausgegeben.

Im folgenden Beispiel wird gezeigt, wie die Zeilengruppe `TransportSecurity` angegeben wird:

```
TransportSecurity:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                   ECDHE_RSA_AES_256_GCM_SHA384
AllowTLSV13=TRUE
```

z/OS ***Erstellen Sie die Bootstrap-und Protokolldatengruppen.***

Verwenden Sie das mitgelieferte Programm CSQJU003, um die Bootstrap-Dateigruppen (BSDSs) und die Protokolldatengruppen vorzubereiten.

**Anmerkung:**

- Wiederholen Sie diese Task für jeden IBM MQ-Warteschlangenmanager.
- Wenn Sie die z/OS -Dateiverschlüsselung zum Schützen des BSDS oder der aktiven Protokolldateien verwenden, müssen Sie diese Option konfigurieren, damit die Dateien in diesem Schritt zugeordnet werden.
- Sie müssen diese Task bei der Migration von einer früheren Version nicht ausführen.
- Wenn Sie einen Warteschlangenmanager migrieren und die z/OS -Dateiverschlüsselung für aktive Protokolldateien oder BSDS hinzufügen, müssen Sie die Dateien konvertieren.
- Weitere Informationen zum Konfigurieren der z/OS -Dateiverschlüsselung und zum Konvertieren vorhandener IBM MQ -Dateien für die Verschlüsselung finden Sie unter Vertraulichkeit für ruhende Daten unter IBM MQ for z/OS mit Dateiverschlüsselung.

Die JCL-und AMS-Steueranweisungen (AMS-Access Method Services) für die Ausführung von CSQJU003 zum Erstellen einer einzigen oder doppelten Protokollierungsumgebung werden in thlqual.SCSQPROC

(CSQ4BSDS) gehalten. Passen Sie diesen Job an, und führen Sie diesen Job aus, um BSDSs und Protokolle zu erstellen und die Protokolle vorzuformatieren.

**Wichtig:** Sie sollten die neueste Version von CSQ4BSDS verwenden oder die JCL manuell aktualisieren, um RECORDS (850 60) zu verwenden.

Die Prozedur für gestartete Tasks CSQ4MSTR, die im Abschnitt „Prozeduren für den IBM MQ-Warteschlangenmanager erstellen" auf Seite 984 beschrieben wird, verweist auf BSDSs in Anweisungen im folgenden Format:

```
//BSDS1   DD DSN=++HLQ++.BSDS01,DISP=SHR
//BSDS2   DD DSN=++HLQ++.BSDS02,DISP=SHR
```

Die Protokolldatensätze werden von den BSDSs (BSDSs) bezeichnet.

**Anmerkung:**

1. Der Wert für BLKSIZE muss in der Datendefinitionsanweisung SYSPRINT im Schritt LOGDEF angegeben werden. Der Wert für BLKSIZE muss 629 sein.
2. Um Bootstrap-Datensätze und Protokolldatensätze von verschiedenen WS-Managern zu identifizieren, müssen Sie den Subsystemnamen in das übergeordnete Qualifikationsmerkmal dieser Dateien aufnehmen.
3. Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie die Bootstrap- und Protokolldatengruppen mit SHAREOPTIONS(2 3) definieren.

Informationen zum Planen von Bootstrap- und Protokolldatengruppen und deren Größen finden Sie in Planung für z/OS.

Ab IBM MQ 8.0 verbessert die 8-Byte-Protokoll-RBA-Erweiterung die Verfügbarkeit eines Warteschlangenmanagers, wie in Relative Byteadresse für größere Protokolle beschrieben. Führen Sie die folgenden Schritte aus, nachdem Sie Ihre Protokollierungsumgebung erstellt haben, um 8-Byte-Protokoll-RBA auf einem Warteschlangenmanager vor dem ersten Start des Warteschlangenmanagers zu aktivieren.

**Anmerkung:** V 9.3.0 Für Warteschlangenmanager, die in IBM MQ 9.3.0 oder höher erstellt wurden, ist die 8-Byte-Protokoll-RBA bereits aktiviert. Daher sind die folgenden Schritte nicht erforderlich.

1. Benennen Sie unter Verwendung von **IDCAMS ALTER**die BSDSs im Format der Version 1 (erstellt mit dem Programm CSQJU003 ) in ++HLQ++.V1.BSDS01um.

   **Anmerkung:** Stellen Sie sicher, dass Sie die Daten-und Indexkomponenten sowie den VSAM-Cluster umbenennen.
2. Zuordnen neuer BSDSs mit denselben Attributen wie die bereits definierten. Dies werden die BSDSs der Version 2, die vom WS-Manager beim Start verwendet werden.
3. Führen Sie das BSDS-Konvertierungsdienstprogramm (CSQJUCNV) aus, um das Format BSDSs der Version 1 in das neue Format BSDSs der Version 2 zu konvertieren.
4. Wenn die Konvertierung erfolgreich abgeschlossen wurde, löschen Sie die BSDSs der Version 1.

**Anmerkung:** Wenn sich der Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange befindet, müssen alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange folgendermaßen gestartet werden, bevor die Protokoll-RBA mit einer Länge von 8 Byte aktiviert werden kann:

- Wenn es sich um einen Warteschlangenmanager in IBM MQ 9.0.0 LTS handelt, muss er mit **OPMODE**(NEWFUNC,900) oder **OPMODE**(*NEWFUNC,800*) gestartet worden sein.
- Wenn sich der Warteschlangenmanager auf IBM MQ 9.0.n CDbefindet, oder IBM MQ 9.1.0 LTSoder höher, muss sie auf dieser Ebene gestartet worden sein

**Zugehörige Konzepte**

„Definieren Sie Ihre Seitengruppen" auf Seite 995
Definieren Sie Seitengruppen für jeden Warteschlangenmanager mit einem der bereitgestellten Beispiele.

## z/OS *Definieren Sie Ihre Seitengruppen*

Definieren Sie Seitengruppen für jeden Warteschlangenmanager mit einem der bereitgestellten Beispiele.

- *Wiederholen Sie diese Tasks für jeden IBM MQ-Warteschlangenmanager.*

  *Wenn Sie die Verschlüsselung von z/OS-Datasets zum Schützen von Seitengruppen verwenden, müssen Sie diese Option konfigurieren, bevor die Datasets in diesem Schritt zugeordnet werden.*
- *Bei der Migration von einer früheren Version muss diese Task nicht ausgeführt werden.*

  *Wenn Sie einen Warteschlangenmanager migrieren und die Verschlüsselung von z/OS-Datasets für Seitengruppen hinzufügen, müssen Sie die Seitengruppen umwandeln.*

  *Lesen Sie den Abschnitt* Vertraulichkeit für ruhende Daten in IBM MQ for z/OS mit Dateiverschlüsselung. *Weitere Informationen zur Konfiguration der Verschlüsselung von z/OS -Datasets und zur Konvertierung vorhandener IBM MQ -Datasets für die Verschlüsselung.*

Definieren Sie separate Seitengruppen für jeden IBM MQ-Warteschlangenmanager. thlqual.SCSQPROC(CSQ4PAGE) und thlqual.SCSQPROC(CSQ4PAGR) enthalten Steueranweisungen für JCL und z/OS-Zugriffsmethodenservices (AMS) zum Definieren und Formatieren von Seitengruppen. Der Member CSQ4PAGE verwendet für jede Nachrichtenklasse eine Seitengruppe. Member CSQ4PAGR verwendet mehrere Seitengruppen für die wichtigsten Nachrichtenklassen. Die JCL führt das mitgelieferte Dienstprogrammprogramm CSQUTIL aus. Überprüfen Sie die Muster, und passen Sie sie für die Anzahl der gewünschten Seitengruppen und die zu verwendenden Größen an. Informationen zu Seitengruppen und zur Berechnung geeigneter Größen finden Sie unter Seitengruppen und Pufferpools planen .

Die in „Prozeduren für den IBM MQ-Warteschlangenmanager erstellen” auf Seite 984 beschriebene Prozedur für gestartete Tasks CSQ4MSTR bezieht sich auf die Seitengruppen, mit einer Anweisung in folgendem Format:

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxxx
```

Hierbei steht *nn* für die Nummer der Seitengruppe zwischen 00 und 99, und *xxxxxxxxx* für den Datensatz, den Sie definieren.

**Anmerkung:**

1. Wenn Sie die dynamische Seitenerweiterungsfunktion verwenden möchten, stellen Sie sicher, dass für jede Seitengruppe sekundäre Bereiche definiert sind. thlqual.SCSQPROC (CSQ4PAGE) zeigt, wie dies ausgeführt wird.
2. Um Seitengruppen aus verschiedenen Warteschlangenmanagern zu identifizieren, müssen Sie den Subsystemnamen in das übergeordnete Qualifikationsmerkmal des Datensatzes aufnehmen, der den einzelnen Seitengruppen zugeordnet ist.
3. Wenn die Option FORCE mit der Funktion FORMAT des Dienstprogrammprogramms CSQUTIL verwendet werden soll, müssen Sie das Attribut REUSE in der Anweisung AMS DEFINE CLUSTER hinzufügen.

   Weitere Informationen zu REUSE finden Sie im Abschnitt Optionale Parameter des Befehls z/OS DEFINE CLUSTER.
4. Wenn Ihre Seitengruppen größer als 4 GB sein sollen, müssen Sie die Funktion "Speicher-Management-System (SMS) EXTENDED ADDRESSABILITY" verwenden.

**Zugehörige Konzepte**

„IBM MQ-Einträge zu den Db2-Tabellen hinzufügen” auf Seite 1030
Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, führen Sie das Dienstprogramm CSQ5PQSG aus, um Einträge in Gruppen mit gemeinsamer Warteschlange und in Warteschlangenmanagern den IBM MQ-Tabellen in der Db2-Gruppe mit gemeinsamer Datennutzung hinzuzufügen.

## z/OS *Passen Sie Ihr Systemparametermodul an*

Über das Systemparametermodul von IBM MQ werden die von IBM MQ während des Betriebs verwendeten Umgebungen für Protokollierung, Archivierung, Tracing und Verbindungen gesteuert. Es wird ein Standardmodul bereitgestellt. Sie sollten ein eigenes Systemparametermodul erstellen, da einige Parameter, z. B. Datensatznamen, in der Regel standortspezifisch sind.

- *Führen Sie diese Task bei Bedarf für jeden IBM MQ-Warteschlangenmanager aus.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden. Weitere Informationen finden Sie im Abschnitt* IBM MQ unter z/OS migrieren.
- *Um Advanced Message Security for z/OS in einem vorhandenen WS-Manager zu aktivieren, müssen Sie SPLCAP nur wie in* „CSQ6SYSP verwenden" auf Seite 998 *beschrieben auf YES setzen. Wenn Sie diesen WS-Manager zum ersten Mal konfigurieren, führen Sie die gesamte Task aus.*

Das Systemparametermodul verfügt über vier Makros wie folgt:

| Makroname | Zweck |
|---|---|
| CSQ6SYSP | Gibt die Verbindungs- und Traceerstellungsparameter an (siehe „CSQ6SYSP verwenden" auf Seite 998). |
| CSQ6LOGP | Steuert die Protokollinitialisierung (siehe „CSQ6LOGP verwenden" auf Seite 1008). |
| CSQ6ARVP | Steuert die Archivinitialisierung (siehe „CSQ6ARVP verwenden" auf Seite 1012). |
| CSQ6USGP | Steuert die Nutzungserfassung (siehe „CSQ6USGP verwenden" auf Seite 1019) |

Im Lieferumfang von IBM MQ ist ein Standardsystemparametermodul (CSQZPARM) enthalten, das automatisch aufgerufen wird, wenn Sie den Befehl START QMGR (ohne den Parameter PARM) ausgeben, um eine Instanz von IBM MQ zu starten. CSQZPARM ist in der APF-autorisierten Bibliothek thlqual.SCSQAUTH enthalten, die ebenfalls zusammen mit IBM MQ bereitgestellt wird. Die Werte dieser Parameter werden beim Start von IBM MQ in Form mehrerer Nachrichten angezeigt.

Weitere Informationen zur Verwendung dieses Befehls finden Sie in START QMGR .

### Eigenes Systemparametermodul erstellen

Wenn CSQZPARM die gewünschten Systemparameter nicht enthält, können Sie ein eigenes Systemparametermodul mit Hilfe der in thlqual.SCSQPROC (CSQ4ZPRM) bereitgestellten Beispiel-JCL erstellen.

Gehen Sie wie folgt vor, um ein eigenes Systemparametermodul

1. Erstellen Sie eine Arbeitskopie der JCL-Stichprobe.
2. Bearbeiten Sie die Parameter für jedes Makro in der Kopie nach Bedarf. Wenn Sie alle Parameter aus den Makroaufrufen entfernen, werden die Standardwerte automatisch zur Ausführungszeit übernommen.
3. Ersetzen Sie den Platzhalter ++NAME++ durch den Namen, den das Lademodul einnehmen soll (dies kann CSQZPARM sein).
4. Wenn Ihr Assembler kein High-Level-Assembler ist, ändern Sie die JCL nach Bedarf in Ihrem Assembler.
5. Führen Sie die JCL aus, um die angepassten Versionen der Systemparametermakros zu assemblieren und zu verlinken, um ein Lademodul zu erstellen. Dies ist das neue Systemparametermodul mit dem Namen, den Sie angegeben haben.
6. Laden Sie das Lademodul in eine APF-autorisierte Benutzerbibliothek.
7. Benutzer READ-Zugriff auf die APF-autorisierte Benutzerbibliothek hinzufügen.

8. Fügen Sie diese Bibliothek in die Prozedur für die gestartete Task des IBM MQ-Warteschlangenmanagers (STEPLIB) ein. Dieser Bibliotheksname muss vor der Bibliothek thlqual.SCSQAUTH in STEPLIB stehen.
9. Rufen Sie das neue Systemparametermodul auf, wenn Sie den WS-Manager starten. Wenn das neue Modul beispielsweise den Namen NEWMODS hat, setzen Sie den folgenden Befehl ab:

```
START QMGR PARM(NEWMODS)
```

10. Sicherstellen, dass der Befehl erfolgreich abgeschlossen wurde, indem das Jobprotokoll überprüft wird. Es sollte ein Eintrag im Protokoll ähnlich dem folgenden vorhanden sein:

```
CSQ9022I CDL1 CSQYASCP 'START QMGR' NORMAL COMPLETION
```

Sie können auch den Modulnamen des Parameters in der Start-JCL des Warteschlangenmanagers angeben. Weitere Informationen finden Sie unter MQSC zum Starten und Stoppen eines Warteschlangenmanagers unter z/OSverwenden.

**Anmerkung:** Wenn Sie Ihr Modul CSQZPARM benennen möchten, müssen Sie den Parameter PARM im Befehl START QMGR nicht angeben.

## Feinabstimmung eines Systemparametermoduls

IBM MQ stellt außerdem drei Assemblerquellenmodule bereit, mit denen bereits vorhandene Systemparametermodule optimiert werden können. Diese Module befinden sich in der Bibliothek thlqual.SCSQASMS. In der Regel verwenden Sie diese Module in einer Testumgebung, um die Standardparameter in den Systemparametermakros zu ändern. Jedes Quellenmodul ruft ein anderes Systemparametermakro auf:

| **Dieses Assemblerquellenmodul ...** | **Ruft dieses Makro auf ...** |
|---|---|
| CSQFSYSP | CSQ6SYSP (Verbindungs-und Traceparameter) |
| CSQJLOGP | CSQ6LOGP (Protokollinitialisierung) |
| CSQJARVP | CSQ6ARVP (Archivierungsinitialisierung) |

So verwenden Sie die folgenden Module:

1. Erstellen Sie Arbeitskopien der einzelnen Assemblerquellenmodule in einer Benutzerassemblerbibliothek.
2. Bearbeiten Sie Ihre Kopien, indem Sie die Werte aller Parameter nach Bedarf hinzufügen oder ändern.
3. Assemblieren Sie Ihre Kopien von bearbeiteten Modulen, um Objektmodule in einer Benutzerobjektbibliothek zu erstellen.
4. Verknüpfen Sie diese Objektcodemodule mit einem vorhandenen Systemparametermodul, um ein Lademodul zu erstellen, das das neue Systemparametermodul ist.
5. Stellen Sie sicher, dass das neue Systemparametermodul Mitglied einer vom Benutzer autorisierten Bibliothek ist.
6. Fügen Sie diese Bibliothek in die Prozedur STEPLIB für die gestartete Task des WS-Managers ein. Diese Bibliothek muss vor der Bibliothek thlqual.SCSQAUTH in STEPLIB stehen.
7. Rufen Sie das neue Systemparametermodul mit dem Befehl START QMGR auf, und geben Sie dabei den neuen Modulnamen wie zuvor im Parameter PARM an.

Im Member CSQ4UZPR von SCSQPROC wird ein Beispiel für usermod bereitgestellt. In diesem Beispiel wird gezeigt, wie angepasste Systemparameter unter SMP/E-Steuerung verwaltet werden.

## Ändern von Systemparametern

Sie können einige Systemparameter ändern, während ein Warteschlangenmanager aktiv ist. Weitere Informationen finden Sie in den Abschnitten SET SYSTEM, SET LOGund SET ARCHIVE -Befehle.

Setzen Sie die SET-Befehle in Ihre Initialisierungseingabedatensätze, so dass sie bei jedem Starten des Warteschlangenmanagers wirksam werden.

### Zugehörige Konzepte

„Kanalinitiatorparameter anpassen” auf Seite 1020
Verwenden Sie ALTER QMGR, um den Kanalinitiator an Ihre Anforderungen anzupassen.

z/OS *CSQ6SYSP verwenden*

Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

Die Standardparameter für CSQ6SYSP sowie Informationen, ob die Parameter mit dem Befehl SET SYSTEM geändert werden können, werden in Tabelle 59 auf Seite 998 angezeigt. Wenn Sie einen dieser Werte ändern möchten, lesen Sie die ausführlichen Beschreibungen der Parameter.

*Tabelle 59. Standardwerte für CSQ6SYSP-Parameter*

| Parameter | Beschreibung | Standardwert | SET, Befehl |
|---|---|---|---|
| V 9.3.0 „[MQ 9.3.0 Jul 2021]ACCTIME” auf Seite 1000 | Die Zeit (in Minuten und Sekunden) zwischen jeder Erfassung von Abrechnungsdaten. | -1 | ✓ |
| „ACELIM” auf Seite 1000 | Größe des ACE-Speicherpools in 1-KB-Blöcken. | 0 (kein Grenzwert) | ✓ |
| „CLCACHE” auf Seite 1001 | Gibt den Typ des zu verwendenden Cluster-Cache an. | STATISCH | - |
| „CMDUSER” auf Seite 1001 | Die Standardbenutzer-ID für die Befehlssicherheitsüberprüfung. | CSQOPR | - |
| „EXCLMSG” auf Seite 1001 | Gibt eine Liste der Nachrichten an, die aus jedem Protokoll ausgeschlossen werden sollen. Nachrichten in dieser Liste werden nicht an die z/OS-Konsole und den Protokollausdruck gesendet. In Bezug auf die CPU ist der Parameter EXCLMSG zum Ausschließen von Nachrichten daher effizienter als die im Abschnitt „Informationsnachrichten unterdrücken” auf Seite 1026 beschriebenen Methoden. | ( ) | ✓ |
| „EXITLIM” auf Seite 1002 | Maximale Zeit (in Sekunden), für die Warteschlangenmanagerexits während jedes Aufrufs ausgeführt werden können. | 30 | - |
| „EXITTCB” auf Seite 1002 | Gibt an, wie viele gestartete Server-Tasks zum Ausführen von WS-Managerexits verwendet werden sollen. | 8 | - |
| „LOGLOAD” auf Seite 1002 | Anzahl Protokollsätze, die von IBM MQ zwischen dem Anfang eines Prüfpunkts und dem nächsten geschrieben werden. | 500 000 | ✓ |

| *Tabelle 59. Standardwerte für CSQ6SYSP-Parameter (Forts.)* | | | |
|---|---|---|---|
| **Parameter** | **Beschreibung** | **Standardwert** | **SET, Befehl** |
| „MULCCAPT” auf Seite 1003 | Bestimmt die Eigenschaft 'Gemessene Nutzungspreisgestaltung', die den Algorithmus für die Erfassung von Daten steuert, die von der Messdaten für die gemessene Nutzungslizenz (MULC-Measured Usage License Charging) verwendet | Siehe Parameterbeschreibung | - |
| „OTMACON” auf Seite 1003 | OTMA-Verbindungsparameter. | Siehe Parameterbeschreibung | - |
| „QINDXBLD” auf Seite 1004 | Legt fest, ob der Neustart des Warteschlangenmanagers wartet, bis alle Indizes neu erstellt oder abgeschlossen sind, bevor alle Indizes erneut erstellt werden. | WARTEN | - |
| „QMCCSID” auf Seite 1004 | ID des codierten Zeichensatzes für den WS-Manager. | Null | - |
| „QSGDATA” auf Seite 1004 | Parameter der Gruppe mit gemeinsamer Warteschlange. | Siehe Parameterbeschreibung | - |
| „RESAUDIT” auf Seite 1005 | Prüfparameter RESLEVEL. | JA | - |
| „ROUTCDE” auf Seite 1005 | Nachrichten-Routing-Code, der Nachrichten zugeordnet ist, die nicht von einer bestimmten Konsole angefordert wurden. | 1 | - |
| „SERVICE” auf Seite 1006 | Reserviert für IBM. | 0 | ✓ |
| „SMFACCT” auf Seite 1006 | Gibt an, ob SMF-Abrechnungsdaten beim Starten des Warteschlangenmanagers erfasst werden sollen.<br>Beachten Sie, dass die Kanalabrechnungsdaten der Klasse 4 nur erfasst werden, wenn der Kanalinitiator gestartet wird. | NEIN | - |
| SMFSTAT | Gibt an, ob SMF-Statistiken beim Starten des Warteschlangenmanagers erfasst werden sollen.<br>Beachten Sie, dass die Daten der Kanalinitiatorstatistikdaten der Klasse 4 nur erfasst werden, wenn der Kanalinitiator gestartet wird. | NEIN | - |
| SPLCAP | Gibt an, ob die Warteschlangensicherheitsrichtlinienfunktion in diesem WS-Manager aktiviert ist. Setzen Sie für Advanced Message Security for z/OS diesen Parameter auf YES. | NEIN | - |
| STATIZEIT | V 9.3.0 Die Zeit in Minuten und Sekunden zwischen den einzelnen Zusammenstellen von Statistikdaten. | 30 | ✓ |
| TRACSTR | Gibt an, ob die Tracefunktion automatisch gestartet werden soll. | NEIN | - |

| *Tabelle 59. Standardwerte für CSQ6SYSP-Parameter (Forts.)* | | | |
|---|---|---|---|
| **Parameter** | **Beschreibung** | **Standardwert** | **SET, Befehl** |
| TRACTBL | Größe der Ablaufverfolgungstabelle in 4-KB-Blöcken, die von der globalen Tracefunktion verwendet werden soll. | 99 (396 KB) | ✓ |
| WLMZEIT | Zeit zwischen dem Durchsuchen des Warteschlangenindex für WLM-verwalteten Warteschlangen. | 30 | - |
| WLMTIMU | Einheiten (Minuten oder Sekunden) für WLMTIME. | MINS | - |

V 9.3.0 **ACCTIME**

Gibt das Intervall (in Minuten und Sekunden) zwischen aufeinander folgenden Erfassungen von Abrechnungsdaten an.

Specify a number, either `-1`, or in the range `0` through 1440 minutes in the format `'mmmm'`, or in the range `0` through 1440 minutes, and `0` - 59 seconds, in the format `'mmmm.ss'`.

**Anmerkungen:**

- Wenn Sie nur ein Intervall von Sekunden angeben, müssen Sie das Intervall mit dem Wert `0` voranstellen. Das kleinste mögliche Intervall ist eine Sekunde: '`0.01`'.
- Wenn Sie einen Wert von `0` angeben, werden Abrechnungsdaten im globalen SMF-Aufzeichnungsintervall erfasst. Weitere Informationen finden Sie im Abschnitt Systemmanagementfunktion verwenden.
- Wenn Sie einen Wert von `-1` angeben, bei dem es sich um den Standardwert handelt, werden Abrechnungsdaten in dem Intervall erfasst, das durch den STATIME-Wert angegeben wird.

For example:

`'0.30'` legt ein Intervall von 30 Sekunden fest.
`'5.30'` legt ein Intervall von 5 Minuten und 30 Sekunden fest.
`'30'` legt ein Intervall von 30 Minuten fest.

**ACELIM**

Gibt die maximale Größe des ACE-Speicherpools in 1 KB-Blöcken an. Die Zahl muss im Bereich 0-999999 liegen. Der Standardwert null bedeutet, dass es abgesehen vom verfügbaren Speicherplatz im System keine Einschränkung gibt.

Sie sollten einen Wert für ACELIM nur in Warteschlangenmanagern festlegen, von denen bekannt ist, dass sie außergewöhnlich viel ECSA-Speicher verwenden. Eine Begrenzung des ACE-Speicherpools bewirkt, dass die Anzahl der Verbindungen im System und damit die Größe des von einem Warteschlangenmanager belegten ECSA-Speichers begrenzt wird.

Sobald der Warteschlangenmanager den Grenzwert erreicht, können Anwendungen keine neuen Verbindungen mehr erhalten. Das Fehlen neuer Verbindungen führt zu Fehlern bei der MQCONN-Verarbeitung und bei Anwendungen, die über die Resource Recovery Services (RRS) koordiniert werden, sind Fehler in einer IBM MQ-API wahrscheinlich.

Ein ACE entspricht etwa 12,5 % des Gesamt-ECSA, der innerhalb einer Verbindung für threadspezifische Steuerblöcke benötigt wird. Wenn Sie also beispielsweise ACELIM=5120 angeben, wird erwartet, dass der vom Warteschlangenmanager zugeordnete ECSA-Gesamtumfang (für threadbezogene Steuerblöcke) bei ungefähr 40960K; , also 5120 multipliziert mit 8, begrenzt wird.

Um die vom Warteschlangenmanager zugeordnete Gesamtmenge an ECSA für threadspezifische Steuerblöcke bei 5120Kzu begrenzt, ist ein ACELIM-Wert von 640 erforderlich.

Mit den vom CLASS(3)-Statistiktrace generierten SMF 115-Einträgen des Subtyps 5 können Sie die Größe des 'ACE/PEB'-Speicherpools überwachen und so einen passenden Wert für ACELIM einsetzen.

Aus den vom CLASS(2)-Statistiktrace generierten SMF 115-Einträgen des Subtyps 17 ermitteln Sie dagegen den Gesamt-ECSA-Speicher, den der Warteschlangenmanager insgesamt für Steuerblöcke verwendet. Die Gesamtmenge des verwendeten ECSA-Speichers ist die Summe der Felder QSRSPHBGF und QSRSPHBGV.

Weitere Informationen zu SMF 115-Statistikeinträgen finden Sie unter Interpretieren von IBM MQ-Leistungsstatistiken.

Eine Festlegung von ACELIM sollte als Mechanismus zum Schutz eines z/OS-Images vor einem sich fehlerhaft verhaltenden Warteschlangenmanager und nicht als Mittel zur Steuerung von Anwendungsverbindungen mit einem Warteschlangenmanager eingesetzt werden.

**CLCACHE**
Gibt den Typ des zu verwendenden Cluster-Cache an.

Der Cluster-Cache ist ein Bereich des Speichers, in dem Informationen zum Cluster gespeichert werden.

Wenn der Clustercache statisch ist, hat er eine feste Größe, die beim Start des Warteschlangenmanagers zugeordnet wird. Wenn der Cache voll ist, wird die Nachricht CSQM060E ausgegeben und die Anwendungsanforderung, die mehr Speicherplatz benötigte, erhält den Fehler MQRC_CLUSTER_RESOURCE_ERROR.

Wenn Sie CLCACHE auf dynamisch setzen, kann der Cluster-Cache nach Bedarf erweitert werden. Sie müssen jedoch zuerst sicherstellen, dass alle installierten Exits für Clusterauslastung mit einem dynamischen Cache funktionieren können.

Wenn ein installierter Exit für Clusterauslastung nicht mit einer dynamischen Cachenachricht funktioniert, wird CSQM061E ausgegeben.

MQXCLWLNwird für Exits für Clusterauslastung bereitgestellt, um den Cluster-Cache auf eine Weise zu navigieren, die funktioniert, unabhängig davon, ob dynamische oder statische Caches verwendet werden.

Für neue Warteschlangenmanager setzen Sie CLCACHE=DYNAMIC, es sei denn, Sie verwenden einen Exit für Clusterauslastung, der keinen dynamischen Cache unterstützt.

Für vorhandene Warteschlangenmanager, die bereits einen statischen Cache verwenden und sich in einem Cluster befinden, dem nicht viele neue Warteschlangen und Warteschlangenmanager hinzugefügt wurden, ist es sinnvoll, CLCACHE=STATIC weiterhin zu verwenden.

Verwenden Sie für vorhandene Warteschlangenmanager, die bereits einen statischen Cache verwenden und sich in einem Cluster befinden, dem viele neue Warteschlangen oder Warteschlangenmanager hinzugefügt werden sollen, CLCACHE=DYNAMIC.

**STATISCH**
Wenn der Cluster-Cache statisch ist, wird seine Größe beim Start des Warteschlangenmanagers festgelegt, genug für die aktuelle Menge der Clusterinformationen plus Speicherplatz für den Erweiterungsspeicher. Die Größe kann nicht erhöht werden, während der Warteschlangenmanager aktiv ist. Dies ist die Standardeinstellung.

**DYNAMIC**
Wenn der Clustercache dynamisch ist, kann die beim Start des Warteschlangenmanagers zugeordnete Anfangsgröße bei Bedarf automatisch erhöht werden, solange der Warteschlangenmanager aktiv ist.

**CMDUSER**
Gibt die Standardbenutzer-ID an, die für die Prüfung der Befehlssicherheit verwendet wird Diese Benutzer-ID muss für den ESM definiert sein (zum Beispiel RACF ). Geben Sie einen Namen von 1 bis 8 alphanumerischen Zeichen an. Das erste Zeichen muss alphabetisch sein.

Der Standardwert ist CSQOPR.

**EXCLMSG**
Gibt eine Liste von Fehlernachrichten an, die ausgeschlossen werden sollen.

Diese Liste ist dynamisch und wird mit dem Befehl SET SYSTEM aktualisiert.

Der Standardwert ist eine Liste ohne Inhalt ( ).

Nachrichten werden ohne das Präfix CSQ und ohne das Aktionscode-Suffix (I-D-E-A) bereitgestellt. Fügen Sie beispielsweise X500 zur Liste hinzu, wenn die Nachricht CSQX500I ausgeschlossen werden soll. Die Liste kann maximal 16 Nachrichten-IDs enthalten.

Um für die Aufnahme in die Liste infrage zu kommen, muss die Nachricht nach einem normalen Start der MSTR- oder CHIN-Adressräume ausgegeben werden und mit einem der folgenden Zeichen beginnen: E, H, I, J, L, M, N, P, R, T, V, W, X, Y, 2 ,3, 5, 9.

Nachrichten-IDs, die als Ergebnis von Verarbeitungsbefehlen ausgegeben werden, können der Liste hinzugefügt werden, werden jedoch nicht ausgeschlossen. So wird beispielsweise eine Nachrichten-ID als Ergebnis des Befehls DISPLAY USAGE PSID (*) ausgegeben, diese Nachricht kann jedoch nicht unterdrückt werden.

**EXITLIM**
Gibt die Zeit in Sekunden an, die für jeden Aufruf des Warteschlangenmanagers zulässig ist. (Dieser Parameter hat keine Auswirkung auf Kanalexits.)

Geben Sie einen Wert im Bereich von 5 bis 9999 an.

Der Standardwert ist 30. Der Warteschlangenmanager fragt die Exits ab, die alle 30 Sekunden ausgeführt werden. Bei jeder Abfrage werden alle, die für mehr als die von EXITLIM angegebene Zeit ausgeführt wurden, zwangsweise beendet.

**EXITTCB**
Gibt die Anzahl der gestarteten Server-Tasks an, die für die Ausführung von Exits im Warteschlangenmanager verwendet werden sollen. (Dieser Parameter hat keine Auswirkung auf Kanalexits.) Sie müssen eine Zahl angeben, die mindestens so hoch ist wie die maximale Anzahl Exits (außer Kanalexits), die der Warteschlangenmanager möglicherweise ausführen muss, andernfalls schlägt er mit einem 6c6-Abbruch fehl.

Geben Sie einen Wert im Bereich von 0 bis 99 an. Der Wert null bedeutet, dass keine Exits ausgeführt werden können.

Der Standardwert ist 8.

**LOGLOAD**
Gibt die Anzahl der Protokollsätze an, die IBM MQ zwischen dem Anfang eines Prüfpunkts und dem nächsten schreibt. IBM MQ startet einen neuen Prüfpunkt, nachdem die von Ihnen angegebene Anzahl Datensätze geschrieben wurde.

Es ist ein Wert im Bereich von 200 bis 16 000 000 anzugeben.

Der Standardwert ist 500 000.

Je größer der Wert, desto besser die Leistung von IBM MQ; allerdings dauert ein Neustart länger, wenn der Parameter auf einen hohen Wert gesetzt wird.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | 10 000 |
| **Produktionssystem** | 500 000<br>In einem Produktionssystem kann der angegebene Standardwert zu einer Prüfpunktfrequenz führen, die zu hoch ist. |

Der Wert von LOGLOAD bestimmt die Häufigkeit der WS-Manager-Prüfpunkte. Ein zu großer Wert bedeutet, dass eine große Datenmenge in das Protokoll zwischen Prüfpunkten geschrieben wird, was zu einer weiteren Vorwärtswiederherstellungszeit für den WS-Manager nach einem Fehler führt. Ein zu kleiner Wert bewirkt, dass Prüfpunkte zu häufig in Spitzenlastzeiten auftreten, wodurch die Antwortzeiten und die Prozessorauslastung beeinträchtigt werden.

Für LOGLOAD wird ein Anfangswert von 500 000 vorgeschlagen. Bei einer persistenten Nachrichtenrate von 1 KB von 100 Nachrichten pro Sekunde (d.h. 100 `MQPUT` s mit COMMIT und 100 `MQGET` s mit Commit) beträgt das Intervall zwischen Prüfpunkten ungefähr 5 Minuten.

**Anmerkung:** Dies ist nur als Richtlinie gedacht, und der optimale Wert für diesen Parameter hängt von den Merkmalen des jeweiligen Systems ab.

**MULCCAPT**
Gibt den Algorithmus an, der für die Erfassung von Daten verwendet werden soll, die von Measured Usage License Charging (MULC) verwendet werden.

**STANDARD**
MULC basiert auf der Zeit vom IBM MQ-API-Aufruf MQCONN bis zur Zeit des IBM MQ-API-Aufrufs MQDISC.

**GEHÄNGT**
MULC basiert auf der Zeit vom Start eines IBM MQ-API-Aufrufs bis zum Ende des IBM MQ-API-Aufrufs.

Der Standardwert ist STANDARD.

**OTMACON**
OTMA-Parameter. Für dieses Schlüsselwort werden fünf positionsgebundene Parameter verwendet:

**OTMACON = ( `Group,Member,Druexit,Age,Tpipepfx` )**

**Gruppe**
Dies ist der Name der XCF-Gruppe, zu der diese Instanz von IBM MQ gehört.

Er kann 1 bis 8 Zeichen lang sein und muss in Großbuchstaben eingegeben werden.

Der Standardwert sind Leerzeichen (gibt an, dass IBM MQ kein Mitglied einer XCF-Gruppe werden soll).

**Mitglied**
Dies ist der Mitgliedsname dieser Instanz von IBM MQ innerhalb der XCF-Gruppe.

Er kann 1 bis 16 Zeichen lang sein und muss in Großbuchstaben eingegeben werden.

Der Standardwert ist der Name des 4-stelligen WS-Managers.

**Druexit**
Dies ist der Name des OTMA-Benutzerexits für die Zielauflösung, der von IMS ausgeführt werden soll.

Er kann 1 bis 8 Zeichen lang sein.

Der Standardwert ist DFSYDRU0.

Dieser Parameter ist optional; er ist erforderlich, wenn IBM MQ Nachrichten von einer IMS-Anwendung empfangen soll, die nicht von IBM MQ gestartet wurde. Der Name muss dem DRU-Exit entsprechen, der im IMS-System codiert ist. Weitere Informationen finden Sie in „OTMA-Exits in IMS verwenden” auf Seite 1096.

**Alter**
Dies ist die Zeitspanne in Sekunden, für die eine Benutzer-ID von IBM MQ als zuvor von IMS geprüft betrachtet wird.

Sie kann im Bereich von 0 bis 2 147 483 647 liegen.

Der Standardwert ist 2 147 483 647.

Es wird empfohlen, diesen Parameter in Verbindung mit dem Parameter `interval` des Befehls ALTER SECURITY zu setzen, um die Konsistenz der Sicherheitscacheeinstellungen über den Mainframe-Computer zu gewährleisten.

**Tpipepfx**
Dies stellt das Präfix dar, das für TPipe-Namen verwendet werden soll.

Er besteht aus drei Zeichen; das erste Zeichen befindet sich im Bereich A bis Z, nachfolgende Zeichen sind A bis Z oder 0 bis 9. Der Standardwert ist CSQ.

Dieses Präfix wird immer verwendet, wenn IBM MQ eine Transaktionspipe erstellt; der Rest des Namens wird von IBM MQ zugewiesen. Für eine von IBM MQ erstellte Transaktionspipe können Sie nicht den vollständigen Transaktionspipenamen festlegen.

**QINDXBLD**
Legt fest, ob der Neustart des Warteschlangenmanagers wartet, bis alle Warteschlangenindizes neu erstellt oder abgeschlossen sind, bevor alle Indizes erneut erstellt werden.

**WARTEN**
Der Neustart des Warteschlangenmanagers wartet darauf, dass alle Warteschlangenindexerstellungen abgeschlossen sind. Dies bedeutet, dass während der normalen IBM MQ-API-Verarbeitung keine Anwendungen verzögert werden, solange der Index erstellt wird, da alle Indizes erstellt werden, bevor eine Anwendung eine Verbindung mit dem Warteschlangenmanager herstellen kann.

Dies ist die Standardeinstellung.

**NOWAIT**
Der WS-Manager kann erneut gestartet werden, bevor die Warteschlangenindexerbildung abgeschlossen ist.

**QMCCSID**
Gibt die standardmäßige ID des codierten Zeichensatzes an, die der Warteschlangenmanager (und somit die verteilte Steuerung von Warteschlangen) verwenden soll.

Geben Sie einen Wert im Bereich von 0 bis 65535 an. Der Wert muss eine EBCDIC-Codepage darstellen, die als native z/OS-Codepage für die von Ihnen ausgewählte Sprache in Landessprachen aufgelistet wird.

Null ist der Standardwert; bedeutet, dass die derzeit festgelegte CCSID verwendet wird oder, wenn keine gesetzt ist, die CCSID 500 verwenden. Dies bedeutet, dass Sie, wenn Sie die CCSID explizit auf einen Wert ungleich null setzen, die CCSID nicht zurücksetzen können, indem Sie QMCCSID auf null setzen. Sie müssen jetzt die korrekte CCSID ungleich Null verwenden. Wenn QMCCSID Null ist, können Sie überprüfen, welche CCSID tatsächlich verwendet wird, indem Sie den Befehl DISPLAY QMGR CCSID absetzen.

**Anmerkung:** Alle Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange sollten dieselbe QMCCSID verwenden.

**QSGDATA**
Daten der Gruppe mit gemeinsamer Warteschlange. Für dieses Schlüsselwort werden fünf positionsgebundene Parameter verwendet:

**QSGDATA = ( `Qsgname,Dsgname,Db2name,Db2serv ,Db2blob`)**

**Qsgname**
Dies ist der Name der Gruppe mit gemeinsamer Warteschlange, zu der der Warteschlangenmanager gehört.

Gültige Zeichen finden Sie unter Regeln für die Benennung von IBM MQ-Objekten. Für den Namen gilt Folgendes:

- Kann 1 bis 4 Zeichen lang sein
- Er darf nicht mit einem numerischen Wert beginnen.
- Darf nicht mit @ enden.

  Dies liegt daran, dass aus Implementierungsgründen Namen von weniger als vier Zeichen intern mit @-Symbolen aufgefüllt werden.

Der Standardwert ist Leerzeichen. Dies bedeutet, dass der Warteschlangenmanager kein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist.

**Dsgname**
Dies ist der Name der Db2-Gruppe mit gemeinsamer Datennutzung, mit der der Warteschlangenmanager eine Verbindung herstellen soll.

Er kann 1 bis 8 Zeichen lang sein und muss in Großbuchstaben eingegeben werden.

Der Standardwert ist Leerzeichen. Dies bedeutet, dass keine Gruppen mit gemeinsamer Warteschlange verwendet werden.

**Db2name**
Dies ist der Name des Db2-Subsystems oder -Gruppenanschlusses, mit dem der Warteschlangenmanager eine Verbindung herstellen soll.

Er kann 1 bis 4 Zeichen lang sein und muss in Großbuchstaben eingegeben werden.

Der Standardwert ist Leerzeichen. Dies bedeutet, dass keine Gruppen mit gemeinsamer Warteschlange verwendet werden.

**Anmerkung:** Das Db2 -Subsystem (oder die Gruppenzuordnung) muss sich in der Db2 -Gruppe mit gemeinsamer Datennutzung befinden, die in `Dsgname`angegeben ist, und alle Warteschlangenmanager müssen dieselbe Db2 -Gruppe mit gemeinsamer Datennutzung angeben.

**Db2serv**
Dies ist die Anzahl der Servertasks für den Zugriff auf Db2.

Sie kann im Bereich von 4 bis 10 liegen.

Der Standardwert ist 4.

**Db2blob**
Dies ist die Anzahl der Db2-Tasks für den Zugriff auf große Binärobjekte (BLOBs, Binary Large Objects).

Sie kann im Bereich von 4 bis 10 liegen.

Der Standardwert ist 4.

Wenn Sie nur einen der Namensparameter angeben (also **`Qsgname`**, **`Dsgname`** oder **`Db2name`**), müssen Sie auch für die übrigen Namen Werte eingeben, da IBM MQ andernfalls fehlschlägt.

**RESAUDIT**
Gibt an, ob RACF-Protokolleinträge für RESLEVEL-Sicherheitsprüfungen, die während der Verbindungsverarbeitung ausgeführt werden, geschrieben werden.

Folgende Werte sind möglich:

**NEIN**
Die RESLEVEL-Prüfung wird nicht ausgeführt.

**JA**
Die Prüfung RESLEVEL wird ausgeführt.

Der Standardwert ist YES.

**ROUTCDE**
Gibt den standardmäßigen z/OS-Nachrichtenrouting-Code an, der Nachrichten zugewiesen wird, die nicht als direkte Antwort auf einen WebSphere MQ-Scriptbefehl gesendet werden.

Folgende Werte sind möglich:

1. Ein Wert im Bereich von 1 bis 16 (einschließlich).
2. Eine Liste von Werten, die durch ein Komma getrennt und in runde Klammern eingeschlossen sind. Jeder Wert muss im Bereich von 1 bis 16 (einschließlich) liegen.

Der Standardwert ist 1.

Weitere Informationen zu z/OS -Routing-Codes finden Sie unter *Routing-Codes* in der *Nachrichtenbeschreibung* in einem der Datenträger der *z/OS MVS System Messages* -Handbücher.

**SERVICE**
Dieses Feld ist für IBM reserviert.

**SMFACCT**
Gibt an, ob IBM MQ beim Start des Warteschlangenmanagers automatisch Berechnungsdaten an SMF sendet.

Folgende Werte sind möglich:

**NEIN**
Die Erfassung von Abrechnungsdaten nicht automatisch starten.

**JA**
Die Erfassung von Abrechnungsdaten automatisch für die Standardklasse 1 starten.

**Ganzzahlen**
V 9.3.0 Eine Liste der Klassen, für die Abrechnungsdaten automatisch im Bereich von 1 bis 4 erfasst werden

V 9.3.0 * SMF-Abrechnung für die Klassen 1, 2 und 3 automatisch starten.

Der Standardwert ist NEIN.

**SMFSTAT**
Gibt an, ob SMF-Statistiken beim Start des Warteschlangenmanagers automatisch erfasst werden sollen.

Folgende Werte sind möglich:

**NEIN**
Die Erfassung von Statistikdaten nicht automatisch starten.

**JA**
Beginnen Sie, die Statistikdaten automatisch für die Standardklasse 1 zu erfassen.

**Ganzzahlen**
V 9.3.0 Eine Liste von Klassen, für die Statistikdaten automatisch im Bereich von 1 bis 5 erfasst werden

Für die Erfassung von Statistikdaten der Klasse 2 oder 3 muss auch die Klasse 1 angegeben werden.

V 9.3.0 * SMF-Statistik automatisch für die Klassen 1, 2 und 3 starten.

Der Standardwert ist NEIN.

**SPLCAP**
Die Funktionalität der Sicherheitsrichtlinie ermöglicht eine höhere Nachrichtensicherheitsstufe über Richtlinien, die steuern, ob Nachrichten signiert oder verschlüsselt werden, da sie aus Warteschlangen geschrieben und gelesen werden.

Die Sicherheitsrichtlinienverarbeitung wird für diesen Warteschlangenmanager konfiguriert, indem SPLCAP auf einen der folgenden Werte gesetzt wird:

**NEIN**
Die Funktion zur Implementierung von Richtlinien zur Nachrichtensicherheit für Warteschlangen ist bei der Initialisierung des Warteschlangenmanagers nicht aktiviert.

**JA**
Nachrichtensicherheitsfunktionen werden während der Initialisierung des Warteschlangenmanagers aktiviert.

Der Warteschlangenmanager überprüft, dass das Attribut AMSPROD auf AMS, ADVANCED oder ADVANCEDVUE gesetzt ist. In diesem Fall wurde das Attribut für AMSlizenziert. Andernfalls wird er nicht gestartet.

Der Warteschlangenmanager überprüft auch, ob die erforderliche AMS -Konfiguration vorhanden ist. Ist dies nicht der Fall, wird der Warteschlangenmanager nicht gestartet.

Wenn der Warteschlangenmanager sowohl für AMSlizenziert als auch die erforderliche Konfiguration vorhanden ist, wird der Warteschlangenmanager während der Initialisierung des Warteschlangenmanagers mit aktivierten Nachrichtensicherheitsfunktionen gestartet und der AMSM-Adressraum gestartet.

Der Standardwert ist NEIN.

**STATIZEIT**

V 9.3.0 Gibt ab IBM MQ for z/OS 9.3.0 die Zeit (in Minuten und Sekunden) zwischen aufeinanderfolgenden Zusammenkünften von statistischen Daten an. Wenn ACCTIME nicht gesetzt ist oder `-1` ist, gibt es auch die Zeit zwischen fortlaufenden Erfassungen von Abrechnungsdaten.

Geben Sie eine Zahl im Bereich von `0` bis 1440 Minuten im Format `'mmmm'` oder im Bereich von `0` bis 1440 Minuten und `0` -59 Sekunden im Format `'mmmm.ss'` an. Der Standard ist 30 Minuten.

**Anmerkungen:**

- Wenn Sie nur ein Intervall von Sekunden angeben, müssen Sie das Intervall mit dem Wert `0` voranstellen. Das kleinste mögliche Intervall ist eine Sekunde: `'0.01'`.
- V 9.3.0 Wenn Sie in IBM MQ for z/OS 9.3.0 einen Wert von `0` angeben, werden Statistikdaten in der SMF-Datenerfassungsübertragung erfasst. Wenn ACCTIME nicht angegeben ist oder `-1` ist, werden Abrechnungsdaten auch in der SMF-Datenerfassungsübertragung erfasst. Weitere Informationen finden Sie im Abschnitt Systemmanagementfunktion verwenden.
- Wenn Sie einen Wert von `-1` angeben, bei dem es sich um den Standardwert handelt, werden Abrechnungsdaten in dem Intervall erfasst, das durch den STATIME-Wert angegeben wird.

**TRACSTR**

Gibt an, ob die globale Traceverarbeitung automatisch gestartet werden soll.

Folgende Werte sind möglich:

**NEIN**
Starten Sie die globale Tracefunktion nicht automatisch.

**JA**
Die globale Traceverarbeitung wird automatisch für die Standardklasse 1 gestartet.

**Ganzzahlen**
Eine Liste der Klassen, für die die globale Traceverarbeitung automatisch im Bereich von 1 bis 4 gestartet werden soll.

*****
Starten Sie den globalen Trace automatisch für alle Klassen.

Der Standardwert ist NO, wenn Sie das Schlüsselwort nicht im Makro angeben.

**Anmerkung:** Das angegebene Lademodul des Standardsystemparameters (CSQZPARM) hat TRACSTR=YES (wird im Assemblermodul CSQFSYSP festgelegt). Wenn die Tracefunktion nicht automatisch gestartet werden soll, erstellen Sie entweder ein eigenes Systemparametermodul oder geben Sie den Befehl STOP TRACE nach dem Start des Warteschlangenmanagers aus.

Ausführliche Informationen zum Befehl STOP TRACE finden Sie unter STOP TRACE .

**TRACTBL**

Gibt die Standardgröße (in 4-KB-Blöcken) der Tracetabelle an, in der von der globalen Tracefunktion IBM MQ-Tracesätze gespeichert werden.

Geben Sie einen Wert im Bereich von 1 bis 999 an.

Der Standardwert ist 99. Dies entspricht einer Größe von 396 KB.

**Anmerkung:** Im erweiterten allgemeinen Servicebereich (ECSA = Extended Common Service Area) wird der Tracetabelle Speicherbereich zugeordnet. Aus diesem Grund müssen Sie diesen Wert mit Sorgfalt auswählen.

**WLMZEIT**
Gibt die Zeit (in Minuten oder Sekunden, abhängig vom Wert von WLMTIMU) zwischen den einzelnen Scannen der Indizes für WLM-verwaltete Warteschlangen an.

Geben Sie einen Wert im Bereich von 1 bis 9999 an.

Der Standardwert ist 30.

**WLMTIMU**
Zeiteinheiten, die mit dem Parameter WLMTIME verwendet werden.

Folgende Werte sind möglich:

**MINS**
WLMTIME stellt eine Anzahl von Minuten dar.

**SECS**
WLMTIME stellt eine Anzahl von Sekunden dar.

Der Standardwert ist MINS.

**Zugehörige Verweise**
„CSQ6LOGP verwenden" auf Seite 1008
Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

„CSQ6ARVP verwenden" auf Seite 1012
Verwenden Sie dieses Thema als Referenz für die Angabe Ihrer Archivierungsumgebung mit CSQ6ARVP.

z/OS *CSQ6LOGP verwenden*
Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

Verwenden Sie CSQ6LOGP, um Ihre Protokollierungsoptionen zu erstellen.

Die Standardparameter für CSQ6LOGP und die Angabe, ob Sie jeden Parameter mit dem Befehl SET LOG ändern können, werden in Standardwerte für CSQ6LOGP-Parameter angezeigt. Wenn Sie einen dieser Werte ändern müssen, lesen Sie die ausführlichen Beschreibungen der Parameter.

*Tabelle 60. Standardwerte für CSQ6LOGP-Parameter*

| Parameter | Beschreibung | Standard-wert | SET, Befehl |
|---|---|---|---|
| COMPLOG | Steuert, ob die Protokollkomprimierung aktiviert ist. | KEINE | X |
| DEALLCT | Zeitdauer, die eine Archivierungsbandeinheit nicht verwendet wird, bevor sie nicht zugeordnet wird. | Null | X |
| INBUFF | Größe des Eingabepufferspeichers für aktive Dateien und Archivierungsprotokolldateien. | 60 KB | - |
| MAXARCH | Maximale Anzahl von Archivierungsprotokolldatenträgern, die aufgezeichnet werden können. | 500 | X |
| MAXCNOFF | Maximale Anzahl der parallelen Auslastungstasks (CSQJOFF7), die parallel ausgeführt werden können. | 31 | - |
| MAXRTU | Maximale Anzahl dedizierter Bandeinheiten, die gleichzeitig zum Lesen von Archivprotokollbanddatenträgern zugeordnet sind. | 2 | X |
| OFFLOAD | Archivieren an oder aus. | JA (ON) | - |
| OUTBUFF | Größe des Ausgabepufferspeichers für aktive Dateien und Archivierungsprotokolldateien. | 4 000 KB | - |
| TWOACTV | Einfache oder doppelte aktive Protokollierung. | JA (Dual) | - |

| *Tabelle 60. Standardwerte für CSQ6LOGP-Parameter (Forts.)* | | | |
|---|---|---|---|
| **Parameter** | **Beschreibung** | **Standard-wert** | **SET, Befehl** |
| TWOARCH | Einzel-oder Dual-Archive-Protokollierung. | JA (Dual) | - |
| TWOBSDS | Einzel-oder DoppelBSDS. | JA (duales BSDS) | - |
| WRTHRSH | Die Anzahl der Ausgabepuffer, die gefüllt werden sollen, bevor sie in die aktiven Protokolldateien geschrieben werden. | 20 | X |
| ZHYWRITE | Gibt an, ob die zHyperWrite-Funktion aktiviert ist. | NEIN | X |

**COMPLOG**
Gibt an, ob die Protokollkomprimierung aktiviert ist.

Geben Sie Folgendes an:

**KEINE**
Die Protokollkomprimierung ist nicht aktiviert.

**RLE**
Die Protokollkomprimierung wird unter Verwendung der Lauflängencodierung aktiviert.

**ANY**
Der Warteschlangenmanager wählt den Komprimierungsalgorithmus aus, der den größten Grad der Komprimierung des Protokollsatzes angibt. Diese Option führt zu einer RLE-Komprimierung.

Der Standardwert ist NONE.

Weitere Informationen zur Protokollkomprimierung finden Sie unter Protokollkomprimierung.

**DEALLCT**
Gibt an, wie lange (in Minuten) eine Archivierungslesebandeinheit nicht verwendet werden darf, bevor sie dezugeordnet wird.

Geben Sie eine der folgenden Optionen an:

- Zeit (in Minuten) im Bereich von 0 bis 1440
- NOLIMIT

Die Angabe von 1440 oder NOLIMIT bedeutet, dass die Bandeinheit nie dezugeordnet wird.

Der Standardwert ist null.

Wenn Archivierungsprotokolldaten von Band gelesen werden, wird empfohlen, diesen Wert so hoch zu setzen, dass IBM MQ die Bandbearbeitung für mehrere Leseanwendungen optimieren kann.

**INBUFF**
Gibt die Größe des Eingabepuffers (in Kilobyte) für das Lesen der aktiven und Archivprotokolle während der Wiederherstellung an. Verwenden Sie eine Dezimalzahl im Bereich von 28 bis 60. Der angegebene Wert wird auf ein Vielfaches von 4 aufgerundet.

Der Standardwert ist 60 KB.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | 28 KB |
| **Produktionssystem** | 60 KB<br>Setzen Sie diesen Wert auf das Maximum für die beste Protokollleseleistung. |

**MAXARCH**
Gibt die maximale Anzahl von Archivprotokolldatenträgern an, die im Bootstrap-Data-Set aufgezeichnet werden können. Sobald diese Anzahl überschritten wird, beginnt die Aufzeichnung wieder am Anfang des Bootstrap-Data-Sets.

Verwenden Sie eine Dezimalzahl im Bereich von 10 bis 1000.

Der Standardwert ist 500.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | 500 (Standardwert) |
| **Produktionssystem** | 1 000<br>Setzen Sie diese auf das Maximum, so dass der BSDS so viele Protokolle wie möglich aufzeichnen kann. |

Informationen zu den Protokollen und BSDS finden Sie im Abschnitt IBM MQ-Ressourcen verwalten.

**MAXCNOFF**
Gibt die Anzahl der CSQJOFF7-Auslastungstasks an, die parallel ausgeführt werden können.

Auf diese Weise können Warteschlangenmanager oder Warteschlangenmanager so optimiert werden, dass sie nicht alle verfügbaren Bandeinheiten verwenden.

Stattdessen wartet der Warteschlangenmanager, bis eine Task "CSQJOFF7 offload" abgeschlossen ist, bevor versucht wird, neue Archivierungsdateien zuzuordnen.

Wenn der Warteschlangenmanager auf Band archiviert wird, setzen Sie diesen Parameter so, dass die Anzahl der gleichzeitig ablaufenden Bandanforderungen die Anzahl der verfügbaren Bandeinheiten nicht überschreiten oder überschreiten sollte, da andernfalls das System blockiert sein könnte.

Beachten Sie, dass bei Verwendung der doppelten Archivierung jede Auslastungstask beide Archive ausführt, so dass der Parameter entsprechend festgelegt werden muss. Wenn der Warteschlangenmanager beispielsweise die doppelte Archivierung auf Band hat, würde ein Wert von MAXCNOFF= 2 es ermöglichen, bis zu zwei aktive Protokolle gleichzeitig auf vier Bändern archiviert zu werden.

Wenn mehrere WS-Manager die Bandeinheiten gemeinsam nutzen, sollten Sie für jeden Warteschlangenmanager den Wert für MAXCNOFF festlegen.

Der Standardwert ist 31.

Geben Sie einen Wert im Bereich von 1 bis 31 an.

**MAXRTU**
Gibt die maximale Anzahl dedizierter Bandeinheiten an, die gleichzeitig zugeordnet werden können, um die Banddatenträger des Archivprotokolls gleichzeitig zu lesen.

Dieser Parameter und der Parameter DEALLCT ermöglichen IBM MQ, das Lesen von Archivprotokolldateien von Bandeinheiten zu optimieren.

Geben Sie einen Wert im Bereich von 1 bis 99 an.

Der Standardwert ist 2.

Es wird empfohlen, den Wert auf mindestens einen Wert zu setzen, der kleiner ist als die Anzahl der Bandeinheiten, die für IBM MQ verfügbar sind. Andernfalls könnte der Auslagerungsprozess verzögert werden, was sich auf die Leistung Ihres Systems auswirken könnte. Geben Sie für maximalen Durchsatz bei der Verarbeitung des Archivierungsprotokolls den größtmöglichen Wert für diese Option an, und merken Sie sich, dass Sie mindestens eine Bandeinheit für die Auslastungsverarbeitung benötigen.

**OFFLOAD**
Gibt an, ob die Archivierung ein-oder ausgeschaltet ist.

Geben Sie Folgendes an:

**JA**
Archivierung ist am

**NEIN**
Archivierung ist inaktiviert

Der Standardwert ist YES.

**Achtung:** Schalten Sie **nicht** die Archivierung aus, es sei denn, Sie arbeiten in einer Testumgebung. Wenn Sie diese abschalten, können Sie nicht garantieren, dass die Daten im Falle eines System-oder Transaktionsfehlers wiederhergestellt werden.

**OUTBUFF**
Gibt die Gesamtgröße (in Kilobyte) des Speichers an, der von IBM MQ für Ausgabepuffer zum Schreiben der aktiven und Archivierungsprotokolldateien verwendet werden soll. Jeder Ausgabepuffer ist 4 KB.

Der Parameter muss im Bereich von 128 bis 4000 liegen. Der angegebene Wert wird auf ein Vielfaches von 4 aufgerundet. Werte zwischen 40 und 128 werden aus Kompatibilitätsgründen akzeptiert und werden als Wert von 128 behandelt.

Der Standardwert beträgt 4000 KB.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | 400 KB |
| **Produktionssystem** | 4 000 KB<br>Setzen Sie diesen Wert auf das Maximum, um das Auslaufen von Protokollausgabepuffern zu vermeiden. |

**TWOACTV**
Gibt die einmalige oder doppelte aktive Protokollierung an.

Geben Sie Folgendes an:

**NEIN**
Einzelne aktive Protokolle

**JA**
Doppelte aktive Protokolle

Der Standardwert ist YES.

Weitere Informationen zur Verwendung von einfacher und doppelter Protokollierung finden Sie im Abschnitt IBM MQ-Ressourcen verwalten.

**TWOCHE**
Gibt die Anzahl der Archivprotokolle an, die IBM MQ erstellt, wenn das aktive Protokoll ausgelagert wird.

Geben Sie Folgendes an:

**NEIN**
Einzelne Archivprotokolle

**JA**
Doppelte Archivprotokolle

Der Standardwert ist YES.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | NEIN |
| **Produktionssystem** | YES (Standardwert) |

Weitere Informationen zur Verwendung von einfacher und doppelter Protokollierung finden Sie im Abschnitt IBM MQ-Ressourcen verwalten.

**TWOBSDS**
Gibt die Anzahl der Bootstrap-Dateigruppen an.

Geben Sie Folgendes an:

**NEIN**
Einzelne BSDS

**JA**
DoppelBSDS

Der Standardwert ist YES.

Weitere Informationen zur Verwendung von einfacher und doppelter Protokollierung finden Sie im Abschnitt IBM MQ-Ressourcen verwalten.

**WRTHRSH**
Gibt die Anzahl der 4 KB großen Ausgabepuffer an, die gefüllt werden müssen, bevor sie in die aktiven Protokolldatasets geschrieben werden.

Je höher die Anzahl der Puffer ist, desto weniger Schreibvorgänge finden statt. Dadurch erhöht sich der Durchsatz von IBM MQ. Die Puffer werden möglicherweise geschrieben, bevor diese Zahl erreicht wird, wenn wichtige Ereignisse, wie z. B. ein Commit-Punkt, auftreten.

Geben Sie die Anzahl der Puffer im Bereich von 1 bis 256 an.

Der Standardwert ist 20.

**ZHYWRITE**
Gibt an, ob Schreibvorgänge in aktiven Protokollen mit aktiviertem zHyperWrite erfolgen.

Weitere Informationen zur Aktivierung von aktiven Protokollen mit zHyperWrite finden Sie im Abschnitt zHyperWrite mit aktiven IBM MQ-Protokollen verwenden.

Folgende Werte sind möglich:

**NEIN**
zHyperWrite ist nicht aktiviert.

**JA**
zHyperWrite ist aktiviert.

**Zugehörige Verweise**

„CSQ6SYSP verwenden” auf Seite 998
Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

„CSQ6ARVP verwenden” auf Seite 1012
Verwenden Sie dieses Thema als Referenz für die Angabe Ihrer Archivierungsumgebung mit CSQ6ARVP.

z/OS *CSQ6ARVP verwenden*
Verwenden Sie dieses Thema als Referenz für die Angabe Ihrer Archivierungsumgebung mit CSQ6ARVP.

Verwenden Sie CSQ6ARVP zum Erstellen Ihrer Archivierungsumgebung.

Die Standardparameter für CSQ6ARVP sowie Informationen, ob die Parameter mit dem Befehl SET ARCHIVE geändert werden können, werden in Tabelle 61 auf Seite 1012 angezeigt. Wenn Sie einen dieser Werte ändern müssen, lesen Sie die ausführlichen Beschreibungen der Parameter. Weitere Informationen zur Planung Ihres Speichers finden Sie unter Speicher-und Leistungsanforderungen unter z/OSplanen .

*Tabelle 61. Standardwerte für CSQ6ARVP-Parameter*

| Parameter | Beschreibung | Standardwert | SET, Befehl |
|---|---|---|---|
| ALCUNIT | Einheiten, in denen primäre und sekundäre Bereichszuordnungen vorgenommen werden. | BLK (Blöcke) | X |

| *Tabelle 61. Standardwerte für CSQ6ARVP-Parameter (Forts.)* | | | |
|---|---|---|---|
| **Parameter** | **Beschreibung** | **Standardwert** | **SET, Befehl** |
| ARCPFX1 | Präfix für den Namen des ersten Archivprotokolldatensatzes. | CSQARC1 | X |
| ARCPFX2 | Präfix für den Namen des zweiten Archivprotokolldatensatzes. | CSQARC2 | X |
| ARCRETN | Der Aufbewahrungszeitraum für die Archivprotokolldatei in Tagen. | 9999 | X |
| ARCWRTC | Liste der Leitwegcodes für Nachrichten an den Bediener zum Archivieren von Protokolldatengruppen. | 1,3,4 | X |
| ARCWTOR | Gibt an, ob eine Nachricht an den Bediener gesendet werden soll, und warten Sie, bevor Sie versuchen, eine Archivprotokolldatei anzuhängen. | JA | X |
| blksize | Blockgröße der Archivprotokolldatei. | 28 672 | X |
| Katalog | Gibt an, ob Archivierungsprotokolldatensätze in der ICF katalogisiert werden. | NEIN | X |
| Kompakt | Gibt an, ob Archivierungsprotokolldatensätze kompilierte werden sollen. | NEIN | X |
| PRIQTY | Zuordnung des primären Bereichs für DASD-Dateien. | 25 715 | X |
| PROTECT | Gibt an, ob Archivierungsprotokolldatensätze durch ESM-Profile geschützt werden, wenn die Datensätze erstellt werden. | NEIN | X |
| QUIESCE | Die maximale Zeit in Sekunden, die für die Stilllegung zulässig ist, wenn ARCHIVE LOG mit MODE (QUIESCE) angegeben wurde. | 5 | X |
| SECQTY | Sekundäre Bereichszuordnung für DASD-Datensätze. Siehe Parameter ALCUNIT für die zu verwendenden Einheiten. | 540 | X |
| TSTAMP | Gibt an, ob der Name des Archivdatensatzes eine Zeitmarke enthalten soll. | NEIN | X |
| unit | Einheitentyp oder Einheitenname, auf dem die erste Kopie der Archivierungsprotokolldateien gespeichert ist. | BAND | X |
| UNIT2 | Einheitentyp oder Einheitenname, auf dem die zweite Kopie der Archivierungsprotokolldateien gespeichert ist. | Leer | X |

**ALCUNIT**
Gibt die Einheit an, in der primäre und sekundäre Bereichszuordnungen vorgenommen werden.

Folgende Werte sind möglich:

**CYL**
Zylinder

**TRK**
Spuren

**BLK**
Blöcke

Es wird empfohlen, BLK zu verwenden, da es unabhängig vom Einheitentyp ist.

Der Standardwert ist BLK.

Wenn der freie Speicherbereich auf den DASD-Archivdatenträgern wahrscheinlich fragmentiert ist, sollten Sie einen kleineren primären Extent angeben und die Erweiterung in sekundäre Speicherbereiche zulassen. Weitere Informationen zur Speicherzuordnung für aktive Protokolle finden Sie unter Protokollarchivierungsspeicher planen.

**ARCPFX1**
Gibt das Präfix für den Namen des ersten Archivprotokolldatensatzes an.

Der Parameter TSTAMP enthält eine Beschreibung der Art und Weise, wie die Dateien benannt werden, sowie für die Einschränkungen der Länge von ARCPFX1.

Dieser Parameter darf nicht leer sein.

Der Standardwert ist CSQARC1.

Möglicherweise müssen Sie die Benutzer-ID berechtigen, die dem Adressraum des IBM MQ-Warteschlangenmanagers zugeordnet ist, um Archivprotokolle mit diesem Präfix zu erstellen.

**ARCPFX2**
Gibt das Präfix für den Namen des zweiten Archivprotokolldatensatzes an.

Der Parameter TSTAMP enthält eine Beschreibung der Art und Weise, wie die Dateien benannt werden, sowie die Einschränkungen für die Länge von ARCPFX2.

Dieser Parameter darf auch dann nicht leer sein, wenn der Parameter TWOARCH als NO angegeben ist.

Der Standardwert ist CSQARC2.

Möglicherweise müssen Sie die Benutzer-ID berechtigen, die dem Adressraum des IBM MQ-Warteschlangenmanagers zugeordnet ist, um Archivprotokolle mit diesem Präfix zu erstellen.

**ARCRETN**
Gibt den Aufbewahrungszeitraum in Tagen an, der verwendet werden soll, wenn die Archivprotokolldatei erstellt wird.

Der Parameter muss im Bereich von 0 bis 9999 liegen.

Der Standardwert ist 9999.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | 3<br>In einem Testsystem sind Archivprotokolle über lange Zeiträume wahrscheinlich nicht erforderlich. |
| **Produktionssystem** | 9 999 (Standardwert)<br>Setzen Sie diesen Wert hoch, um die automatische Löschung des automatischen Archivierungsprotokolls wirksam zu inaktivieren |

Weitere Informationen zum Löschen von Archivprotokolldateien finden Sie im Abschnitt Archivprotokolldateien löschen.

**ARCWRTC**
Gibt die Liste der z/OS-Routing-Codes für Nachrichten zu den Archivprotokolldateien an den Operator an. Dieses Feld wird ignoriert, wenn ARCWTOR auf NO gesetzt ist.

Geben Sie bis zu 14 Routing-Codes an, die jeweils einen Wert im Bereich von 1 bis 16 haben. Sie müssen mindestens einen Code angeben. Trennen Sie die Codes in der Liste durch Kommas, nicht durch Leerzeichen.

Der Standardwert ist die Liste der Werte: 1,3,4.

Weitere Informationen zu z/OS -Routing-Codes finden Sie unter *Routing-Codes* in der *Nachrichtenbeschreibung* in einem der Datenträger der *z/OS MVS System Messages* -Handbücher.

**ARCWTOR**

Gibt an, ob eine Nachricht an den Bediener gesendet werden soll und ob eine Antwort empfangen wird, bevor versucht wird, eine Archivprotokolldatei anzuhängen.

Andere IBM MQ-Benutzer müssen möglicherweise warten, bis die Datei bereitgestellt ist, aber sie sind nicht betroffen, solange IBM MQ auf die Antwort auf die Nachricht wartet.

Geben Sie Folgendes an:

**JA**
Die Einheit benötigt eine lange Zeit zum Anhängen von Archivierungsprotokolldatensätzen. Zum Beispiel ein Bandlaufwerk.

**NEIN**
Die Einheit hat keine langen Verzögerungen. Beispiel: DASD.

Der Standardwert ist YES.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | NEIN |
| **Produktionssystem** | YES (Standardwert)<br>Dies hängt von den Betriebsprozeduren ab. Wenn Bandroboter verwendet werden, ist NO möglicherweise besser geeignet. |

**BLKGRÖSS**

Gibt die Blockgröße für die Archivprotokolldatei an. Die von Ihnen angegebene Blockgröße muss mit dem Einheitentyp kompatibel sein, den Sie im Parameter UNIT angeben.

Der Parameter muss im Bereich von 4 097 bis 28 672 liegen. Der Wert, den Sie angeben, wird auf ein Vielfaches von 4 096 aufgerundet.

Der Standardwert ist 28 672.

Dieser Parameter wird von der Blockgröße der Datenklasse für das Storage Management Subsystem (SMS) überschrieben, falls diese bereitgestellt wird

Wenn die Archivprotokolldatei in die DASD-Einheit geschrieben wird, wird empfohlen, die maximale Blockgröße auszuwählen, die zwei Blöcke für jede Spur zulässt. Für eine Einheit IBM 3390 sollten Sie z. B. eine Blockgröße von 24 576 verwenden.

Wenn die Archivprotokolldatei auf Band geschrieben wird, verbessert die Angabe der größtmöglichen Blockgröße die Geschwindigkeit beim Lesen des Archivprotokolls. Sie sollten eine Blockgröße von 28 672 verwenden.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | Verwenden Sie die Empfehlung zur Blockgröße abhängig von den Datenträgern, die für Archivprotokolle verwendet werden.<br>Das heißt, für die Platte 24 576, und das Band 28 672. |
| **Produktionssystem** | Verwenden Sie die Empfehlung zur Blockgröße abhängig von den Datenträgern, die für Archivprotokolle verwendet werden.<br>Das heißt, für die Platte 24 576, und das Band 28 672. |

**CATALOG**
Gibt an, ob Archivierungsprotokolldatensätze im primären ICF-Katalog (ICF = Primary Integrated Catalog Facility) katalogisiert werden.

Geben Sie Folgendes an:

**NEIN**
Archivierungsprotokolldatensätze sind nicht katalogisiert

**JA**
Archivierungsprotokolldatensätze werden katalogisiert

Der Standardwert ist NEIN.

Alle in der DASD-Einheit zugeordneten Archivprotokolldateien müssen katalogisiert werden. Wenn Sie auf der DASD-Einheit mit dem Parameter CATALOG auf NO archivieren, wird die Nachricht CSQJ072E jedes Mal angezeigt, wenn eine Archivprotokolldatei zugeordnet wird, und IBM MQ katalogisiert den Datensatz.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | JA |
| **Produktionssystem** | JA, wenn Archive auf DASD zugeordnet werden |

**COMPACT**
Gibt an, ob Daten, die in Archivprotokolle geschrieben werden, verdichtet werden sollen. Diese Option gilt nur für eine Einheit IBM 3480 oder IBM 3490, die über die Funktion für die verbesserte Datenaufzeichnungsfunktion (IDRC) verfügt. Wenn diese Funktion aktiviert ist, werden die Daten von der Hardware im Bandcontroller mit höherer Schreibdichte als normalerweise üblich geschrieben. Dadurch können mehr Daten auf den Datenträgern gespeichert werden. Geben Sie NO an, wenn Sie kein 3480-Gerät mit der IDRC-Funktion oder einem 3490-Basismodell verwenden, mit Ausnahme des 3490E-Modells. Geben Sie YES an, wenn die Daten kompaktiert werden sollen.

Geben Sie Folgendes an:

**NEIN**
Die Dateigruppen nicht komprimier-

**JA**
Komprimieren Sie die Datensätze.

Der Standardwert ist NEIN.

Die Angabe von YES wirkt sich negativ auf die Leistung aus Beachten Sie außerdem, dass Daten, die auf Band komprimiert sind, nur mit Hilfe einer Einheit gelesen werden können, die die IDRC-Funktion unterstützt. Dies kann ein Problem sein, wenn Sie Archivierungsbänder an einen anderen Standort senden müssen, um eine ferne Wiederherstellung zu erreichen.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | Nicht zutreffend |
| **Produktionssystem** | NO (Standardwert)<br><br>Dies gilt nur für die IDR-Komprimierung 3480 und 3490. Wenn Sie diese Einstellung auf YES setzen, kann die Leseleistung des Archivprotokolls während der Wiederherstellung und des Neustarts beeinträchtigt werden. Dies hat jedoch keine Auswirkungen auf das Schreiben auf Band. |

**PRIQTY**
Gibt die primäre Speicherbereichszuordnung für DASD-Datensätze in ALCUNITs an.

Der Wert muss größer als null sein.

Der Standardwert ist 25 715.

Dieser Wert muss groß genug sein für eine Kopie entweder der Protokolldaten oder des entsprechenden Bootstrap-Data-Sets (je nachdem, was größer ist). Führen Sie die folgenden Schritte aus, um den erforderlichen Wert zu ermitteln:

1. Bestimmen Sie die Anzahl der zugeordneten aktiven Protokollsätze ( c), wie in „Erstellen Sie die Bootstrap-und Protokolldatengruppen.” auf Seite 993erläutert.
2. Ermitteln Sie die Anzahl der 4096-Byte-Blöcke in jedem Archivprotokollblock:

   ```
   d = BLKSIZE / 4096
   ```

   wobei BLKSIZE der aufgerundete Wert ist.
3. Wenn ALCUNIT = BLK:

   ```
   PRIQTY = INT(c / d) + 1
   ```

   wobei INT auf eine ganze Zahl abrundumzurunden ist.

   Wenn ALCUNIT = TRK:

   ```
   PRIQTY = INT(c / (d * INT(e/BLKSIZE))) + 1
   ```

   Dabei steht e für die Anzahl der Byte für jede Spur (56664 für eine Einheit IBM 3390) und INT bedeutet, dass sie auf eine ganze Zahl abgerundet wird.

   Bei ALCUNIT = CYL:

   ```
   PRIQTY = INT(c / (d * INT(e/BLKSIZE) * f)) + 1
   ```

   Dabei steht f für die Anzahl der Spuren für jeden Zylinder (15 für eine Einheit IBM 3390), und INT bedeutet, dass er auf eine ganze Zahl abgerundet wird.

Informationen zur erforderlichen Größe Ihrer Protokoll- und Archivdateien finden Sie in den Abschnitten „Erstellen Sie die Bootstrap-und Protokolldatengruppen.” auf Seite 993 und „Definieren Sie Ihre Seitengruppen” auf Seite 995.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | 1 680<br>Ausreichend, um das gesamte aktive Protokoll aufzunehmen, d. a.:<br>`10  080 / 6 = 1  680 blocks` |
| **Produktionssystem** | Nicht anwendbar, wenn die Archivierung auf Band durchgeführt wird. |

Wenn der freie Speicherbereich auf den DASD-Archivdatenträgern wahrscheinlich fragmentiert ist, sollten Sie einen kleineren primären Extent angeben und die Erweiterung in sekundäre Speicherbereiche zulassen. Weitere Informationen zur Speicherplatzzuordnung für aktive Protokolle finden Sie unter Protokollarchivierungsspeicher planen.

**PROTECT**
Gibt an, ob Archivierungsprotokolldatensätze durch diskrete ESM-Profile (ESM = External Security Manager) geschützt werden, wenn die Datensätze erstellt werden.

Geben Sie Folgendes an:

**NEIN**
Profile werden nicht erstellt.

**JA**
Es werden diskrete Dateigruppe-Profile erstellt, wenn Protokolle ausgelagert werden. Wenn Sie YES angeben:

- Der ESM-Schutz muss für IBM MQ aktiv sein.
- Die Benutzer-ID, die dem Adressraum des IBM MQ-Warteschlangenmanagers zugeordnet ist, muss die Berechtigung zum Erstellen dieser Profile haben.
- Die Klasse TAPEVOL muss aktiv sein, wenn die Archivierung auf Band erfolgt.

Andernfalls schlägt die Ausladung fehl.

Der Standardwert ist NEIN.

**QUIESCE**
Gibt die maximale Zeit in Sekunden an, die für die Stilllegung zulässig ist, wenn ein ARCHIVE LOG-Befehl mit der Angabe MODE (QUIESCE) ausgegeben wird.

Der Parameter muss im Bereich von 1 bis 999 liegen.

Der Standardwert ist 5.

**SECQTY**
Gibt die sekundäre Bereichszuordnung für DASD-Datensätze in ALCUNITs an. Der sekundäre Speicherbereich kann bis zu 15 Mal zugeordnet werden. Weitere Informationen zu ALCUNIT enthält das Handbuch IBM z/OS Management Facility Programming Guide .

Der Parameter muss größer als null sein.

Der Standardwert ist 540.

**TSTAMP**
Gibt an, ob der Name des Archivprotokolldatensatzes eine Zeitmarke in ihm enthält.

Geben Sie Folgendes an:

**NEIN**
Namen enthalten keine Zeitmarke. Die Archivprotokolldateien werden wie folgt benannt:

```
arcpfxi.A nnnnnnn
```

Dabei ist *arcpfxi* das durch ARCPFX1 oder ARCPFX2 angegebene Präfix für die Dateigruppe. *arcpfxi* kann bis zu 35 Zeichen lang sein.

**JA**
Die Namen enthalten einen Zeitstempel. Die Archivprotokolldateien werden wie folgt benannt:

```
arcpfxi.cyyddd.T hhmmsst.A nnnnnnn
```

Dabei ist *c* 'D' für die Jahre bis einschließlich 1999 bzw. 'E' für die Jahre ab dem Jahr 2000, und *arcpfxi* ist das bei ARCPFX1 bzw. ARCPFX2 angegebene Dateinamenspräfix. *arcpfxi* kann bis zu 19 Zeichen lang sein.

**EXT**
Die Namen enthalten einen Zeitstempel. Die Archivprotokolldateien werden wie folgt benannt:

```
arcpfxi.D yyyyddd.T hhmmsst.A nnnnnnn
```

Dabei ist *arcpfxi* das durch ARCPFX1 oder ARCPFX2 angegebene Präfix für die Dateigruppe. *arcpfxi* kann aus maximal 17 Zeichen bestehen.

Der Standardwert ist NEIN.

**UNIT**
Gibt den Einheitentyp oder den Einheitennamen der Einheit an, die zum Speichern der ersten Kopie der Archivprotokolldatei verwendet wird.

Geben Sie einen Einheitentyp oder einen Einheitennamen mit 1 bis 8 alphanumerischen Zeichen an. Das erste Zeichen muss alphabetisch sein.

Dieser Parameter darf nicht leer sein.

Der Standardwert ist TAPE.

Wenn Sie Daten auf DASD archivieren, können Sie einen generischen Einheitentyp mit einem begrenzten Datenträgerbereich angeben, z. B. UNIT = 3390.

Wenn Sie auf DASD archivieren, stellen Sie Folgendes sicher:

- Die primäre Bereichszuordnung ist groß genug, um alle Daten aus den aktiven Protokolldatengruppen aufzunehmen.
- Die Katalogoption des Archivprotokolldatensatzes (CATALOG) ist auf YES gesetzt.
- Sie haben einen korrekten Wert für BLKSIZE verwendet.

Wenn Sie auf Band archivieren, kann IBM MQ sich auf maximal 20 Datenträger verteilen.

Empfohlene Einstellungen:

| | |
|---|---|
| **Testsystem** | DASD |
| **Produktionssystem** | BAND |

Weitere Informationen zur Auswahl einer Position für Archivprotokolle finden Sie unter Protokollarchivierungsspeicher planen.

**UNIT2**
Gibt den Einheitentyp oder den Einheitennamen der Einheit an, die zum Speichern der zweiten Kopie der Archivierungsprotokolldatensätze verwendet wird.

Geben Sie einen Einheitentyp oder einen Einheitennamen mit 1 bis 8 alphanumerischen Zeichen an. Das erste Zeichen muss alphabetisch sein. Wenn dieser Parameter leer ist, wird der für den Parameter UNIT festgelegte Wert verwendet.

Der Standardwert ist leer.

**Zugehörige Verweise**
„CSQ6SYSP verwenden” auf Seite 998
Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

„CSQ6LOGP verwenden” auf Seite 1008
Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

z/OS *CSQ6USGP verwenden*
Verwenden Sie dieses Thema als Referenz für die Verwendung von CSQ6USGP zum Festlegen der Systemparameter.

Verwenden Sie CSQ6USGP, um die Aufzeichnung der Produktverwendung zu steuern.

Die Standardparameter für CSQ6USGP werden in Tabelle 62 auf Seite 1020 angezeigt. Wenn Sie einen dieser Werte ändern müssen, lesen Sie die ausführlichen Beschreibungen der Parameter.

**Achtung:** Sie können keinen dieser Parameter mit dem Befehl SET SYSTEM ändern.

| *Tabelle 62. Standardwerte für CSQ6USGP-Parameter* | | |
|---|---|---|
| **Parameter** | **Beschreibung** | **Standardwert** |
| QMGRPROD | Produkt, für das die Verwendung des Warteschlangenmanagers aufgezeichnet werden soll | Leer |
| AMSPROD | Produkt, für das die Verwendung von Advanced Message Security (AMS) aufgezeichnet werden soll | Leer |

**QMGRPROD**
Gibt das Produkt an, für das die Verwendung des Warteschlangenmanagers aufgezeichnet werden soll.

Folgende Werte sind möglich:

**MQ**
Die Verwendung des Warteschlangenmanagers wird als eigenständiges IBM MQ for z/OS-Produkt mit der Produkt-ID 5655-MQ9 aufgezeichnet.

**VUE**
Die Verwendung des Warteschlangenmanagers wird als eigenständiges IBM MQ for z/OS Value Unit Edition (VUE)-Produkt mit der Produkt-ID 5655-VU9 aufgezeichnet.

**ADVANCEDVUE**
Die Verwendung des Warteschlangenmanagers wird als Teil eines IBM MQ Advanced for z/OS Value Unit Edition-Produkts mit der Produkt-ID 5655-AV1 aufgezeichnet.

**AMSPROD**
Wenn dieser Parameter nicht festgelegt ist, wird der AMS-Adressraum nicht gestartet, und die Nachricht CSQY024I wird ausgegeben.

Gibt das Produkt an, für das die Verwendung von Advanced Message Security aufgezeichnet werden soll, wenn es verwendet wird.

Folgende Werte sind möglich:

**AMS**
Die AMS-Verwendung wird als eigenständiges Advanced Message Security for z/OS-Produkt mit der Produkt-ID 5655-AM9 aufgezeichnet.

**ADVANCED**
Die AMS-Verwendung wird als Teil eines IBM MQ Advanced for z/OS-Produkts mit der Produkt-ID 5655-AV9 aufgezeichnet.

**ADVANCEDVUE**
Die AMS-Verwendung wird als Teil eines IBM MQ Advanced for z/OS Value Unit Edition-Produkts mit der Produkt-ID 5655-AV1 aufgezeichnet.

Weitere Informationen zur Aufzeichnung der Produktnutzung finden Sie unter Produktinformationen melden.

**Zugehörige Verweise**
„CSQ6SYSP verwenden” auf Seite 998
Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

„CSQ6LOGP verwenden” auf Seite 1008
Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

## z/OS *Kanalinitiatorparameter anpassen*

Verwenden Sie ALTER QMGR, um den Kanalinitiator an Ihre Anforderungen anzupassen.

- *Führen Sie diese Task bei Bedarf für jeden IBM MQ-Warteschlangenmanager aus.*
- *Diese Task muss bei der Migration von einer früheren Version ausgeführt werden.*

Eine Reihe von Warteschlangenmanagerattributen steuert, wie die verteilte Steuerung von Warteschlangen ausgeführt wird. Legen Sie diese Attribute mit dem WebSphere MQ-Scriptbefehl ALTER QMGR fest. Das Beispiel für die Initialisierungsdatengruppe thlqual.SCSQPROC (CSQ4INYG) enthält einige Einstellungen, die Sie anpassen können. Weitere Informationen finden Sie in ALTER QMGR .

Die Werte dieser Parameter werden bei jedem Starten des Kanalinitiators als Folge von Nachrichten angezeigt.

## Die Beziehung zwischen Adaptern, Dispatchern und der maximalen Anzahl an Kanälen

Die ALTER QMGR-Parameter CHIADAPS und CHIDISPS definieren die Anzahl der Tasksteuerblöcke (Task Control Blocks, TCBs), die vom Kanalinitiator verwendet werden. CHIADAPS (Adapter) TCBs werden verwendet, um IBM MQ-API-Aufrufe an den Warteschlangenmanager zu machen. CHIDISPS (Dispatcher) TCBs werden verwendet, um Anrufe an das Kommunikationsnetz zu tätigen.

Der Parameter MAXCHL des Befehls ALTER QMGR wirkt sich auf die Verteilung von Kanälen auf die Dispatcher-TCBs aus.

**CHIDISPS**

Wenn Sie eine kleine Anzahl von Kanälen verwenden, verwenden Sie den Standardwert.

Eine Task für jeden Prozessor optimiert die Systemleistung. Da die Dispatcher-Tasks CPU-intensiv sind, besteht das Prinzip darin, so wenig Aufgaben wie möglich zu halten, so dass die Zeit zum Suchen und Starten von Threads minimiert wird.

CHIDISPS (20) ist für Systeme mit mehr als 100 Kanälen geeignet. Es ist unwahrscheinlich, dass es in der Anwendung von CHIDISPS (20), wo dies mehr Dispatcher-TCBs ist, als notwendig ist, ein erheblicher Nachteil ist.

Wenn Sie über mehr als 1000 Kanäle verfügen, können Sie als Richtlinie einen Dispatcher für alle 50 aktuellen Kanäle zulassen. Geben Sie z. B. CHIDISPS (40) an, um bis zu 2000 aktive Kanäle zu verarbeiten.

Wenn Sie TCP/IP verwenden, beträgt die maximale Anzahl an Dispatchern, die für TCP/IP-Kanäle verwendet werden, 100, selbst wenn Sie einen größeren Wert in CHIDISPS angeben.

**CHIADAPS**

Jeder IBM MQ-API-Aufruf an den Warteschlangenmanager ist unabhängig von einem anderen und kann auf einem beliebigen Adapter-TCB ausgeführt werden. Aufrufe, die persistente Nachrichten verwenden, können aufgrund der Protokoll-E/A viel länger dauern als die für nicht persistente Nachrichten. So kann eine Kanalinitiatorverarbeitung, die eine große Anzahl persistenter Nachrichten über viele Kanäle hinweg verarbeitet, mehr als die Standard-8-Adapter-TCBs für eine optimale Leistung benötigen. Dies ist besonders dann der Fall, wenn die Batchgröße klein ist, da das Ende der Batchverarbeitung auch die Protokoll-E/A erfordert und die Thin-Client-Kanäle verwendet werden.

Der vorgeschlagene Wert für eine Produktionsumgebung lautet CHIADAPS (30). Die Verwendung von mehr als dem ist wahrscheinlich keinen nennenswerten zusätzlichen Nutzen zu bringen, und es ist unwahrscheinlich, dass es einen erheblichen Nachteil in der Verwendung von CHIADAPS (30) hat, wenn dies mehr Adapter-TCBs als notwendig ist.

**MAXCHL**

Jeder Kanal wird einem bestimmten Dispatcher-TCB beim Kanalstart zugeordnet und bleibt diesem TCB zugeordnet, bis der Kanal gestoppt wird. Jeder TCB kann von vielen Kanälen gemeinsam genutzt werden. MAXCHL wird verwendet, um Kanäle über die verfügbaren Dispatcher-TCBs zu verteilen. Die ersten Kanäle ( MIN( (MAXCHL / CHIDISPS ) , 10 ) ), die gestartet werden müssen, werden dem ersten Dispatcher-TCB zugeordnet usw., bis alle Dispatcher-TCBs in Gebrauch sind.

Die Auswirkung dieser Option auf eine kleine Anzahl von Kanälen und ein großes MAXCHL-Wert ist, dass die Kanäle nicht gleichmäßig auf die Dispatcher verteilt sind. Wenn Sie beispielsweise CHIDISPS (10) festlegen und MAXCHL mit dem Standardwert von 200 belassen, aber nur 50 Kanäle hatten, würden fünf Dispatcher mit jeweils 10 Kanälen verbunden sein, fünf würden jedoch nicht verwendet.

Es wird empfohlen, MAXCHL auf die Anzahl der Kanäle einzustellen, die tatsächlich verwendet werden sollen, wenn es sich um eine kleine feste Zahl handelt.

Wenn Sie diese Eigenschaft des Warteschlangenmanagers ändern, müssen Sie auch die Eigenschaften ACTCHL, LU62CHL und TCPCHL in den WS-Managern überprüfen, um sicherzustellen, dass die Werte kompatibel sind. Eine vollständige Beschreibung dieser Eigenschaften und ihrer Beziehungen finden Sie unter Parameter des Warteschlangenmanagers .

## z/OS UNIX System Services-Umgebung für Kanalinitiatoren einrichten

Der Kanalinitiator (CHINIT) verwendet OMVS-Threads. Überprüfen Sie die OMVS-Konfigurationsparameter, bevor Sie einen neuen CHINIT erstellen, oder ändern Sie die Anzahl der Dispatcher oder SSLTASKS.

Jeder CHINIT verwendet 3 + CHIDISP + SSLTASKS OMVS-Threads. Diese Beiträge tragen zur Gesamtzahl der in der logischen Partition verwendeten OMVS-Threads und zur Anzahl der Threads bei, die von der gestarteten Task-Benutzer-ID von CHINIT verwendet werden.

Sie können die **D OMVS,L** verwenden und die aktuelle Nutzung, die Hochwassernutzung und den Systemgrenzwert von MAXPROCSYS (die maximale Anzahl von Prozessen, die das System zulässt) überprüfen.

Wenn Sie einen neuen CHINIT hinzufügen oder die Werte von CHIDISPS oder SSLTASKS erhöhen, müssen Sie die Zunahme der Threads berechnen und die Auswirkungen auf die MAXPROCSYS-Werte überprüfen. Mit dem Befehl **SETOMVS** können Sie MAXPROCSYS dynamisch ändern und/oder den parmlib-Wert BPXPRCxx aktualisieren.

Der OMVS-Parameter MAXPROCUSER ist die Anzahl der OMVS-Threads, die ein einzelner OMVS-Benutzer mit derselben UID haben kann. Die Threads zählen zu diesem Wert. Wenn Sie also 2 CHINITS mit derselben gestarteten Task-Benutzer-ID mit 10 Dispatchern und 3 SSLTASKS haben, dann gibt es 2 * (3 + 10 + 3) = 32 Threads für die OMVS-UID.

Sie können den Standardwert für MAXPROCUSER anzeigen, indem Sie den Befehl **D OMVS,O** absetzen, und Sie können den Befehl **SETOMVS** verwenden, um den Wert für MAXPROCUSER dynamisch zu ändern und/oder den parmlib-Wert für BPXPRCxx zu aktualisieren.

Sie können diesen Wert auf Benutzerbasis mit dem RACF -Befehl **ALTUSER userid OMVS(PROCUSERMAX(nnnn))** oder einem entsprechenden Wert überschreiben.

Geben Sie den folgenden Befehl aus, um den Kanalinitiator zu starten:

```
START CHINIT
```

Um sicherzustellen, dass der Kanalinitiator erfolgreich gestartet wurde, stellen Sie sicher, dass im Jobprotokoll xxxxCHIN (ssidCHIN) kein ICH408I-Fehler aufgetreten ist.

**Zugehörige Konzepte**

„Batch-, TSO-und RRS-Adapter konfigurieren” auf Seite 1022
Stellen Sie die Adapter für Anwendungen zur Verfügung, indem Sie Bibliotheken zu entsprechenden STEPLIB-Verkettungen hinzufügen. Um SNAP-Speicherauszüge, die von einem Adapter ausgegeben werden, zu erfüllen, ordnen Sie einen CSQSNAP-DDnamen zu. Sie können CSQBDEFV verwenden, um die Portierbarkeit Ihrer Anwendungsprogramme zu verbessern.

**Zugehörige Verweise**

Kanalinitiatorstatistikdatensätze

## z/OS *Batch-, TSO-und RRS-Adapter konfigurieren*

Stellen Sie die Adapter für Anwendungen zur Verfügung, indem Sie Bibliotheken zu entsprechenden STEPLIB-Verkettungen hinzufügen. Um SNAP-Speicherauszüge, die von einem Adapter ausgegeben werden, zu erfüllen, ordnen Sie einen CSQSNAP-DDnamen zu. Sie können CSQBDEFV verwenden, um die Portierbarkeit Ihrer Anwendungsprogramme zu verbessern.

- *Führen Sie diese Task je nach Bedarf für jeden einzelnen IBM MQ-Warteschlangenmanager durch.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*

Damit Adapter für Stapelanwendungen sowie andere Anwendungen, die Stapelverbindungen verwenden, verfügbar sind, müssen Sie der STEPLIB-Verkettung für Ihre Anwendung die folgenden IBM MQ-Bibliotheken hinzufügen:

- thlqual.SCSQANL *x*
- thlqual.SCSQAUTH

Hierbei steht *x* für den Sprachenbuchstabe für Ihre Landessprache. (Sie müssen dies nicht tun, wenn sich die Bibliotheken im LPA oder in der Linkliste befinden.)

Fügen Sie für TSO-Anwendungen die Bibliotheken zur STEPLIB-Verkettung in die TSO-Anmeldeprozedur hinzu, oder aktivieren Sie sie mit dem TSO-Befehl TSOLIB.

Wenn der Adapter einen unerwarteten IBM MQ-Fehler feststellt, gibt er einen z/OS-Kurzspeicherauszug an den DDnamen CSQSNAP aus und gibt den Ursachencode `MQRC_UNEXPECTED_ERROR` an die Anwendung aus.Wenn die DD-Anweisung CSQSNAP nicht in der Anwendungs-JCL enthalten ist oder CSQSNAP keinem Datensatz zugeordnet ist, der unter TSO gesetzt ist, wird kein Speicherauszug erstellt. In diesem Fall könnten Sie die DD-Anweisung CSQSNAP in die Anwendungs-JCL aufnehmen oder CSQSNAP einem unter TSO festgelegten Datensatz zuordnen und die Anwendung erneut ausführen. Da jedoch einige Probleme sporadisch auftreten, wird empfohlen, dass Sie eine Anweisung CSQSNAP in die Anwendungs-JCL aufnehmen oder CSQSNAP einem Datensatz in der TSO-Anmeldeprozedur zuordnen, um die Ursache für den Fehler zu dem Zeitpunkt zu erfassen, zu dem sie auftritt.

Das mitgelieferte Programm CSQBDEFV verbessert die Portierbarkeit Ihrer Anwendungsprogramme. In CSQBDEFV können Sie den Namen eines Warteschlangenmanagers oder einer Gruppe mit gemeinsamer Warteschlange für die Herstellung der Verbindung angeben, anstatt ihn im Aufruf `MQCONN` oder `MQCONNX` in einem Anwendungsprogramm anzugeben. Sie können für jeden Warteschlangenmanager und jede Gruppe mit gemeinsamer Warteschlange eine neue Version von CSQBDEFV erstellen. Führen Sie dazu die folgenden Schritte aus:

1. Kopieren Sie das IBM MQ-Assemblerprogramm CSQBDEFV aus thlqual.SCSQASMS in eine Benutzerbibliothek.
2. Das angegebene Programm enthält den Standardsubsystemnamen CSQ1. Sie können diesen Namen für Test-und Installationsprüfung beibehalten. Bei Produktionssubsystemen können Sie NAME=CSQ1 in den Namen Ihres 1-bis Vier-Zeichen-Subsystems ändern oder CSQ1 verwenden.

   Bei der Verwendung von Gruppen mit gemeinsamer Warteschlange können Sie anstelle von CSQ1 den Namen einer Gruppe mit gemeinsamer Warteschlange angeben. Wenn Sie dies tun, gibt das Programm eine Verbindungsanforderung an einen aktiven Warteschlangenmanager in dieser Gruppe aus.
3. Assemblieren und verbinden Sie das Programm, um das CSQBDEFV-Lademodul zu erstellen. Geben Sie für die Assemblierung die Bibliothek thlqual.SCSQMACS in Ihrer SYSLIB-Verkettung an; verwenden Sie die Parameter für die Linkeditierung `RENT`, `AMODE=31`, `RMODE=ANY`. Dies wird in der Beispiel-JCL in thlqual.SCSQPROC (CSQ4DEFV) angezeigt. Geben Sie dann die Ladebibliothek in die z/OS-Stapelverarbeitung oder die TSO-STEPLIB ein, vor thlqual.SCSQAUTH.

**Zugehörige Konzepte**

„Konfigurieren Sie die Operationen und Steuerkonsolen.” auf Seite 1023
Um die Operationen und Steuerkonsolen zu konfigurieren, müssen Sie zuerst die Bibliotheken einrichten, die die erforderlichen Anzeigen, EXECs, Nachrichten und Tabellen enthalten. Dazu müssen Sie berücksichtigen, welche Landessprachenfunktion für die Anzeigen verwendet werden soll. Anschließend können Sie bei Bedarf das ISPF-Hauptmenü für Betriebs- und Steuerkonsolen in IBM MQ aktualisieren und die Belegung der Funktionstasten ändern.

## z/OS *Konfigurieren Sie die Operationen und Steuerkonsolen.*

Um die Operationen und Steuerkonsolen zu konfigurieren, müssen Sie zuerst die Bibliotheken einrichten, die die erforderlichen Anzeigen, EXECs, Nachrichten und Tabellen enthalten. Dazu müssen Sie berücksichtigen, welche Landessprachenfunktion für die Anzeigen verwendet werden soll. Anschließend können Sie bei Bedarf das ISPF-Hauptmenü für Betriebs- und Steuerkonsolen in IBM MQ aktualisieren und die Belegung der Funktionstasten ändern.

- *Diese Task muss einmal für jedes z/OS-System durchgeführt werden, auf dem IBM MQ ausgeführt werden soll.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*

## Bibliotheken konfigurieren

Führen Sie die folgenden Schritte aus, um die Betriebs- und Steuerkonsolen für IBM MQ einzurichten:

1. Stellen Sie sicher, dass alle in Ihren Verkettungen enthaltenen Bibliotheken im selben Format (F, FB, V, VB) vorhanden sind und die gleiche Blockgröße haben oder in der Reihenfolge abnehmender Blockgrößen liegen. Andernfalls haben Sie möglicherweise Probleme bei der Verwendung dieser Anzeigen.
2. Fügen Sie die Bibliothek thlqual.SCSQEXEC in Ihre SYSEXEC-oder SYSPROC-Verkettung ein, oder aktivieren Sie sie mit dem Befehl TSO ALTLIB. Diese Bibliothek, die bei der Installation mit einem Satzformat mit fester Blockierung 80 zugeordnet wird, enthält die erforderlichen EXECs.

   Es ist vorzuziehen, die Bibliothek in die SYSEXEC-Verkettung zu stellen. Wenn Sie es jedoch in SYSPROC stellen möchten, muss das Kassettenarchiv eine Satzlänge von 80 Byte haben.
3. Fügen Sie 'thlqual.SCSQAUTH' und 'thlqual.SCSQANLx' zur TSO-Anmeldeprozedur STEPLIB hinzu, oder aktivieren Sie sie mit dem TSOLIB-Befehl TSO, wenn sie nicht in der Linkliste oder im LPA enthalten ist.
4. Sie können die IBM MQ-Anzeigenbibliotheken entweder permanent der ISPF-Bibliothekskonfiguration hinzufügen oder angeben, dass sie bei Verwendung der Anzeigen dynamisch konfiguriert werden. Für die erstgenannte Auswahl müssen folgende Schritte getan werden:
   a. Fügen Sie die Bibliothek, die die Operationen und Steuerkonsoldefinitionen enthält, in Ihre ISPPLIB-Verkettung ein. Der Name ist thlqual.SCSQPNLx, wobei x für den Sprachenbuchstabe für Ihre Landessprache steht.
   b. Fügen Sie die Bibliothek mit den erforderlichen Tabellen in Ihre ISPTLIB-Verkettung ein. Der Name ist thlqual.SCSQTBLx, wobei x für den Sprachenbuchstabe für Ihre Landessprache steht.
   c. Schließen Sie die Bibliothek ein, die die erforderlichen Nachrichten in der ISPMLIB-Verkettung enthält. Der Name ist thlqual.SCSQMSGx, wobei x für den Sprachenbuchstabe für Ihre Landessprache steht.
   d. Fügen Sie die Bibliothek mit den erforderlichen Lademodulen in Ihre ISPLLIB-Verkettung ein. Der Name dieser Bibliothek lautet thlqual.SCSQAUTH.

   Verwenden Sie für die letztere Option den z/OS -Befehl LIBDEF . Einen Link zu verschiedenen Schlüsselwörtern, die Sie verwenden können, finden Sie unter Beispiele .
5. Prüfen Sie, ob Sie über die Anzeige des TSO-Befehlsprozessors auf die IBM MQ-Anzeigen zugreifen können. Dies ist in der Regel die Option 6 im Menü ISPF/PDF Primary Options. Der Name der EXEC, die Sie ausführen, ist CSQOREXX.Es sind keine Parameter anzugeben, wenn Sie die IBM MQ-Bibliotheken wie in Schritt 4 beschrieben permanent in Ihrer ISPF-Konfiguration eingerichtet haben. Wenn dies nicht der Fall ist, verwenden Sie die folgenden Schritte:

   ```
   CSQOREXX thlqual langletter
   ```

   Dabei steht `langletter` für einen Buchstaben, der die zu verwendende Landessprache angibt:

   **C**
   Vereinfachtes Chinesisch

   **E**
   U.S. Englisch (Groß-/Kleinschreibung)

   **F**
   Französisch

**K**
Japanisch

**U**
U.S. Englisch (Großschreibung)

## ISPF-Menü aktualisieren

Sie können das ISPF-Hauptmenü aktualisieren, damit ein Zugriff auf die Betriebs- und Steuerkonsolen von IBM MQ über ISPF möglich ist. Die erforderliche Einstellung für & ZSEL lautet:

```
CMD(%CSQOREXX thlqual langletter)
```

Informationen zu `thlqual` und `langletter`finden Sie in Schritt „5” auf Seite 1024.

Weitere Details finden Sie in *z/OS: ISPF Dialog Developer's Guide and Reference*.

## Aktualisieren der Funktionstasten und Befehlseinstellungen

Sie können die normalen ISPF-Prozeduren verwenden, um die Funktionstasten und die Befehlseinstellungen zu ändern, die von den Anzeigen verwendet werden. Die Anwendungs-ID ist CSQO.

Dies wird jedoch nicht empfohlen, da die Hilfinformationen nicht aktualisiert werden, um Änderungen zu widerzuspiegeln, die Sie vorgenommen haben.

**Zugehörige Konzepte**
„Teildatei für die Speicherauszugsformatierung in IBM MQ einschließen” auf Seite 1025
Damit IBM MQ-Speicherauszüge mit IPCS (Interactive Problem Control System) formatiert werden können, müssen Sie einige Systembibliotheken aktualisieren.

## z/OS *Teildatei für die Speicherauszugsformatierung in IBM MQ einschließen*

Damit IBM MQ-Speicherauszüge mit IPCS (Interactive Problem Control System) formatiert werden können, müssen Sie einige Systembibliotheken aktualisieren.

- *Diese Task muss einmal für jedes z/OS-System durchgeführt werden, auf dem IBM MQ ausgeführt werden soll.*
- *Sie müssen diese Task ausführen, wenn Sie eine Migration von einer früheren Version durchführen.*

Damit IBM MQ-Speicherauszüge mithilfe von IPCS (Interactive Problem Control System) formatiert werden können, kopieren Sie die Datei thlqual.SCSQPROC(CSQ7IPCS) in die Datei SYS1.PARMLIB. Sie sollten diese Dateigruppe nicht bearbeiten müssen.

Wenn Sie die TSO-Prozedur für IPCS angepasst haben, kann thlqual.SCSQPROC (CSQ7IPCS) in eine beliebige Bibliothek in der IPCSPARM-Definition kopiert werden. Weitere Informationen finden Sie im *z/OS MVS IPCS Benutzerhandbuch* .

Sie müssen auch die Bibliothek thlqual.SCSQPNLA in Ihre ISPPLIB-Verkettung einschließen.

Um die Formatierungsprogramme für Speicherauszüge für Ihre TSO-Sitzung oder den IPCS-Job verfügbar zu machen, müssen Sie auch die Bibliothek thlqual.SCSQAUTH in Ihre STEPLIB-Verkettung einschließen oder sie mit dem TSO-Befehl TSOLIB aktivieren (auch wenn sie bereits in der Linkliste oder LPA enthalten ist).

**Zugehörige Konzepte**
„Informationsnachrichten unterdrücken” auf Seite 1026
Ihr IBM MQ-System erzeugt möglicherweise eine große Anzahl an Informationsnachrichten. Sie können verhindern, dass ausgewählte Nachrichten an die Konsole oder an das Hardcopy-Protokoll gesendet werden.

## z/OS *Informationsnachrichten unterdrücken*

Ihr IBM MQ-System erzeugt möglicherweise eine große Anzahl an Informationsnachrichten. Sie können verhindern, dass ausgewählte Nachrichten an die Konsole oder an das Hardcopy-Protokoll gesendet werden.

- *Diese Task muss einmal für jedes z/OS-System durchgeführt werden, auf dem IBM MQ ausgeführt werden soll.*
- *Bei der Migration von einer früheren Version muss diese Task nicht ausgeführt werden.*

Wenn Ihr IBM MQ-System stark ausgelastet ist und viele Kanäle gestoppt und gestartet werden, wird eine große Anzahl an Informationsnachrichten an die z/OS-Konsole und an das Hardcopy-Protokoll gesendet. Die IBM MQ - IMS-Bridge und der Puffermanager können auch eine große Anzahl an Informationsnachrichten erzeugen.

Falls erforderlich, können Sie einige dieser Konsolnachrichten unterdrücken, indem Sie die von den MPFLSTxx-Membern von SYS1.PARMLIB angegebene Liste der Nachrichtenverarbeitungs-Facility von z/OS verwenden. Die von Ihnen angegebenen Nachrichten werden weiterhin im Hardcopyprotokoll angezeigt, aber nicht in der Konsole.

Das Beispiel `thlqual.SCSQPROC(CSQ4MPFL)` zeigt empfohlene Einstellungen für MPFLSTxx. Weitere Informationen finden Sie unter MPFLSTxx (Message Processing Facility List) .

Wenn Sie ausgewählte Informationsnachrichten im Hardcopy-Protokoll unterdrücken möchten, können Sie den z/OS-Installationsexit IEAVMXIT verwenden. Sie können die folgenden Bit-Schalter für die erforderlichen Nachrichten festlegen:

**CTXTRDTM**
Löschen Sie die Nachricht.

Die Nachricht wird nicht auf Konsolen angezeigt oder in Hardcopy protokolliert.

**CTXTESJL**
Unterdrückt aus Jobprotokoll.

Die Nachricht wird nicht in das JES-Jobprotokoll umgesetzt.

**CTXTNWTP**
Führen Sie die WTP-Verarbeitung nicht durch.

Die Nachricht wird nicht an ein TSO-Terminal oder an die Systemnachrichtendatei eines Stapeljobs gesendet.

**Anmerkung:**

1. Ausführliche Informationen zu den anderen Parameter finden Sie unter MVS Installation Exits.
2. Es wird nicht empfohlen, andere Nachrichten als die in der vorgeschlagenen Unterdrückungsliste, CSQ4MPFL, zu unterdrücken.

Darüber hinaus können Sie den zusätzlichen Parameter angeben:

**EXCLMSG**
Gibt eine Liste der Nachrichten an, die aus jedem Protokoll ausgeschlossen werden sollen.

Nachrichten in dieser Liste werden nicht an die z/OS-Konsole und den Protokollausdruck gesendet. Weitere Informationen finden Sie unter EXCLMSG in „CSQ6SYSP verwenden” auf Seite 998.

**Zugehörige Tasks**

„Warteschlangenmanager auf z/OS testen” auf Seite 1043
Wenn Sie Ihren Warteschlangenmanager angepasst oder migriert haben, können Sie ihn testen, indem Sie die Installationsprüfprogramme und einige der Beispielanwendungen ausführen, die mit IBM MQ for z/OS geliefert wurden.

## z/OS Gruppe mit gemeinsamer Warteschlange konfigurieren

Wenn Sie gemeinsam genutzte Warteschlangen für hohe Verfügbarkeit verwenden möchten, verwenden Sie diese Themen als Schritt für Schritt im Handbuch für die Konfiguration der Gruppe mit gemeinsamer Warteschlange.

Wenn Sie die Schritte in diesem Teil des Prozesses für die Konfiguration Ihres IBM MQ for z/OS-Systems ausgeführt haben, sollten Sie „Passen Sie Ihr Systemparametermodul an" auf Seite 996 zum Hinzufügen von Daten zur Gruppe mit gemeinsamer Warteschlange lesen. Sie müssen den Parameter CSQ6SYSP ändern, um den Parameter QSGDATA anzugeben.

### z/OS *Konfiguration der Db2-Umgebung*

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie die erforderlichen Db2-Objekte erstellen, indem Sie eine Reihe von Beispieljobs anpassen und ausführen.

### Konfiguration der Db2-Umgebung

Sie müssen die erforderlichen Db2-Objekte erstellen und binden, indem Sie eine Reihe von Beispieljobs anpassen und ausführen.

- Wiederholen Sie diese Task für jede Db2-Datennutzung-Gruppe.
- Sie müssen die Schritte `bind` und `grant` ausführen, wenn Sie eine Migration von einer früheren Version durchführen.
- Übergehen Sie diese Task, wenn Sie keine Gruppen mit gemeinsamer Warteschlange verwenden.

  Wenn Sie Gruppen mit gemeinsamer Warteschlange später verwenden möchten, führen Sie diese Task zu diesem späteren Zeitpunkt aus.

IBM MQ stellt zwei funktional entsprechende Gruppen von Jobs bereit. Die mit dem Präfix CSQ45 stehen für die Kompatibilität mit früheren Versionen von IBM MQ und für die Verwendung mit IBM MQ Version 11 und früheren Versionen. Wenn Sie eine neue Gruppe mit gemeinsamer Datennutzung mit Db2 V12 oder höher einrichten, sollten Sie die Jobs mit dem Präfix CSQ4X verwenden, da diese Jobs neuere Db2 -Funktionen für die dynamische Dimensionierung und Universal Table Spaces (UTS) nutzen.

Die folgenden Schritte müssen für jede neue Db2 Gruppe mit gemeinsamer Datennutzung ausgeführt werden. Die gesamte Muster-JCL befindet sich in thlqual.SCSQPROC.

1. Passen Sie die Beispiel-JCL CSQ4XCSG an und führen Sie sie aus, um die Speichergruppe zu erstellen, die für die IBM MQ-Datenbank, Tabellenbereiche und Tabellen verwendet werden soll.
2. Passen Sie die Beispiel-JCL CSQ4XCDB an und führen Sie sie aus, um die Datenbank zu erstellen, die von allen Warteschlangenmanagern verwendet werden soll, die eine Verbindung zu dieser Db2-Gruppe mit gemeinsamer Datennutzung herstellen.
3. Passen Sie die Beispiel-JCL CSQ4XCTS an und führen Sie sie aus, um die Tabellenbereiche zu erstellen, die die Tabellen für den Warteschlangenmanager und den Kanalinitiator für die Verwendung in Gruppen mit gemeinsamer Warteschlange enthalten.
4. Passen Sie die Beispiel-JCL CSQ4XCTB an, und führen Sie sie aus, um die 15 Db2-Tabellen und die zugehörigen Indizes zu erstellen. Ändern Sie keine der Zeilennamen oder Attribute.
5. Passen Sie die Beispiel-JCL CSQ45BPL an, und führen Sie sie aus, um die Db2-Pläne für den Warteschlangenmanager, die Dienstprogramme und den Kanalinitiator zu binden.
6. Passen Sie die Beispiel-JCL CSQ45GEX an und führen Sie sie aus, um die Ausführungsberechtigung für die Pläne für die Benutzer-IDs zu erteilen, die vom Warteschlangenmanager, den Dienstprogrammen und dem Kanalinitiator verwendet werden. Die Benutzer-IDs für den WS-Manager und den Kanalinitiator sind die Benutzer-IDs, unter denen die gestarteten Taskprozeduren ausgeführt werden. Die Benutzer-IDs für die Dienstprogramme sind die Benutzer-IDs, unter denen die Stapeljobs übergeben werden können.

   Die Namen der entsprechenden Pläne werden in der folgenden Tabelle angezeigt.

| User | Pläne (LTS) | Pläne (CD) |
|---|---|---|
| Warteschlangenmanager | CSQ5A 930<br>CSQ5C 930<br>CSQ5D 930<br>CSQ5K 930,<br>CSQ5L 930<br>CSQ5M 930<br>CSQ5P 930,<br>CSQ5R 930<br>CSQ5S 930<br>CSQ5T 930<br>CSQ5U 930<br>CSQ5W 930 | CSQ5A 9X0,<br>CSQ5C 9X0,<br>CSQ5D 9X0,<br>CSQ5K 9X0,<br>CSQ5L 9X0,<br>CSQ5M 9X0,<br>CSQ5P 9X0,<br>CSQ5R 9X0,<br>CSQ5S 9X0,<br>CSQ5T 9X0,<br>CSQ5U 9X0,<br>CSQ5W 9X0 |
| SDEFS-Funktion des Stapeldienst-programms CSQUTIL | CSQ52 930 | CSQ52 9X0 |
| CSQ5PQSG-und CSQJUCNV-Stapel-dienstprogramme | CSQ5B 930 | CSQ5B 9X0 |
| CSQUZAP, Servicedienstprogramm | CSQ5Z 930 | CSQ5Z 9X0 |

Wenn während der Db2-Konfiguration ein Fehler aufgetreten ist, können die folgenden Jobs angepasst und ausgeführt werden:

- CSQ45DTB, um die Tabellen und Indizes zu löschen.
- CSQ4XDTS, um die Tabellenbereiche zu löschen.
- CSQ4XDDB, um die Datenbank zu löschen.
- CSQ4XDSG, um die Speichergruppe zu löschen.

**Anmerkung:** Wenn diese Jobs aufgrund eines Db2-Sperrungsproblems fehlschlagen, ist dies wahrscheinlich auf eine Konkurrenzsituation für eine Db2-Ressource zurückzuführen, insbesondere dann, wenn das System stark genutzt wird. Übergeben Sie die Jobs später erneut. Es ist vorzuziehen, dass diese Jobs ausgeführt werden, wenn das System leicht verwendet oder in den Quiescemodus versetzt wird.

Weitere Informationen zum Einrichten von Db2finden Sie in Db2 Administration in *Db2 für z/OS 12.0.0* .

Informationen zu Db2 -Tabellengrößen finden Sie in Planung für z/OS .

**Zugehörige Konzepte**

„Einrichten der Coupling-Facility” auf Seite 1028
Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, definieren Sie mithilfe von IXCMIAPU die Coupling-Facility-Strukturen, die von den Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange (QSG) im Datensatz für die CFRM-Richtlinie (Coupling Facility Resource Management) verwendet werden.

## z/OS *Einrichten der Coupling-Facility*

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, definieren Sie mithilfe von IXCMIAPU die Coupling-Facility-Strukturen, die von den Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange (QSG) im Datensatz für die CFRM-Richtlinie (Coupling Facility Resource Management) verwendet werden.

Weitere Informationen zu IXCMIAPU finden Sie unter Dienstprogramm für Verwaltungsdaten .

- Wiederholen Sie diese Task für jede Gruppe mit gemeinsamer Warteschlange.
- Möglicherweise müssen Sie diese Task ausführen, wenn Sie eine Migration von einer früheren Version durchführen.

- Übergehen Sie diese Task, wenn Sie keine Gruppen mit gemeinsamer Warteschlange verwenden.

  Wenn Sie Gruppen mit gemeinsamer Warteschlange später verwenden möchten, führen Sie diese Task zu diesem späteren Zeitpunkt aus.

Alle Strukturen für die Gruppe mit gemeinsamer Warteschlange beginnen mit dem Namen der Gruppe mit gemeinsamer Warteschlange. Definieren Sie die folgenden Strukturen:

- Eine Verwaltungsstruktur mit dem Namen *qsg-name* CSQ_ADMIN. Diese Struktur wird von IBM MQ selbst verwendet; sie enthält keine Benutzerdaten.
- Eine Systemanwendungsstruktur mit dem Namen *qsg-name* CSQSYSAPPL. Diese Struktur wird von IBM MQ-Systemwarteschlangen verwendet, um Statusinformationen zu speichern.
- Eine oder mehrere Strukturen, die zum Speichern von Nachrichten für gemeinsam genutzte Warteschlangen verwendet werden. Diese können einen beliebigen Namen haben, der bis zu 16 Zeichen lang ist.
  - Die ersten vier Zeichen müssen der Name der Gruppe mit gemeinsamer Warteschlange sein. (Wenn der Name der Gruppe mit gemeinsamer Warteschlange kürzer als vier Zeichen ist, muss er mit @-Symbolen auf vier Zeichen aufgefüllt werden werden.)
  - Das fünfte Zeichen muss ein alphabetisches Zeichen sein, und die nachfolgenden Zeichen können alphabetisch oder numerisch sein. Diesen Teil des Namens (ohne den Namen der Gruppe mit gemeinsamer Warteschlange) geben Sie beim Definieren einer gemeinsamen Warteschlange oder eines CF-Strukturobjekts für den Namen CFSTRUCT an.

  Sie können nur alphabetische und numerische Zeichen in den Namen der Strukturen verwenden, die zum Speichern von Nachrichten für gemeinsam genutzte Warteschlangen verwendet werden. Sie können keine anderen Zeichen (z. B. das Zeichen _ verwenden, das im Namen der Verwaltungsstruktur verwendet wird).

Die Beispielsteueranweisungen für IXCMIAPU befinden sich in der Datei thlqual.SCSQPROC (CSQ4CFRM). Passen Sie diese an und fügen Sie sie zu Ihrem IXCMIAPU-Job für die Coupling Facility hinzu und führen Sie sie aus.

Wenn Sie Ihre Strukturen erfolgreich definiert haben, aktivieren Sie die CFRM-Richtlinie, die verwendet wird. Geben Sie dazu den folgenden z/OS-Befehl aus:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

Informationen zur Planung von CF-Strukturen und deren Größen finden Sie unter Ressourcen für Coupling-Facility definieren.

**Zugehörige Konzepte**

„Implementieren Sie Ihre ESM-Sicherheitskontrollen.” auf Seite 987
Implementieren Sie die Sicherheitssteuerungen für Warteschlangenmanager und den Kanalinitiator.

## z/OS *Richten Sie die SMDS-Umgebung ein.*

Wenn Sie SMDS zum Auslagern von Nachrichten in gemeinsam genutzten Warteschlangen verwenden möchten, konfigurieren Sie die SMDS-Auslagerung-Speicherumgebung.

- *Führen Sie diese Task für jeden Warteschlangenmanager und jede Struktur in der Gruppe mit gemeinsamer Warteschlange aus, die Sie für die Auslagerung von Daten in SMDS konfigurieren möchten.*
- *Wenn Sie zusätzliche Strukturen konfigurieren möchten, um Daten später in SMDS auslagern zu können, kann diese Task zu diesem Zeitpunkt erneut ausgeführt werden.*
- *Übergehen Sie diese Task, wenn Sie keine Gruppen mit gemeinsamer Warteschlange verwenden.*

  *Wenn Sie Gruppen mit gemeinsamer Warteschlange später verwenden möchten, führen Sie diese Task zu diesem späteren Zeitpunkt aus*.

## Richten Sie die SMDS-Umgebung ein.

1. Schätzen Sie die Struktur-und Datenspeicherplatzanforderungen. Siehe Hinweise zur Kapazitätsüberlegungen für gemeinsam genutzte Nachrichtendaten .
2. Dateien zuordnen und vorformatieren. Siehe Gemeinsam genutzte Nachrichtendatei erstellen .
3. Wenn Sie die CF-Struktur für IBM MQ definieren, stellen Sie sicher, dass Sie CFSTRUCT mit CFLEVEL(5) und OFFLOAD(SMDS) definieren.

**Zugehörige Konzepte**

„Einrichten der Coupling-Facility” auf Seite 1028
Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, definieren Sie mithilfe von IXCMIAPU die Coupling-Facility-Strukturen, die von den Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange (QSG) im Datensatz für die CFRM-Richtlinie (Coupling Facility Resource Management) verwendet werden.

## z/OS *IBM MQ-Einträge zu den Db2-Tabellen hinzufügen*

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, führen Sie das Dienstprogramm CSQ5PQSG aus, um Einträge in Gruppen mit gemeinsamer Warteschlange und in Warteschlangenmanagern den IBM MQ-Tabellen in der Db2-Gruppe mit gemeinsamer Datennutzung hinzuzufügen.

- *Führen Sie diese Task für jede IBM MQ-Gruppe mit gemeinsamer Warteschlange und jeden Warteschlangenmanager aus.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*
- *Übergehen Sie diese Task, wenn Sie keine Gruppen mit gemeinsamer Warteschlange verwenden.*

  *Wenn Sie Gruppen mit gemeinsamer Warteschlange später verwenden möchten, führen Sie diese Task zu diesem späteren Zeitpunkt aus.*

Führen Sie CSQ5PQSG für jede Gruppe mit gemeinsamer Warteschlange und jeden Warteschlangenmanager aus, der Mitglied einer Gruppe mit gemeinsamer Warteschlange sein soll.

Führen Sie die folgenden Aktionen in der angegebenen Reihenfolge aus:

1. Fügen Sie mithilfe der ADD QSG-Funktion des Programms CSQ5PQSG einen Eintrag in der Gruppe mit gemeinsamer Warteschlange den IBM MQ Db2-Tabellen hinzu. Eine Probe wird in thlqual.SCSQPROC (CSQ45AQS) bereitgestellt.

   Führen Sie diese Funktion einmal für jede Gruppe mit gemeinsamer Warteschlange aus, die in der Db2-Gruppe mit gemeinsamer Datennutzung definiert ist. Der Eintrag in der Gruppe mit gemeinsamer Warteschlange muss vorhanden sein, bevor Sie Warteschlangenmanagereinträge hinzufügen können, die auf die Gruppe mit gemeinsamer Warteschlange verweisen.
2. Fügen Sie mit der Funktion ADD QMGR des Programms CSQ5PQSG einen Warteschlangenmanagereintrag in die Tabellen von IBM MQ Db2 ein. Ein Beispiel wird in thlqual.SCSQPROC (CSQ45AQM) bereitgestellt.

   Führen Sie diese Funktion für jeden Warteschlangenmanager aus, der Mitglied der Gruppe mit gemeinsamer Warteschlange sein soll.

   **Anmerkung:**

   a. Ein Warteschlangenmanager kann nur Mitglied einer Gruppe mit gemeinsamer Warteschlange sein.

   b. RRS muss aktiv sein, damit Gruppen mit gemeinsamer Warteschlange verwendet werden können.

**Zugehörige Konzepte**

„Passen Sie Ihr Systemparametermodul an” auf Seite 996
Über das Systemparametermodul von IBM MQ werden die von IBM MQ während des Betriebs verwendeten Umgebungen für Protokollierung, Archivierung, Tracing und Verbindungen gesteuert. Es wird ein Standardmodul bereitgestellt. Sie sollten ein eigenes Systemparametermodul erstellen, da einige Parameter, z. B. Datensatznamen, in der Regel standortspezifisch sind.

### z/OS *Implementieren Sie die ESM-Sicherheitssteuerelemente für die Gruppe mit gemeinsamer Warteschlange.*

Implementieren Sie Sicherheitsmaßnahmen für alle Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange für den Zugriff auf Db2 und die Listenstrukturen der Coupling-Facility.

- *Wiederholen Sie diese Task für jeden IBM MQ-Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange.*
- *Diese Task muss unter Umständen bei der Migration von einer früheren Version ausgeführt werden.*

Stellen Sie sicher, dass die Benutzer-IDs, die dem Warteschlangenmanager, dem Kanalinitiator und den Dienstprogrammen zugeordnet sind, berechtigt sind, eine RRSAF-Verbindung zu jedem Db2-Subsystem herzustellen, mit dem eine Verbindung hergestellt werden soll. Die Benutzer-IDs für den WS-Manager und den Kanalinitiator sind die Benutzer-IDs, unter denen die gestarteten Taskprozeduren ausgeführt werden.

Die Benutzer-IDs für die Dienstprogramme sind die Benutzer-IDs, unter denen die Stapeljobs übergeben werden können. Das RACF-Profil, für das die Benutzer-ID READ-Zugriff erfordert, ist `Db2ssid.RRSAF` in der DSNR-Ressourcenklasse.

Die Benutzer-IDs, die den einzelnen Warteschlangenmanagern in einer Gruppe mit gemeinsamer Warteschlange zugeordnet sind, müssen die entsprechende Zugriffsebene für die Coupling Facility-Listenstrukturen erhalten. Die RACF-Klasse ist FACILITY.

Für die folgenden Benutzer-IDs ist ein ALTER-Zugriff erforderlich:

- Die WS-Manager-ID für das `IXLSTR.structure-name` -Profil.
- Die Benutzer-ID, die CSQ5PQSG ausführt.

**Zugehörige Konzepte**

„Implementieren Sie Ihre ESM-Sicherheitskontrollen.” auf Seite 987
Implementieren Sie die Sicherheitssteuerungen für Warteschlangenmanager und den Kanalinitiator.

## z/OS Advanced Message Security for z/OS konfigurieren

Verwenden Sie diese Themen als Schritt-für-Schritt-Anweisung für die Konfiguration von Advanced Message Security (AMS).

### Vorbereitende Schritte

Bevor Sie mit der Konfiguration von AMS beginnen, müssen Sie sicherstellen, dass die folgenden Konfigurationsschritte für Warteschlangenmanager ausgeführt wurden:

1. Fügen Sie dass Modul CSQ0DRTM zum LPA hinzu, wie in „Aktualisieren der z/OS-Linkliste und LPA” auf Seite 975 beschrieben.
2. Fügen Sie einen Eintrag für CSQ0DSRV zur z/OS-Programmeigenschaftentabelle (PPT) hinzu, wie in „Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften” auf Seite 979 beschrieben.
3. Schließen Sie das Member CSQ4INSM in die CSQINP2-Verkettung der gestarteten Taskprozedur des Warteschlangenmanagers ein, wie in „Passen Sie die Initialisierungseingabedatensätze an.” auf Seite 988 beschrieben.
4. Aktivieren Sie AMS mithilfe des Attributs AMSPROD. Weitere Einzelheiten finden Sie im Abschnitt Aufzeichnung der Produktnutzung bei IBM MQ for z/OS-Produkten.

### Nächste Schritte

Konfigurieren Sie Richtlinien für Warteschlangen, die durch AMS geschützt sind. Sicherheitsrichtlinien werden im Abschnitt Advanced Message Security-Sicherheitsrichtlinien verwalten beschrieben.

Beispiele für AMS-Konfigurationen finden Sie im Abschnitt Beispielkonfigurationen unter z/OS.

## z/OS *Prozeduren für Advanced Message Security erstellen*

Jedes IBM MQ-Subsystem, das für die Verwendung von Advanced Message Security (AMS) konfiguriert werden soll, erfordert eine katalogisierte Prozedur zum Starten des AMS-Adressraums. Sie können eigene Prozeduren erstellen oder die von IBM bereitgestellte Prozedurbibliothek verwenden.

### Vorgehensweise

1. Kopieren Sie die gestartete Taskprozedur *thlqual* .SCSQPROC (CSQ4AMSM) in die Datei SYS1.PROCLIB oder, wenn Sie nicht SYS1.PROCLIB verwenden, die Prozedurbibliothek. Benennen Sie die Prozedur xxxxAMSM, wobei xxxx der Name Ihres IBM MQ-Subsystems ist. Beispiel: CSQ1AMSM wäre die AMS-Prozedur für gestartete Tasks für den Warteschlangenmanager CSQ1.
2. Erstellen Sie eine Kopie für jedes IBM MQ-Subsystem, das verwendet werden soll.
3. Passen Sie die Prozeduren gemäß den Anweisungen in der Beispielprozedur CSQ4AMSM an Ihre Anforderungen an. Sie können auch symbolische Parameter in der JCL verwenden, um die Prozedur zu ändern, wenn sie gestartet wird.
4. Überprüfen und ändern Sie ggf. die an die AMS-Task übergebenen Parameter mit Hilfe der Datei Language Environment® _CEE_ENVFILE. In dem Beispiel thlqual.SCSQPROC (CSQ40ENV) werden die unterstützten Parameter aufgelistet.
5. Wiederholen Sie die Schritte 1 bis 4 für jeden IBM MQ-Warteschlangenmanager.

### Nächste Schritte

„Einrichten der Advanced Message Security-Benutzer-ID für gestartete Tasks” auf Seite 1032

## z/OS *Einrichten der Advanced Message Security-Benutzer-ID für gestartete Tasks*

Die Task Advanced Message Security (AMS) erfordert eine Benutzer-ID, die es zulässt, dass sie als z/OS UNIX System Services (z/OS UNIX)-Prozess bezeichnet wird.

### Informationen zu diesem Vorgang

Darüber hinaus müssen die Benutzer, in deren Namen die Task ausgeführt wird, auch über eine entsprechende Definition einer UNIX-UID (Benutzer-ID) und einer GID (Gruppen-ID) verfügen, damit diese Benutzer als Benutzer von z/OS UNIX System Services bekannt sind. Weitere Informationen zum Definieren von z/OS UNIX System Services UIDs und GIDs finden Sie in *z/OS: Security Server RACF Security Administrator's Guide*.

Lesen Sie *z/OS UNIX System Services Planung* , um sicherzustellen, dass Sie die Sicherheitsunterschiede zwischen der traditionellen UNIX -Sicherheit und der z/OS UNIX -Sicherheit verstehen. Auf diese Weise können Sie die Advanced Message Security-Task entsprechend der Sicherheitsrichtlinie Ihrer Installation verwalten, um privilegierte z/OS UNIX System Services-Prozesse zu implementieren und auszuführen.

Der primäre Unterschied zwischen der traditionellen Sicherheit von UNIX und der z/OS-Sicherheit besteht darin, dass die Kernel-Services zwei Ebenen der entsprechenden Berechtigungen unterstützen: die UNIX-Ebene und die z/OS UNIX-Ebene.

Je nach Sicherheitsrichtlinie Ihrer Installation kann die Task Advanced Message Security entweder mit Superuserberechtigung (uid (0)) oder mit ihrer RACF -Identität ausgeführt werden, die für die BPX RACF FACILITY-Klasse BPX.DAEMON und BPX.SERVER -Profile, da diese Task in der Lage sein muss, die RACF -Identität ihrer Benutzer anzunehmen.

Wenn die letztgenannte Methode verwendet wird oder Sie BPX.DAEMON oder BPX.SERVER -Profile muss sich das Advanced Message Security -Taskprogramm (thlqual.SCSQAUTH(CSQ0DSRV)) in programmgesteuerten RACF -Bibliotheken befinden.

**Anmerkung:** Wählen Sie die Benutzer-ID für diese Task sorgfältig aus, da die Advanced Message Security-Empfängerzertifikate in einen Schlüsselring geladen werden, der dieser Benutzer-ID zugeordnet ist. Dieser Aspekt wird im Abschnitt Zertifikate unter z/OS verwendenerläutert.

Die hier genannten Schritte beschreiben die Einrichtung des Benutzers für die gestartete Task von Advanced Message Security. In den Schritten werden RACF-Befehle als Beispiele verwendet. Wenn Sie einen anderen Sicherheitsmanager verwenden, sollten Sie funktional entsprechende Befehle verwenden.

**Anmerkung:** In den Beispielen in diesem Abschnitt wird vorausgesetzt, dass Sie die generische Profilbefehlsverarbeitung für die RACF-Klassen STARTED, FACILITY und SURROGAT und die generische Profilprüfung aktiviert haben. Weitere Informationen dazu, wie RACF generische Profile behandelt, finden Sie unter *z/OS: Security Server RACF Command Language Reference*.

## Vorgehensweise

1. Definieren Sie den gestarteten Task-Benutzer von Advanced Message Security für RACF. Die Beispiele in diesem Abschnitt verwenden die Benutzer-ID WMQAMSM.

   ```
   ADDUSER WMQAMSM NAME('AMS user') OMVS (UID(0)) DFLTGRP(group)
   ```

   Wählen Sie eine Standardgruppe aus, die Ihren Installationsstandards entsprechend ist.

   **Anmerkung:** Wenn Sie die z/OS UNIX -Superuserberechtigung (UID (0)) nicht erteilen möchten, müssen Sie die Advanced Message Security -Benutzer-ID für BPX.DAEMON und BPX.SERVER Facility-Klassenprofile:

   ```
   PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
   ```

   Außerdem muss sich das Advanced Message Security-Taskprogramm (*thlqual*.SCSQAUTH(CSQ0DSRV)) in der programmgesteuerten RACF-Bibliothek befinden.

   Um das Bibliotheksprogramm SCSQAUTH zu steuern, können Sie den folgenden Befehl verwenden:

   ```
   RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-
   RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)
   SETROPTS WHEN(PROGRAM) REFRESH
   ```

   Sie müssen auch die Programmsteuerung für die Landessprachenbibliothek (*thlqual*.SCSQANLx) aktivieren, die von der Task Advanced Message Security verwendet wird.
2. Stellen Sie fest, ob die RACF-Klasse STARTED aktiv ist. Ist dies nicht der Fall, aktivieren Sie die RACF-Klasse STARTED:

   ```
   SETROPTS CLASSACT(STARTED)
   ```
3. Definieren Sie ein gestartetes Klassenprofil für die Advanced Message Security-Tasks und geben Sie dabei die in Schritt 1 erstellte oder erstellte Benutzer-ID an:

   ```
   RDEFINE STARTED qmgrAMSM.* STDATA(USER(WMQAMSM))
   ```

   Dabei steht *qmgr* für das Präfix des Namens der gestarteten Task. Die gestartete Task kann z. B. CSQ1AMSM genannt werden. In diesem Fall würden Sie *qmgr*AMSM.* durch CSQ1AMSM.* ersetzen.

   Die gestarteten AMS -Tasks müssen den Namen *qmgr*AMSMhaben.
4. Verwenden Sie den Befehl **SETROPTS** RACF , um die im Speicher RACLISTed STARTED-Klassenprofile zu aktualisieren:

   ```
   SETROPTS RACLIST(STARTED) REFRESH
   ```
5. Die Advanced Message Security-Task übernimmt während der Zugriffsschutzverarbeitung von IBM MQ-Nachrichten vorübergehend die Identität der Hostbenutzer-ID des Anforderers. Daher ist es erforderlich, Profile in der Klasse SURROGAT für jede Benutzer-ID, die Anforderungen stellen kann, zu definieren.

Wenn die RACF-Klasse SURROGAT aktiv ist, ermöglicht die Definition eines einzelnen generischen Profils der Advanced Message Security-Task, die Identität eines beliebigen Benutzers zu übernehmen. Die Prüfung wird ignoriert, wenn die SURROGAT-Klasse nicht aktiv ist. Die erforderlichen SURROGAT-Profile werden in *z/OS UNIX System Services Planung*beschrieben.

Gehen Sie wie folgt vor, um Profile in der Klasse SURROGAT zu definieren:

a) Aktivieren Sie die RACF-Klasse SURROGAT mit dem RACF-Befehl SETROPTS:

```
SETROPTS CLASSACT(SURROGAT)
```

b) Aktivieren Sie die generische Profilverarbeitung für die RACF SURROGAT-Klasse:

```
SETROPTS GENERIC(SURROGAT)
```

c) Aktivieren Sie die Befehlsverarbeitung von generischen Profilen für die RACF-Klasse SURROGAT:

```
SETROPTS GENCMD(SURROGAT)
```

d) Definieren Sie ein generisches Profil in der Klasse SURROGAT:

```
RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
```

e) Lassen Sie die Advanced Message Security-Benutzer-ID für das generische SURROGAT-Klassenprofil zu:

```
PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQAMSM) ACCESS(READ)
```

**Anmerkung:** Sie können spezifischere Profile definieren, wenn Sie bestimmte Benutzer auf die Verarbeitung durch die Advanced Message Security -Task beschränken möchten, wie in *z/OS UNIX System Services Planung*beschrieben.

Ein Profil mit dem Namen BPX.SRV.MQUSER1 steuert beispielsweise, ob die AMS-Task die Identität der Benutzer-ID MQUSER1 annehmen kann.

f) Ermöglichen Sie die Advanced Message Security-Benutzer-ID der BPX.SERVER-Funktion (falls dies nicht bereits in Zertifikate und Schlüsselringe erstellen geschehen ist):

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

g) Verwenden Sie den Befehl **SETROPTS** RACF , um die gestarteten RACLISTed-Klassenprofile im Speicher zu aktualisieren:

```
SETROPTS RACLIST(SURROGAT) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
```

6. Die Advanced Message Security-Task verwendet die von den z/OS System SSL-Services bereitgestellten Facilitys zum Öffnen von SAF-verwalteten Schlüsselringen. Die zugrunde liegende System Authorization Facility (SAF), die auf den Inhalt der Schlüsselringe zugreift, wird von RACF oder einem funktional entsprechenden Sicherheitsmanager gesteuert.

   Bei diesem Service handelt es sich um den aufrufbaren IRRSDL00-Service (R_datalib). Dieser aufrufbare Service wird mit den gleichen Profilen geschützt, die zum Schutz der RACF-Befehle des Typs RACDCERT verwendet werden, welche für die RACF-Klasse FACILITY definiert sind. Daher muss die Advanced Message Security-Benutzer-ID mit folgenden Befehlen für die Profile zugelassen werden:

   a) Falls noch nicht erfolgt, definieren Sie ein generisches RACF-Profil für die RACF-Klasse FACILITY, die den RACDCERT-Befehl und den aufrufbaren IRRSDL00-Service schützt:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
```

b) Erteilen Sie der Benutzer-ID der gestarteten Task die Berechtigung für das generische RACF-Profil:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(WMQAMSM) ACC(READ)
```

Alternativ können Sie dem Schlüsselring des Datenservice-Taskbenutzers in der Klasse RDATALIB wie folgt Lesezugriff erteilen:

```
PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(WMQAMSM) ACC(READ)
```

7. Ressourcensicherheit konfigurieren:

a) Der Benutzer der gestarteten Advanced Message Security-Task benötigt die Berechtigung, eine Verbindung zum Warteschlangenmanager als Stapelanwendung herzustellen.

Wenn für Ihren Warteschlangenmanager die Verbindungssicherheit aktiviert ist, erteilen Sie der AMS-Task mit folgendem Befehl die Berechtigung, eine Verbindung zum Warteschlangenmanager herzustellen:

```
PERMIT hlq.BATCH CLASS(MQCONN) ID(WMQAMSM) ACC(READ)
```

Dabei kann *hlq* entweder der Name der Gruppe mit gemeinsamer Warteschlange mit gemeinsamer Warteschlange und Name der Gruppe mit gemeinsamer Warteschlange sein

Weitere Informationen finden Sie unter Verbindungssicherheitsprofile für Stapelverbindungen .

b) Der Benutzer der gestarteten Advanced Message Security -Task benötigt die Berechtigung zum Durchsuchen des SYSTEM.PROTECTION.POLICY.QUEUE.

Wenn die Warteschlangensicherheit auf dem Warteschlangenmanager aktiv ist, erteilen Sie dem AMS-Benutzer mit den folgenden Befehlen die Berechtigung für den Zugriff auf die Warteschlange:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE) ID(WMQAMSM) ACCESS(READ)
```

Dabei kann *hlq* entweder der Name der Gruppe mit gemeinsamer Warteschlange mit gemeinsamer Warteschlange und Name der Gruppe mit gemeinsamer Warteschlange sein

Wenn der WS-Manager Profile mit Groß-/Kleinschreibung verwendet, definieren Sie stattdessen das Profil in der Klasse MXQUEUE.

Zum Verwalten von AMS -Sicherheitsrichtlinien mit dem Dienstprogramm CSQ0UTIL benötigen Administratoren Zugriff zum Einreihen von Nachrichten in das SYSTEM.PROTECTION.POLICY.QUEUE. Dies wird durch die Erteilung von UPDATE-Zugriff auf das Profil, das die Warteschlange schützt, ausgeführt.

Weitere Informationen finden Sie unter Profile für die Warteschlangensicherheit.

## Nächste Schritte

„Dem Sicherheitsadministrator RACDCERT-Berechtigungen für Advanced Message Security erteilen” auf Seite 1035

## z/OS *Dem Sicherheitsadministrator RACDCERT-Berechtigungen für Advanced Message Security erteilen*

Ihr Advanced Message Security-Sicherheitsadministrator benötigt die Berechtigung zur Verwendung des Befehls RACDCERT, um digitale Zertifikate zu erstellen und zu verwalten.

## Prozedur

- Geben Sie die entsprechende Benutzer-ID für diese Rolle an und erteilen Sie die Berechtigung, den Befehl RACDCERT zu verwenden. For example:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

  Hierbei steht admin für die Benutzer-ID Ihres Advanced Message Security-Sicherheitsadministrators.

## Nächste Schritte

„Benutzern Ressourcenberechtigungen für Advanced Message Security erteilen” auf Seite 1036

# z/OS *Benutzern Ressourcenberechtigungen für Advanced Message Security erteilen*

Advanced Message Security-Benutzer benötigen relevante Ressourcenberechtigungen.

## Informationen zu diesem Vorgang

Advanced Message Security-Benutzer, d. h., Benutzer, die Advanced Message Security-geschützte Nachrichten einreihen oder erhalten, benötigen Folgendes:

- Ein OMVS-Segment, das ihrer Benutzer-ID zugeordnet ist.
- Berechtigungen für IRR.DIGTCERT.LISTRING oder RDATALIB
- Berechtigungen für ICSF-Klassen CSFSERV-und CSFKEYS-Profile
- Berechtigung zum Versetzen in SYSTEM.PROTECTION.ERROR.QUEUE

Die Task Advanced Message Security nimmt vorübergehend die Identität ihrer Clients an, d. h. die Task fungiert als Ersatz für die z/OS-Benutzer-ID von Benutzern von Advanced Message Security während der Verarbeitung von IBM MQ-Nachrichten in Warteschlangen, die von Advanced Message Security geschützt werden.

Damit die Task die z/OS-Identität eines Benutzers übernehmen kann, muss der Client-z/OS-Benutzer-ID ein definiertes OMVS-Segment zugeordnet sein, das dem Benutzerprofil zugeordnet ist.

Als Verwaltungshilfe bietet RACF die Möglichkeit, ein Standard-OMVS-Segment zu definieren, das mit RACF-Benutzer- und -Gruppenprofilen verknüpft sein kann. Dieser Standardwert wird verwendet, wenn für die z/OS-Benutzer-ID oder das Gruppenprofil kein OMVS-Segment explizit definiert ist. Wenn in Ihrem Fall viele Benutzer Advanced Message Security nutzen sollen, kann es sinnvoll sein, diesen Standardwert zu wählen und das OMVS-Segment nicht für jeden einzelnen Benutzer explizit zu definieren.

Das *z/OS: Security Server RACF Security Administrator's Guide* enthält die detaillierte Prozedur zum Definieren von OMVS-Standardsegmenten. Überprüfen Sie die in dieser Veröffentlichung beschriebene Prozedur, um festzustellen, ob die Definition der Standard-OMVS-Segmente in den RACF-Profilen für Benutzer und Gruppen für Ihre Installation geeignet ist.

## Vorgehensweise

1. Erteilen Sie die Berechtigung READ für das Profil IRR.DIGTCERT.LISTRING in der Klasse FACILITY:
   - Um allen Benutzern die Berechtigung READ für das Profil IRR.DIGTCERT.LISTRING in der Klasse FACILITY zu erteilen, geben Sie den folgenden Befehl aus:

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
```

   - Geben Sie den folgenden Befehl aus, um dem IRR.DIGTCERT.LISTRING-Profil in der Klasse FACILITY pro Benutzer die Berechtigung READ zu erteilen:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

Dabei ist 'Benutzer-ID' der Name des Benutzers von Advanced Message Security.

- Alternativ können Sie auch die Klasse RDATALIB verwenden, um Zugriff auf bestimmte Schlüsselringe zu erteilen. Die Berechtigungen von RDATALIB haben Vorrang vor Berechtigungen für IRR.DIGTCERT.LISTRING. For example:

```
PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(user) ACC(READ)
```

2. Wenn Sie ICSF-verwaltete Zertifikate und private Schlüssel verwenden, müssen die Benutzer von Advanced Message Security auf bestimmte Profile der CSFSERV- und CSFKEYS-Klassen zugreifen können. Dieser Zugriff wird in der folgenden Tabelle beschrieben:

*Tabelle 63. Erforderlicher Benutzerzugriff auf Profile der Klassen CSFSERV und CSFKEYS*

| Klasse | Profil | Berechtigung |
|---|---|---|
| CSFSERV | CSFDSG | READ |
| CSFSERV | CSFPKE | READ |
| CSFSERV | CSFPKD | READ |
| CSFSERV | CSFDSV | READ |
| CSFKEYS | ICSF-PKDS-Kennsatz | READ |

3. Anwendungen, die Operationen für Warteschlangen mit definierten AMS -Richtlinien ausführen, benötigen Zugriff zum Einreihen von Nachrichten in SYSTEM.PROTECTION.ERROR.QUEUE. Erteilen Sie der Warteschlange mit den folgenden Befehlen Zugriff auf die Warteschlange:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE) ID(userId) ACCESS(UPDATE)
```

Dabei kann *hlq* entweder der Name der Gruppe mit gemeinsamer Warteschlange für den WS-Manager und *userID* die Anwendungsbenutzer-ID sein.

## Nächste Schritte

„Schlüsselringe für Advanced Message Security erstellen” auf Seite 1037

# z/OS Schlüsselringe für Advanced Message Security erstellen

Zertifikate, die von Advanced Message Security (AMS) für die Signatur und Verschlüsselung verwendet werden, werden in SAF-Schlüsselringen von z/OS gespeichert. Sie müssen diese Schlüsselringe und Zertifikate erstellen, bevor Sie AMS verwenden können.

## Informationen zu diesem Vorgang

Advanced Message Security greift auf Zertifikate in den folgenden Schlüsselringen zu:

- Ein einzelner Schlüsselring, der Eigentum des AMS-Adressraumbenutzers ist.
- Schlüsselringe, die Eigentum der einzelnen Benutzer sind, die Nachrichten in Warteschlangen mit definierten AMS-Richtlinien senden oder empfangen.

Diese Schlüsselringe müssen alle mit dem Namen `drq.ams.keyring` benannt werden.

Weitere Informationen zu den von AMS verwendeten Schlüsselringen und Zertifikaten sowie ein Beispielszenario finden Sie im Abschnitt Zertifikate unter z/OS verwenden.

Führen Sie die folgenden Schritte aus, um die von AMS benötigten Schlüsselringe zu erstellen, und verknüpfen Sie Zertifikate mit den Schlüsselringen. Den Schlüsselring, dessen Eigner der AMS-Adressraum-

benutzer ist, müssen Sie erstellen, bevor Sie AMS starten. Die Schlüsselringe, deren Eigner die Benutzer sind, die Nachrichten senden oder empfangen, können Sie zu einem beliebigen Zeitpunkt erstellen.

## Vorgehensweise

1. Geben Sie den folgenden Befehl aus, um einen Schlüsselring zu erstellen, dessen Eigner der AMS-Adressraumbenutzer ist:

   ```
   RACDCERT ID(amsUser) ADDRING(drq.ams.keyring)
   ```

   Dabei steht *amsUser* für die Benutzer-ID des AMS-Adressraums.
2. Erstellen Sie einen Schlüsselring für jeden Benutzer, der Nachrichten sendet oder empfängt, die von AMS geschützt werden, indem Sie den Befehl in Schritt 1 für jede Benutzer-ID ausgeben.
3. Verbinden Sie das Zertifikat der Zertifizierungsstelle (CA) für den Aussteller der Benutzerzertifikate mit dem Schlüsselring, dessen Eigner der AMS-Adressraumbenutzer-ID ist. Geben Sie den folgenden Befehl ein:

   ```
   RACDCERT ID(amsUser) CONNECT(CERTAUTH LABEL('caLabel') RING(drq.ams.keyring))
   ```

   Dabei steht *amsUser* für die Benutzer-ID des AMS-Adressraums und *caLabel* ist die Bezeichnung des CA-Zertifikats.

   Wenn Sie RACF als CA verwenden und ein Zertifikat-Berechtigungszertifikat erstellen müssen, folgen Sie dem Beispiel in Zertifikat der lokalen Zertifizierungsinstanz definieren.
4. Wenn Sie Sicherheitsrichtlinien für den Datenschutz oder die Vertraulichkeit verwenden, um Nachrichten in von AMS geschützten Warteschlangen zu verschlüsseln, verbinden Sie die Zertifikate von Nachrichtenempfängern mit dem Schlüsselring, dessen Eigner die AMS-Adressraumbenutzer-ID ist. Geben Sie den folgenden Befehl ein:

   ```
   RACDCERT ID(amsUser) CONNECT(ID(userId) LABEL('certLabel')
   RING(drq.ams.keyring) USAGE(SITE))
   ```

   Dabei steht *amsUser* für die Benutzer-ID des AMS-Adressraums, *userId* für den Nachrichtenempfänger und *certLabel* für die Bezeichnung des Benutzer-Zertifikats.

   Das Attribut USAGE(SITE) verhindert, dass der private Schlüssel im Schlüsselring zugänglich ist.

   Wenn Sie Ihre eigenen Zertifikate mit RACF erstellen, folgen Sie zum Erstellen des Zertifikats dem Beispiel im Abschnitt Digitales Zertifikat mit einem privaten Schlüssel erstellen.
5. Verbinden Sie die Zertifikate eines jeden Benutzers, der Nachrichten sendet oder empfängt, die von AMS geschützt werden, an einen Schlüsselring, dessen Eigner der Benutzer ist. Das Zertifikat muss als Standardzertifikat in der Schlüsseldatei verbunden sein. Geben Sie den folgenden Befehl ein:

   ```
   RACDCERT ID(userId) CONNECT(ID(userId) LABEL('certLabel')
   RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
   ```

   Hierbei steht *userId* für den Benutzer, der Nachrichten sendet oder empfängt, und *certLabel* ist der Kennsatz des Benutzerzertifikats.

   **Anmerkungen:**

   a. Die Schritte „2” auf Seite 1038 und „5” auf Seite 1038 sind nicht erforderlich, wenn die Anwendung eine Warteschlange nur für die Ausgabe öffnet und Nachrichten an Warteschlangen sendet, die durch eine AMS -Vertraulichkeitsrichtlinie geschützt sind.

   b. Die Schritte „2” auf Seite 1038 und „5” auf Seite 1038 sind nicht erforderlich, wenn die Anwendung eine Warteschlange nur zur Eingabe/Anzeige öffnet und Nachrichten aus Warteschlangen empfängt, die durch eine AMS -Integritätsrichtlinie geschützt sind.

## Nächste Schritte

„Advanced Message Security aktivieren” auf Seite 1039

### z/OS *Advanced Message Security aktivieren*

Die Sicherheitsrichtlinienfunktionalität für einen Warteschlangenmanager wird durch den Parameter SPLCAP im Systemparametermodul gesteuert.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um Advanced Message Security (AMS) für einen einzelnen Warteschlangenmanager zu aktivieren.

Diese Task erfordert, dass Sie eine Änderung am Systemparametermodul vornehmen. Weitere Informationen zum Erstellen und Anpassen des Systemparametermoduls finden Sie im Abschnitt „Passen Sie Ihr Systemparametermodul an" auf Seite 996.

## Vorgehensweise

1. Setzen Sie **SPLCAP** auf YES in CSQ6SYSP. Weitere Informationen zum Makro CSQ6SYSP finden Sie im Abschnitt „CSQ6SYSP verwenden" auf Seite 998.
2. Setzen Sie **AMSPROD** entsprechend Ihren Lizenzberechtigungen entweder auf AMS, ADVANCED oder ADVANCEDVUE. Weitere Informationen zum Makro CSQ6USGP finden Sie unter CSQ6USGP verwenden.
3. Kompilieren Sie das Systemparametermodul erneut.
4. Starten Sie den WS-Manager mit dem aktualisierten Systemparametermodul erneut. Der AMS-Adressraum wird automatisch gestartet, wenn der Warteschlangenmanager gestartet wird.

# z/OS Den mqweb-Server konfigurieren

Verwenden Sie diese Themen als Schritt für Schritt nach Schritt für die Konfiguration des mqweb-Servers.

**Zugehörige Tasks**

„IBM MQ Console und REST API konfigurieren" auf Seite 874

Der mqweb-Server, auf dem die IBM MQ Console und REST API ausgeführt werden, wird mit einer Standardkonfiguration bereitgestellt. Um eine dieser Komponenten verwenden zu können, müssen eine Reihe von Konfigurationstasks ausgeführt werden, z. B. die Konfiguration der Sicherheit, damit Benutzer sich anmelden können. In diesem Thema werden alle Konfigurationsoptionen beschrieben, die verfügbar sind.

### z/OS *Mqweb-Server erstellen*

Wenn Sie IBM MQ for z/OS UNIX System Services Web Components installiert haben und die IBM MQ Console oder die REST API verwenden wollen, müssen Sie den mqweb-Server erstellen und anpassen.

## Vorbereitende Schritte

Bevor Sie das Script **crtmqweb** ausführen, um den mqweb-Server zu erstellen, setzen Sie die Umgebungsvariable JAVA_HOME auf eine 64-Bit-Version von Java auf Ihrem System.

Für IBM MQ Console und administrative REST API ist SYSTEM.REST.REPLY.QUEUE Warteschlange, die erstellt werden soll Erstellen Sie diese Warteschlange mithilfe des **CSQ4INSG** -Beispiels in „Passen Sie die Initialisierungseingabedatensätze an." auf Seite 988.

**Achtung:** Wenn beim Starten des mqweb-Servers die Fehlernachricht CWWKG0014Eangezeigt wird, wie in der folgenden Ausgabe dargestellt:

```
     Launching mqweb (MQM MVS/ESA V9 R2.0/wlp...) (en_US)
               ÝAUDIT   ¨ CWWKE0001I: The server mqweb has been laun□
ched.
               ÝWARNING ¨ CWWKF0009W: The server has not been configured to install any
features.
               ÝAUDIT   ¨ CWWKF0011I: The mqweb server is ready to run a smarter planet.
The mqweb server started in 6.348 seconds.
               ÝERROR   ¨ CWWKG0014E: The configuration parser detected an XML syntax
error while parsing the root of the configuration and the referenced configuration docu□
```

```
ments.
                                Error: An invalid XML character (Unicode: 0x4c) was found
in the prolog of the document.
                                 File: file:<your filepath>/servers/mqweb/server.xml Line:
1 Column: 1
```

Sie sollten die Einstellung z/OS von AUTOCVT überprüfen (Dateien automatisch von einem codierten Zeichensatz in einen anderen konvertieren) und den Wert wie erforderlich anpassen, indem Sie einen der folgenden Schritte ausführen.

**In einem USS-Terminal:**
Geben Sie den Befehl echo **$_BPXK_AUTOCVT** aus, um den Wert dieser Umgebungsvariablen anzuzeigen. Ist die Umgebungsvariable nicht definiert, wird kein Wert angezeigt.

Informationen zum Festlegen der Umgebungsvariable finden Sie unter _BPXK environment variables.

**Systemweit:**
Beispiel 6 unter Status anzeigen von z/OS UNIX System Services (OMVS) zeigt, wie der Wert der systemweiten AUTOCVT-Anweisung in BPXPRMxx angezeigt wird.

Um die Umgebungsvariable systemweit festzulegen, verwenden Sie die Anweisung AUTOCVT in BPXPRMxx.

Wenn die Umgebungsvariable _BPXX_AUTOCVT in einem USS-Terminal festgelegt ist, überschreibt sie die systemweite Einstellung der Anweisung AUTOCVT in BPXPRMxx.

## Informationen zu diesem Vorgang

- Führen Sie diese Task einmal für jedes z/OS -System aus, auf dem IBM MQ Console oder REST APIausgeführt werden soll.
- Für die Verwendung von administrative REST APIbenötigen Sie einen mqweb-Server für jede Version von IBM MQ , die ausgeführt wird. Wenn Sie beispielsweise IBM MQ 9.3.0, 9.2.5 und 9.2.0ausführen, benötigen Sie drei verschiedene mqweb-Server.
- Möglicherweise müssen Sie die Serverkonfiguration aktualisieren oder ändern, wenn Sie von einer früheren Version migrieren.

Für die IBM MQ Console und REST API ist die Erstellung eines einzelnen WebSphere Liberty-Servers namens 'mqweb' erforderlich.

Die Serverkonfigurations- und -protokolldateien sind alle unter dem Liberty-Benutzerverzeichnis gespeichert.

Der mqweb-Server muss mit einer Produkt-ID (PID) konfiguriert werden, unter der er ausgeführt wird. Die PID wird festgelegt, wenn der mqweb-Server erstellt wird. Verwenden Sie dieselbe PID, die für die Ausführung der lokalen Warteschlangenmanager verwendet wird, zu denen der mqweb-Server eine Verbindung herstellt.

**Anmerkung:** Wenn die lokalen Warteschlangenmanager mit mehreren verschiedenen PIDs ausgeführt werden, wählen Sie eine dieser PIDs aus, unter der der mqweb-Server ausgeführt wird.

Weitere Informationen zu PIDs und ihrer Verwendung in z/OSfinden Sie unter Aufzeichnung der Produktnutzung mit IBM MQ for z/OS -Produkten.

Es ist möglich, die PID, unter der der mqweb-Server ausgeführt wird, nach der Erstellung mit dem Befehl setmqweb zu ändern.

Gehen Sie wie folgt vor, um den mqweb-Server zu erstellen:

## Vorgehensweise

1. Entscheiden Sie, unter welcher PID der mqweb-Server ausgeführt wird.
2. Wählen Sie eine geeignete Position für das Liberty-Benutzerverzeichnis aus.

Die Benutzer-ID, unter der der mqweb-Server ausgeführt wird, benötigt Lese-und Schreibzugriff auf dieses Benutzerverzeichnis und seinen Inhalt. Da dieses Benutzerverzeichnis Protokolldateien enthält, erstellen Sie dieses Verzeichnis zusätzlich zur Serverkonfiguration in einem separaten Dateisystem.

**Anmerkung:** Beim Starten des mqweb-Servers gibt es eine beträchtliche Menge an Platten-E/A. Um die zum Starten des Mqweb-Servers benötigte Zeit zu reduzieren, stellen Sie sicher, dass sowohl das Dateisystem IBM MQ des Installationsverzeichnisses z/OS UNIX als auch das Dateisystem des Liberty -Benutzerverzeichnisses sysplexfähig oder lokal auf dem System angehängt sind, auf dem der Mqweb-Server ausgeführt wird.

3. Ändern Sie in z/OS UNIX System ServicesIhr aktuelles Arbeitsverzeichnis in `PathPrefix/web/bin` , indem Sie den folgenden Befehl ausgeben:

   ```
   cd PathPrefix/web/bin
   ```

   Dabei steht *PathPrefix* für den Installationspfad von IBM MQ for z/OS UNIX System Services Components.

4. Erstellen Sie das Benutzerverzeichnis Liberty , das die mqweb-Serverdefinition enthält, indem Sie das Script **crtmqweb** ausführen.

   Das Format des Befehls **crtmqweb** lautet wie folgt:

   ```
   crtmqweb user_directory -p pid_value
   ```

   Dabei gilt:

   **Benutzerverzeichnis**
   Das Liberty -Benutzerverzeichnis, das in Schritt „2” auf Seite 1040festgelegt wurde. Dieser Parameter ist optional. Wenn dieser Parameter nicht angegeben wird, dann wird das Liberty -Standardbenutzerverzeichnis `/var/mqm/web/installation1` verwendet.

   **PID-Wert**
   Gibt die PID an, unter der der mqweb-Server ausgeführt wird. Diese PID ist die, die Sie in Schritt „1” auf Seite 1040ausgewählt haben. *pid_value* ist einer der folgenden Werte:

   **MQ**
   Der mqweb-Server wird unter PID IBM MQ for z/OS ausgeführt (5655-MQ9).

   **VUE**
   Der Mqweb-Server wird unter PID IBM MQ for z/OS Value Unit Edition (VUE) ausgeführt (5655-VU9).

   **ADVANCEDVUE**
   Der mqweb-Server wird unter der PID IBM MQ Advanced for z/OS VUE (5655-AV1) ausgeführt.

   Führen Sie beispielsweise den folgenden Befehl aus, wenn Sie den mqweb-Server mit dem Liberty -Benutzerverzeichnis `/usr/mqweb` und der PID IBM MQ Advanced for z/OS VUE (5655-AV1) erstellen wollen:

   ```
   ./crtmqweb /usr/mqweb -p ADVANCEDVUE
   ```

5. Ändern Sie das Eigentumsrecht für die Verzeichnisse und Dateien im Benutzerverzeichnis Liberty , sodass sie zu der Benutzer-ID und Gruppe gehören, unter der der mqweb-Server ausgeführt wird. Verwenden Sie dazu den folgenden Befehl:

   ```
   chown -R userid:group path
   ```

   Geben Sie den folgenden Befehl aus, um der Gruppe Schreibzugriff auf den Pfad zu erteilen:

   ```
   chmod -R 770 path
   ```

## Nächste Schritte

„Prozedur für den mqweb-Server erstellen” auf Seite 1042

**Zugehörige Tasks**

„IBM MQ Console und REST API konfigurieren” auf Seite 874
Der mqweb-Server, auf dem die IBM MQ Console und REST API ausgeführt werden, wird mit einer Standardkonfiguration bereitgestellt. Um eine dieser Komponenten verwenden zu können, müssen eine Reihe von Konfigurationstasks ausgeführt werden, z. B. die Konfiguration der Sicherheit, damit Benutzer sich anmelden können. In diesem Thema werden alle Konfigurationsoptionen beschrieben, die verfügbar sind.

### z/OS *Prozedur für den mqweb-Server erstellen*

Wenn Sie die IBM MQ for z/OS UNIX System Services Web Components installiert haben und die IBM MQ Console oder die REST API verwenden möchten, müssen Sie eine katalogisierte Prozedur erstellen, um den mqweb-Server zu starten. Der mqweb-Server ist ein Liberty-Server, auf dem sich IBM MQ Console und REST API befinden.

- Sie müssen diese Task einmal für jedes z/OS-System ausführen, auf dem die IBM MQ Console oder REST API ausgeführt werden soll.
- Sie benötigen einen mqweb-Server für jede Version von IBM MQ, die ausgeführt wird. Beispiel: Eine gestartete Task namens 'MQWB0910' für Warteschlangenmanager in IBM MQ for z/OS 9.1.0 und eine gestartete Task namens 'MQWB0905' für Warteschlangenmanager in IBM MQ for z/OS 9.0.5.

  Wenn Sie nur einen Warteschlangenmanager auf dem z/OS-System haben, können Sie eine einzelne gestartete Task des Liberty-Servers ausführen und die verwendeten Bibliotheken ändern, wenn Sie den Warteschlangenmanager migrieren.
- Möglicherweise müssen Sie die katalogisierte Prozedur ändern, wenn Sie eine Migration von einer früheren Version durchführen.

Gehen Sie wie folgt vor, um eine katalogisierte Prozedur zu erstellen:

1. Kopieren Sie die Beispielprozedur für die gestartete Task `thlqual.SCSQPROC(CSQ4WEBS)` in die Prozedurbibliothek.

   Benennen Sie das Verfahren gemäß den Standards Ihres Unternehmens.

   Mit `MQWB0910` geben sie z. B. an, dass es sich um die katalogisierte Prozedur für den IBM MQ for z/OS 9.1.0-mqweb-Server handelt.
2. Passen Sie die Prozedur anhand der Anweisungen in der Beispielprozedur CSQ4WEBS an Ihre Anforderungen an.

   Beachten Sie, dass das Liberty-Benutzerverzeichnis das Verzeichnis ist, das beim Ausführen des Scripts **`crtmqweb`** angegeben wurde, um die mqweb-Serverdefinition zu erstellen.

   Ausführliche Informationen finden Sie in „Mqweb-Server erstellen” auf Seite 1039.

   **Anmerkung:** Stellen Sie sicher, dass Sie **Caps off** beim Bearbeiten des Mitglieds angeben, da die Datei Daten in Kleinschreibung enthält.
3. Autorisieren Sie die Prozedur, die unter Ihrem externen Sicherheitsmanager ausgeführt werden soll.
4. Verwenden Sie IBM Workload Manager (WLM), um diesen Adressraum zu klassifizieren.

   Der mqweb-Server ist eine IBM MQ-Anwendung, mit der die Benutzer interagieren. Die Anwendung muss in WLM nicht von großer Bedeutung sein, und es kann eine Serviceklasse von **STCUSER** geeignet sein.

## Nächste Schritte

Führen Sie die Schritte in „Basiskonfiguration für den mqweb-Server” auf Seite 875 aus, um die Konfiguration des mqweb-Servers abzuschließen.

**Zugehörige Tasks**

„IBM MQ Console und REST API konfigurieren” auf Seite 874
Der mqweb-Server, auf dem die IBM MQ Console und REST API ausgeführt werden, wird mit einer Standardkonfiguration bereitgestellt. Um eine dieser Komponenten verwenden zu können, müssen eine

Reihe von Konfigurationstasks ausgeführt werden, z. B. die Konfiguration der Sicherheit, damit Benutzer sich anmelden können. In diesem Thema werden alle Konfigurationsoptionen beschrieben, die verfügbar sind.

# z/OS Warteschlangenmanager auf z/OS testen

Wenn Sie Ihren Warteschlangenmanager angepasst oder migriert haben, können Sie ihn testen, indem Sie die Installationsprüfprogramme und einige der Beispielanwendungen ausführen, die mit IBM MQ for z/OS geliefert wurden.

## Informationen zu diesem Vorgang

Nachdem Sie IBM MQ for z/OS installiert und angepasst haben, können Sie das mitgelieferte Installationsprüfprogramm (CSQ4IVP1) verwenden, um zu bestätigen, dass IBM MQ for z/OS betriebsbereit ist.

Das Basisinstallationsprogramm CSQ4IVP1 testet nicht gemeinsam genutzte Warteschlangen und prüft die Basis-IBM MQ, ohne die Beispiele C, COBOL oder CICS zu verwenden.

Nach der Ausführung der grundlegenden Installationsprüfung können Sie für gemeinsam genutzte Warteschlangen testen, indem Sie CSQ4IVP1 mit verschiedenen Warteschlangen verwenden. Außerdem können Sie testen, ob Db2 und die Coupling-Facility ordnungsgemäß konfiguriert sind. Um zu bestätigen, dass die verteilte Steuerung von Warteschlangen betriebsbereit ist, können Sie das mitgelieferte Installationsprüfprogramm CSQ4IVPX verwenden.

CSQ4IVP1 wird als Lademodul bereitgestellt und stellt eine Reihe von prozeduralen Beispielanwendungen als Quellenmodule zur Verfügung, die typische Verwendungen der Schnittstelle für Nachrichtenwarteschlangen (MQI) veranschaulichen. Sie können diese Quellenmodule verwenden, um verschiedene Programmiersprachenumgebungen zu testen. Sie können die anderen Mustercodes kompilieren und mit der mitgelieferten Beispiel-JCL verbinden, die die anderen Beispiele für Ihre Installation geeignet sind.

## Prozedur

- Informationen zum Testen des Warteschlangenmanagers unter z/OS finden Sie in den folgenden Unterabschnitten:
  - „Ausführen des Basisinstallationsprüfprogramms” auf Seite 1043
  - „Gruppen mit gemeinsamer Warteschlange testen” auf Seite 1047
  - „Tests für verteilte Steuerung von Warteschlangen” auf Seite 1048
  - „Tests für C-, C++-, COBOL-, PL/I- und CICS-Programme mit IBM MQ for z/OS” auf Seite 1051

**Zugehörige Konzepte**

IBM MQ for z/OS - Konzepte

**Zugehörige Tasks**

IBM MQ-Umgebung unter z/OS planen

„Warteschlangenmanager unter z/OS erstellen” auf Seite 966

Verwenden Sie diese Anweisungen zum Konfigurieren von Warteschlangenmanagern unter IBM MQ for z/OS.

IBM MQ for z/OS verwalten

# z/OS Ausführen des Basisinstallationsprüfprogramms

Nachdem Sie IBM MQ installiert und angepasst haben, können Sie das mitgelieferte Installationsprüfprogramm, CSQ4IVP1, verwenden, um zu bestätigen, dass IBM MQ betriebsbereit ist.

Das Basisinstallationsprüfprogramm ist ein Stapelverarbeitungsprogramm, das die Basis IBM MQ ohne Verwendung der C-, COBOL- oder CICS-Beispiele überprüft.

Der Batch Assembler IVP wird von SMP/E verlinkseditiert und die Lademodule werden in der Bibliothek thlqual.SCSQLOAD. ausgeliefert.

Nachdem Sie sowohl den SMP/E-APPLY-Schritt als auch die Anpassungsschritte ausgeführt haben, führen Sie den Batch Assembler IVP aus.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- Übersicht über die Anwendung CSQ4IVP1
- Ausführung von CSQ4IVP1 vorbereiten
- CSQ4IVP1 ausführen
- Ergebnisse von CSQ4IVP1 überprüfen

## Übersicht über die Anwendung CSQ4IVP1

CSQ4IVP1 ist eine Stapelanwendung, die eine Verbindung zu Ihrem IBM MQ-Subsystem herstellt und die folgenden Basisfunktionen ausführt:

- Gibt IBM MQ-Aufrufe aus
- Kommuniziert mit dem Befehlsserver
- Prüft, ob die Auslösung aktiv ist.
- Generiert und löscht eine dynamische Warteschlange
- Prüft die Nachrichtenverfallsverarbeitung
- Prüft die Nachrichtencommitverarbeitung

## Vorbereiten der Ausführung von CSQ4IVP1

Vor der Ausführung von CSQ4IVP1:

1. Überprüfen Sie, ob sich die IVP-Einträge in der CSQINP2-Datenmenge Verkettung im Startprogramm des Warteschlangenmanagers befinden. Die IVP-Einträge werden in der Teildatei thlqual.SCSQPROC (CSQ4IVPQ) bereitgestellt. Ist dies nicht der Fall, fügen Sie die in thlqual.SCSQPROC (CSQ4IVPQ) angegebenen Definitionen zu Ihrer CSQINP2-Verkettung hinzu. Wenn der Warteschlangenmanager derzeit aktiv ist, müssen Sie ihn erneut starten, damit diese Definitionen wirksam werden können.
2. Die Beispiel-JCL CSQ4IVPR, die für die Ausführung des Installationsprüfprogramms erforderlich ist, befindet sich in der Bibliothek thlqual.SCSQPROC.

   Passen Sie die JCL CSQ4IVPR an, indem Sie das übergeordnete Qualifikationsmerkmal für die IBM MQ-Bibliotheken, die zu verwendende Landessprache, den vierstelligen Namen des IBM MQ-Warteschlangenmanagers und das Ziel für die Jobausgabe angeben.
3. Aktualisieren Sie RACF so, dass CSQ4IVP1 auf seine Ressourcen zugreifen kann, wenn die IBM MQ-Sicherheitsfunktion aktiv ist.

   Um die CSQ4IVP1-Funktion auszuführen, wenn die IBM MQ-Sicherheit aktiviert ist, benötigen Sie eine RACF-Benutzer-ID, die zum Zugriff auf die Objekte berechtigt ist. Details zum Definieren von Ressourcen für RACF finden Sie unter Sicherheit für z/OS einrichten. Die Benutzer-ID, unter der der IVP ausgeführt wird, muss über die folgende Zugriffsberechtigung verfügen:

| Berechtigung | Profil | Klasse |
|---|---|---|
| READ | ssid.DISPLAY.PROCESS | MQCMDS |
| UPDATE | ssid.SYSTEM.COMMAND.INPUT | MQQUEUE |
| UPDATE | ssid.SYSTEM.COMMAND.REPLY.MODEL | MQQUEUE |
| UPDATE | ssid.CSQ4IVP1.** | MQQUEUE |
| READ | ssid.BATCH | MQCONN |

Bei diesen Anforderungen wird davon ausgegangen, dass die gesamte IBM MQ-Sicherheitsfunktion aktiv ist. Die RACF -Befehle zum Aktivieren der IBM MQ -Sicherheit werden in Abbildung 101 auf Seite 1045gezeigt. In diesem Beispiel wird davon ausgegangen, dass der Name des WS-Managers CSQ1 lautet und dass die Benutzer-ID der Person, auf der CSQ4IVP1 ausgeführt wird, TS101 ist.

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

*Abbildung 101. RACF-Befehle für CSQ4IVP1*

## CSQ4IVP1 wird ausgeführt

Wenn Sie diese Schritte ausgeführt haben, starten Sie den Warteschlangenmanager. Wenn der WS-Manager bereits aktiv ist und Sie CSQINP2 geändert haben, müssen Sie den Warteschlangenmanager stoppen und erneut starten.

Der IVP wird als Stapeljob ausgeführt. Passen Sie die Jobkarte so an, dass sie den Einreichungsvoraussetzungen für Ihre Installation entspricht.

## Überprüfen der Ergebnisse von CSQ4IVP1

Der IVP wird in 10 Stufen aufgeteilt; jede Stufe muss mit einem Nullbeendigungscode abgeschlossen werden, bevor die nächste Stufe ausgeführt wird. Der IVP generiert einen Bericht, in dem Folgendes aufgeführt ist:

- Der Name des Warteschlangenmanagers, mit dem eine Verbindung hergestellt wird.
- Eine Einzeilennachricht mit dem Beendigungscode und dem Ursachencode, der von jeder Stage zurückgegeben wurde.
- Eine einzeileleine Informationsnachricht, falls zutreffend.

In Abbildung 102 auf Seite 1047 wird ein Beispielbericht gezeigt.

z/OS Eine Erläuterung der Beendigungs- und Ursachencodes finden Sie in den IBM MQ for z/OS-Nachrichten, Beendigungs- und Ursachencodes.

Einige Schritte enthalten mehr als einen IBM MQ-Aufruf und im Falle eines Fehlers wird eine Nachricht ausgegeben, die den jeweiligen IBM MQ-Aufruf angibt, der den Fehler zurückgegeben hat. Für einige Phasen reiht der IVP außerdem erläuternden und Diagnoseinformationen in ein Kommentarfeld ein.

Der IVP-Job fordert die exklusive Steuerung bestimmter WS-Manager-Objekte an und sollte daher mit einem einzigen Thread über das System verbunden werden. Es gibt jedoch keine Begrenzung für die Anzahl der Ausführungszeiten des Installationsprüfzifferns (IVP) für den Warteschlangenmanager.

Die Funktionen der einzelnen Phasen lauten wie folgt:

**Stufe 1**
Stellen Sie eine Verbindung zum WS-Manager her, indem Sie den API-Aufruf `MQCONN` absetzen.

**Stufe 2**
Bestimmen Sie den Namen der Eingabewarteschlange des Systembefehls, die vom Befehlsserver zum Abrufen von Anforderungsnachrichten verwendet wird. Diese Warteschlange empfängt Anzeigeanforderungen von Stufe 5.

Dazu ist die Reihenfolge der Aufrufe wie folgt:

1. Geben Sie einen `MQOPEN` -Aufruf aus, und geben Sie dabei den Namen des Warteschlangenmanagers an, um das WS-Manager-Objekt zu öffnen.
2. Geben Sie einen `MQINQ` -Aufruf aus, um den Namen der Eingabewarteschlange für Systembefehle zu ermitteln.
3. Geben Sie einen `MQINQ` -Aufruf aus, um Informationen zu verschiedenen Ereignisschaltern des Warteschlangenmanagers zu erhalten.
4. Setzen Sie den Aufruf `MQCLOSE` ab, um das WS-Manager-Objekt zu schließen.

Nach dem erfolgreichen Abschluss dieser Phase wird der Name der Eingabewarteschlange für Systembefehle im Kommentarfeld angezeigt.

**Stufe 3**
Öffnen Sie eine Initialisierungswarteschlange mit einem **MQOPEN** -Aufruf.

Diese Warteschlange wird zu diesem Zeitpunkt im Vorgriff auf eine Auslösenachricht geöffnet, die als Ergebnis des Befehlsservers, der auf die Anforderung von Stufe 5 antwortet, ankommt. Die Warteschlange muss für die Eingabe geöffnet werden, damit die Auslöserkriterien erfüllt werden können.

**Stufe 4**
Erstellen Sie eine permanente dynamische Warteschlange mit Hilfe der Warteschlange CSQ4IVP1.MODEL als Modell. Die dynamische Warteschlange verfügt über dieselben Attribute wie das Modell, aus dem sie erstellt wurde. Dies bedeutet, dass eine Auslösenachricht in die Initialisierungswarteschlange geschrieben wird, die in Phase 3 geöffnet wurde, wenn die Antworten von der Befehlsserveranforderung in Stufe 5 in diese Warteschlange geschrieben werden.

Nach dem erfolgreichen Abschluss dieser Phase wird der Name der permanenten dynamischen Warteschlange im Kommentarfeld angezeigt.

**Stufe 5**
Geben Sie eine `MQPUT1` -Anforderung an die Befehlswarteschlange des Befehlsservers aus.

Eine Nachricht vom Typ MQMT_REQUEST wird in die Eingabewarteschlange des Systembefehls geschrieben, in der eine Anzeige des Prozesses CSQ4IVP1 angefordert wird. Der Nachrichtendeskriptor für die Nachricht gibt die permanente dynamische Warteschlange an, die in Stage 4 als Antwort-Warteschlange für die Antwort des Befehlsservers erstellt wurde.

**Stufe 6**
Setzen Sie eine **MQGET** -Anforderung aus der Initialisierungswarteschlange ab. In dieser Phase wird ein GET WAIT mit einem Intervall von 1 Minute für die in Stage 3 geöffnete Initialisierungswarteschlange ausgegeben. Es wird erwartet, dass die zurückgegebene Nachricht die Auslösenachricht sein wird, die von den Antwortnachrichten des Befehlsservers generiert wird, die in die Empfangswarteschlange für Antworten geschrieben werden.

**Stufe 7**
Löschen Sie die permanente dynamische Warteschlange, die in Stage 4 erstellt wurde. Da in der Warteschlange noch Nachrichten vorhanden sind, wird die Option MQCO_PURGE_DELETE verwendet.

**Stufe 8**

1. Öffnen Sie eine dynamische Warteschlange.
2. MQPUT-Nachricht mit einem Ablaufintervall-Set.
3. Warten Sie, bis die Nachricht abgelaufen ist.
4. Es wurde versucht, die abgelaufene Nachricht zu MQGET zu verwenden.
5. MQCLOSE die Warteschlange.

**Stufe 9**

1. Öffnen Sie eine dynamische Warteschlange.
2. MQPUT-Nachricht.
3. Setzen Sie MQCMIT ab, um die aktuelle UOG festzuschreiben.
4. MQGET-Nachricht.
5. Setzen Sie MQBACK ab, um die Nachricht zu sichern.
6. MQGET dieselbe Nachricht und stellen Sie sicher, dass der Rücksettzähler auf 1 gesetzt ist.
7. Setzen Sie MQCLOSE ab, um die Warteschlange zu schließen.

**Stufe 10**

Trennen Sie die Verbindung zum Warteschlangenmanager mit **MQDISC**.

Nachdem Sie den IVP ausgeführt haben, können Sie alle Objekte löschen, die Sie nicht mehr benötigen.

Wenn der IVP nicht erfolgreich ausgeführt wird, versuchen Sie jeden Schritt manuell, um herauszufinden, welche Funktion fehlgeschlagen ist.

```
DATE : 2005.035                IBM MQ for z/OS - V6            PAGE : 0001
INSTALLATION VERIFICATION PROGRAM
PARAMETERS ACCEPTED. PROGRAM WILL CONNECT TO : CSQ1
,OBJECT QUALIFER : CSQ4IVP1
INSTALLATION VERIFICATION BEGINS :
STAGE 01 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR BRIDGE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS EXCP FOR CHANNEL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR SSL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR INHIBITED EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR LOCAL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR PERFORMENCE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR REMOTE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR START/STOP EVENTS
STAGE 02 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 SYSTEM.COMMAND.INPUT
STAGE 03 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 04 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810EFEAC8980
STAGE 05 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 06 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 07 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 08 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810F0070E645
STAGE 09 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9812BA8706803
STAGE 10 COMPLETE. COMPCODE : 0000 REASON CODE : 0000>>>>>>>>>>   END OF REPORT   <<<<<<<<<<
```

*Abbildung 102. Beispielbericht von CSQ4IVP1*

## z/OS Gruppen mit gemeinsamer Warteschlange testen

Das Basisinstallationsprogramm CSQ4IVP1 testet nicht gemeinsam genutzte Warteschlangen.

CSQ4IVP1 kann unabhängig davon verwendet werden, ob der Warteschlangenmanager ein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist oder nicht. Nach der Ausführung des grundlegenden Installationsprüfprogramms können Sie mit Hilfe des Installationsprüfprogramms CSQ4IVP1 mit verschiedenen Warteschlangen nach gemeinsam genutzten Warteschlangen testen. Außerdem wird hierdurch geprüft, ob Db2 und die Coupling Facility korrekt konfiguriert sind.

### Ausführung von CSQ4IVP1 für eine Gruppe mit gemeinsamer Warteschlange vorbereiten

Vor der Ausführung von CSQ4IVP1:

1. Fügen Sie wie in „Einrichten der Coupling-Facility” auf Seite 1028 beschrieben die vom IVP verwendete Coupling-Facility-Struktur zu Ihrer CFRM-Richtlinie hinzu. Die bereitgestellten Beispiele verwenden eine Struktur mit dem Namen APPLICATION1. Sie können diese jedoch ändern, wenn Sie möchten.
2. Überprüfen Sie, ob sich die IVP-Einträge in der CSQINP2-Datenmenge Verkettung im Startprogramm des Warteschlangenmanagers befinden. Die IVP-Einträge werden in der Teildatei thlqual.SCSQPROC (CSQ4IVPG) bereitgestellt. Ist dies nicht der Fall, fügen Sie die Definitionen in thlqual.SCSQPROC

(CSQ4IVPG) zu Ihrer CSQINP2-Verkettung hinzu. Wenn der Warteschlangenmanager derzeit aktiv ist, müssen Sie ihn erneut starten, damit diese Definitionen wirksam werden können.

3. Ändern Sie bei Bedarf den Namen der Coupling-Facility-Struktur, die in thlqual.SCSQPROC (CSQ4IVPG) verwendet wird.
4. Die Beispiel-JCL CSQ4IVPS, die für die Ausführung des Installationsprüfprogramms für eine Gruppe mit gemeinsamer Warteschlange erforderlich ist, befindet sich in der Bibliothek 'thlqual.SCSQPROC'.

   Passen Sie die JCL CSQ4IVPS mit dem übergeordneten Qualifikationsmerkmal für die IBM MQ-Bibliotheken, der Landessprache, die Sie verwenden möchten, dem vierstelligen Namen des IBM MQ-Warteschlangenmanagers und dem Ziel für die Jobausgabe an.
5. Aktualisieren Sie RACF so, dass CSQ4IVP1 auf seine Ressourcen zugreifen kann, wenn die IBM MQ-Sicherheitsfunktion aktiv ist.

   Um die CSQ4IVP1-Funktion auszuführen, wenn die IBM MQ-Sicherheit aktiviert ist, benötigen Sie eine RACF-Benutzer-ID, die zum Zugriff auf die Objekte berechtigt ist. Details zum Definieren von Ressourcen für RACF finden Sie unter Sicherheit für z/OS einrichten. Die Benutzer-ID, die den IVP ausführt, muss über die folgende Zugriffsberechtigung verfügen, die zusätzlich zu der für die Ausführung des Basis-IVP erforderlichen Zugriffsberechtigung erforderlich ist:

| Berechtigung | Profil | Klasse |
|---|---|---|
| UPDATE | ssid.CSQ4IVPG.** | MQQUEUE |

Bei diesen Anforderungen wird davon ausgegangen, dass die gesamte IBM MQ-Sicherheitsfunktion aktiv ist. Die RACF-Befehle zum Aktivieren der IBM MQ-Sicherheit werden in Abbildung 103 auf Seite 1048 angezeigt. In diesem Beispiel wird davon ausgegangen, dass der Name des WS-Managers CSQ1 lautet und dass die Benutzer-ID der Person, auf der CSQ4IVP1 ausgeführt wird, TS101 ist.

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

*Abbildung 103. RACF-Befehle für CSQ4IVP1 für eine Gruppe mit gemeinsamer Warteschlange*

## CSQ4IVP1 für eine Gruppe mit gemeinsamer Warteschlange ausführen

Wenn Sie diese Schritte ausgeführt haben, starten Sie den Warteschlangenmanager. Wenn der WS-Manager bereits aktiv ist und Sie CSQINP2 geändert haben, müssen Sie den Warteschlangenmanager stoppen und erneut starten.

Der IVP wird als Stapeljob ausgeführt. Passen Sie die Jobkarte so an, dass sie den Einreichungsvoraussetzungen für Ihre Installation entspricht.

## Ergebnisse von CSQ4IVP1 für eine Gruppe mit gemeinsamer Warteschlange prüfen

Der IVP für Gruppe mit gemeinsamer Warteschlange funktioniert auf die gleiche Weise wie der Basis-IVP, mit der Ausnahme, dass die erstellten Warteschlangen die Bezeichnung CSQIVPG haben. *xx*. Folgen Sie den Anweisungen im Abschnitt „Überprüfen der Ergebnisse von CSQ4IVP1” auf Seite 1045, um die Ergebnisse des IVP für Gruppen mit gemeinsamer Warteschlange zu prüfen.

## z/OS Tests für verteilte Steuerung von Warteschlangen

Sie können das mitgelieferte Installationsprüfprogramm (CSQ4IVPX) verwenden, um zu bestätigen, dass die verteilte Steuerung von Warteschlangen betriebsbereit ist.

## Übersicht über den Job CSQ4IVPX

CSQ4IVPX ist ein Stapeljob, der den Kanalinitiator startet und den Befehl IBM MQ DISPLAY CHINIT ausgibt. Dadurch wird sichergestellt, dass alle wichtigen Aspekte der verteilten Steuerung von Wartesch-langen betriebsbereit sind, während die Notwendigkeit zum Festlegen von Kanal-und Netzdefinitionen vermieden wird.

## Ausführung von CSQ4IVPX wird vorbereitet

Vor der Ausführung von CSQ4IVPX:

1. Die JCL-Beispiel-JCL CSQ4IVPX, die für die Ausführung des Installationsprüfprogramms erforderlich ist, befindet sich in der Bibliothek thlqual.SCSQPROC.

   Passen Sie die JCL CSQ4IVPX an, indem Sie das übergeordnete Qualifikationsmerkmal für die IBM MQ-Bibliotheken, die zu verwendende Landessprache, den vierstelligen Namen des Warteschlangen-managers und die Zieladresse für die Jobausgabe angeben.
2. Aktualisieren Sie RACF so, dass CSQ4IVPX auf seine Ressourcen zugreifen kann, wenn die IBM MQ-Sicherheitsfunktion aktiv ist. Um die CSQ4IVPX-Funktion auszuführen, wenn die IBM MQ-Sicherheit aktiviert ist, benötigen Sie eine RACF-Benutzer-ID, die zum Zugriff auf die Objekte berechtigt ist. Details zum Definieren von Ressourcen für RACF finden Sie unter Sicherheit für z/OS einrichten. Die Benutzer-ID, unter der der IVP ausgeführt wird, muss über die folgende Zugriffsberechtigung verfügen:

| Berechtigung | Profil | Klasse |
|---|---|---|
| CONTROL | ssid.START.CHINIT und ssid.STOP.CHINIT | MQCMDS |
| UPDATE | ssid.SYSTEM.COMMAND.INPUT | MQQUEUE |
| UPDATE | ssid.SYSTEM.CSQUTIL.* | MQQUEUE |
| READ | ssid.BATCH | MQCONN |
| READ | ssid.DISPLAY.CHINIT | MQCMDS |

   Bei diesen Anforderungen wird davon ausgegangen, dass das Verbindungssicherheitsprofil ssid.CHIN definiert wurde (wie in Verbindungssicherheitsprofile für den Kanalinitiator dargestellt) und dass die gesamte IBM MQ-Sicherheitsfunktion aktiv ist. Die RACF-Befehle dazu sind in Abbildung 104 auf Seite 1050 angezeigt. In diesem Beispiel wird Folgendes vorausgesetzt:

   - Der Name des WS-Managers lautet CSQ1.
   - Die Benutzer-ID des Benutzers, auf dem CSQ4IVPX ausgeführt wird, ist TS101.
   - Der Adressraum des Kanalinitiators wird unter der Benutzer-ID CSQ1MSTR ausgeführt.
3. Aktualisieren Sie RACF, um dem Adressraum des Kanalinitiators die folgende Zugriffsberechtigung zu ermöglichen:

| Berechtigung | Profil | Klasse |
|---|---|---|
| READ | ssid.CHIN | MQCONN |
| UPDATE | ssid.SYSTEM.COMMAND.INPUT | MQQUEUE |
| UPDATE | ssid.SYSTEM.CHANNEL.INITQ | MQQUEUE |
| UPDATE | ssid.SYSTEM.CHANNEL.SYNCQ | MQQUEUE |
| ALTER | ssid.SYSTEM.CLUSTER.COMMAND.QUEUE | MQQUEUE |
| UPDATE | ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE | MQQUEUE |
| ALTER | ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE | MQQUEUE |
| CONTROL | ssid.CONTEXT.** | MQADMIN |

Die RACF-Befehle dazu werden auch in Abbildung 104 auf Seite 1050angezeigt.

```
RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)
```

*Abbildung 104. RACF-Befehle für CSQ4IVPX*

## CSQ4IVPX wird ausgeführt

Wenn Sie diese Schritte ausgeführt haben, starten Sie den Warteschlangenmanager.

Der IVP wird als Stapeljob ausgeführt. Passen Sie die Jobkarte so an, dass sie den Einreichungsvoraussetzungen für Ihre Installation entspricht.

## Überprüfen der Ergebnisse von CSQ4IVPX

CSQ4IVPX führt das Dienstprogramm CSQUTIL IBM MQ aus, um drei MQSC-Befehle auszugeben. Die SYSPRINT-Ausgabedatei sollte wie in Abbildung 105 auf Seite 1051 dargestellt aussehen, obwohl die Details in Abhängigkeit von den Attributen Ihres Warteschlangenmanagers unterschiedlich sein können.

- Die Befehle **(1)** , jeweils gefolgt von mehreren Nachrichten, sollten angezeigt werden.
- Die letzte Nachricht aus jedem Befehl sollte lauten."CSQ9022I ... NORMALE BEENDIGUNG" **(2)**.
- Der Job als Ganzes sollte mit dem Rückkehrcode Null **(3)** ausgeführt werden.

```
CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I  COUNT=   12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRDQM Channel initiator active
CSQX002I +CSQ1 CSQXRDQM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRDQM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRDQM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRDQM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRDQM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRDQM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRDQM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRDQM TCP/IP system name is TCPIP
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)
```

*Abbildung 105. Beispielausgabe von CSQ4IVPX*

## z/OS Tests für C-, C++-, COBOL-, PL/I- und CICS-Programme mit IBM MQ for z/OS

Sie können für C, C++, COBOL, PL/I oder CICS die Beispielanwendungen testen, die mit IBM MQ bereitgestellt werden.

Das IVP (CSQ4IVP1) wird als Lademodul bereitgestellt und stellt die Beispiele als Quellenmodule zur Verfügung. Sie können diese Quellenmodule verwenden, um verschiedene Programmiersprachenumgebungen zu testen.

Weitere Informationen zu Beispielanwendungen finden Sie in Beispielanwendungen fürIBM MQ for z/OS .

# z/OS Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten

In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

## Informationen zu diesem Vorgang

Zum Definieren der Anforderungen für die verteilte Steuerung von Warteschlangen müssen Sie die folgenden Elemente definieren:

- Kanalinitiatorprozeduren und -dateien
- Kanaldefinitionen
- Warteschlangen und andere Objekte
- Zugriffsschutz

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, lesen Sie den Abschnitt Verteilte Warteschlangen- und Warteschlangen-Sharing-Gruppen.

Weitere Punkte, die bei der Konfiguration der verteilten Steuerung von Warteschlangen mit IBM MQ for z/OSberücksichtigt werden müssen, finden Sie unter „Hinweise zur Verwendung der verteilten Steuerung von Warteschlangen unter z/OS" auf Seite 1052.

## Vorgehensweise

Gehen Sie wie folgt vor, um die verteilte Steuerung von Warteschlangen zu aktivieren:

- Passen Sie die verteilte Warteschlangenfunktion an und definieren Sie die IBM MQ-Objekte, die gemäß der Beschreibung in Systemobjekte definieren und „Vorbereiten der Anpassung von Warteschlangenmanagern unter z/OS" auf Seite 967 erforderlich sind.
- Definieren Sie die Zugriffssicherheit wie in Sicherheitsaspekte für den Kanalinitiator unter z/OS beschrieben.
- Richten Sie Ihre Kommunikation gemäß der Beschreibung in „Kommunikation für z/OS konfigurieren" auf Seite 1073 ein.

**Zugehörige Konzepte**

„IBM MQ for z/OS einrichten" auf Seite 972
Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren" auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

# z/OS Hinweise zur Verwendung der verteilten Steuerung von Warteschlangen unter z/OS

Punkte, die zu berücksichtigen sind, wenn Sie die Verwendung der verteilten Steuerung von Warteschlangen unter z/OSvorbereiten

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, lesen Sie den Abschnitt Verteilte Warteschlangen- und Warteschlangen-Sharing-Gruppen.

## Bedienernachrichten

Da der Kanalinitiator mehrere asynchron arbeitende Dispatcher verwendet, können die Bedienernachrichten im Protokoll aus chronologischer Reihenfolge auftreten.

## Kanaloperationsbefehle

Kanaloperationsbefehle umfassen in der Regel zwei Stufen. Wenn die Befehlssyntax geprüft wurde und die Existenz des Kanals geprüft wurde, wird eine Anforderung an den Kanalinitiator gesendet. Die Nachricht CSQM134I oder CSQM137I wird an den Befehlsaussteller gesendet, um den Abschluss der ersten Phase anzuzeigen. Wenn der Kanalinitiator den Befehl verarbeitet hat, werden weitere Nachrichten, die angeben, ob der Befehl erfolgreich war oder nicht, zusammen mit der Nachricht CSQ9022I oder CSQ9023Ean den Befehlsaussteller gesendet. Alle generierten Fehlernachrichten können auch an die z/OS-Konsole gesendet werden.

Alle Clusterbefehle mit Ausnahme von **DISPLAY CLUSQMGR**funktionieren jedoch asynchron. Befehle, die Objektattribute ändern, aktualisieren das Objekt und senden eine Anforderung an den Kanalinitiator. Befehle zum Arbeiten mit Clustern werden auf die Syntax überprüft, und eine Anforderung wird an den Kanalinitiator gesendet. In beiden Fällen wird die Nachricht CSQM130I an den Befehlsaussteller gesendet, der angibt, dass eine Anforderung gesendet wurde. Auf diese Nachricht folgt die Nachricht CSQ9022I , die angibt, dass der Befehl erfolgreich ausgeführt wurde, da eine Anforderung gesendet wurde. Er gibt nicht an, dass die Clusteranforderung erfolgreich abgeschlossen wurde. Die Anforderungen, die an den Kanalinitiator gesendet werden, werden asynchron verarbeitet, zusammen mit Clusteranforderungen, die von anderen Membern des Clusters empfangen wurden. In einigen Fällen müssen diese Anforderungen an den gesamten Cluster gesendet werden, um festzustellen, ob sie erfolgreich sind oder nicht. Alle Fehler werden an den z/OS auf dem System gemeldet, auf dem der Kanalinitiator ausgeführt wird. Sie werden nicht an den Befehlsaussteller gesendet.

## Nicht zugegebene Nachrichtenwarteschlange

Ein Handler für nicht zustellbare Mail wird mit IBM MQ for z/OS bereitgestellt. Weitere Informationen finden Sie unter Dienstprogramm für die Steuerroutine der Warteschlange für nicht zustellbare Nachrichten (CSQUDLQH).

## Warteschlangen im Gebrauch

MCAs für Empfängerkanäle können die Zielwarteschlangen auch dann offen halten, wenn Nachrichten nicht übertragen werden. Dieses Verhalten führt dazu, dass die Warteschlangen im Gebrauch 'im Gebrauch' sind.

## Sicherheitsänderungen

Wenn Sie den Sicherheitszugriff für eine Benutzer-ID ändern, wird die Änderung möglicherweise nicht sofort wirksam. Weitere Informationen finden Sie unter Sicherheitsaspekte für den Kanalinitiator unter z/OS , Profile für Warteschlangensicherheitund „Implementieren Sie Ihre ESM-Sicherheitskontrollen.” auf Seite 987.

## Kommunikation gestoppt-TCP

Wenn TCP aus irgendeinem Grund gestoppt und anschließend erneut gestartet wird, wird der TCP-Listener von IBM MQ for z/OS, der auf einen TCP-Port wartet, gestoppt.

Durch die automatische Kanalwiederverbindung kann der Kanalinitiator feststellen, dass TCP/IP nicht verfügbar ist, und den TCP/IP-Listener automatisch erneut starten, wenn TCP/IP zurückgegeben wird. Dieser automatische Neustart verringert die Notwendigkeit von Betriebspersonal, das Problem mit TCP/IP zu erkennen und den Listener manuell erneut zu starten. Während das Empfangsprogramm außer Funktion ist, kann der Kanalinitiator auch verwendet werden, um das Empfangsprogramm in dem durch LSTRTMRangegebenen Intervall erneut zu versuchen. Diese Versuche können fortgesetzt werden, bis TCP/IP zurückkehrt und das Empfangsprogramm automatisch erneut gestartet wird. Weitere Informationen zu LSTRTMRfinden Sie unter ALTER QMGR und Distributed queuing messages (CSQX ...).

## Kommunikation gestoppt-LU6.2

Wenn APPC gestoppt wird, wird auch der Listener gestoppt. Auch in diesem Fall wiederholt das Empfangsprogramm automatisch den Versuch im Intervall LSTRTMR , damit das Empfangsprogramm bei einem APPC-Neustart ebenfalls erneut gestartet werden kann.

Wenn der Db2 fehlschlägt, werden die gemeinsam genutzten Kanäle, die bereits ausgeführt werden, weiterhin ausgeführt, aber alle neuen Kanalstartanforderungen schlagen fehl. Wenn die Db2 wiederhergestellt ist, können neue Anforderungen ausgeführt werden.

## z/OS Automatic Restart Management (ARM)

Automatic Restart Management (ARM) ist eine z/OS-Wiederherstellungsfunktion, mit der die Verfügbarkeit bestimmter Stapeljobs oder gestarteter Tasks (z. B. Subsysteme) verbessert werden kann. Es kann daher zu einer schnelleren Wiederaufnahme der produktiven Arbeit führen.

Um ARM verwenden zu können, müssen Sie Ihre Warteschlangenmanager und Kanalinitiatoren in einer bestimmten Weise einrichten, damit sie automatisch erneut gestartet werden. Weitere Informationen finden Sie im Abschnitt z/OS Automatic Restart Manager (ARM) verwenden.

**Zugehörige Konzepte**

„IBM MQ for z/OS einrichten” auf Seite 972
Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

# z/OS IBM MQ -Objekte unter z/OS definieren

Verwenden Sie unter z/OSeine der Eingabemethoden des Befehls IBM MQ , um IBM MQ -Objekte zu definieren.

Weitere Informationen zum Definieren von Objekten finden Sie unter „Kanäle in z/OS überwachen und steuern” auf Seite 1055.

## Übertragungswarteschlangen und Auslöserkanäle

Definieren Sie Folgendes:

- Eine lokale Warteschlange mit der Verwendung von XMITQ für jeden sendenden Nachrichtenkanal.
- Definitionen ferner Warteschlangen.

  Ein fernes Warteschlangenobjekt hat drei unterschiedliche Verwendungszwecke, je nachdem, wie der Name und der Inhalt angegeben werden:

  – Definition der fernen Warteschlange
  – WS-Manager-Aliasdefinition
  – Aliasdefinition der Warteschlange für Antwortwarteschlange

  Diese drei Methoden werden im Abschnitt Drei Methoden zur Verwendung des Definitionsobjekts für ferne Warteschlangen angezeigt.

Verwenden Sie das Feld TRIGDATA in der Übertragungswarteschlange, um den angegebenen Kanal auszulösen. For example:

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)
```

```
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

Das mitgelieferte Beispiel 'CSQ4INYD' enthält zusätzliche Beispiele für die erforderlichen Definitionen.

z/OS Der Verlust der Konnektivität zur CF-Struktur, bei der die Synchronisationswarteschlange für gemeinsam genutzte Kanäle definiert ist, oder ähnliche Probleme, kann vorübergehend verhindern, dass ein Kanal gestartet wird. Wenn Sie nach der Fehlerbehebung einen Auslösertyp von FIRST verwenden und der Kanal nicht gestartet werden kann, wenn er ausgelöst wird, müssen Sie den Kanal manuell starten. Wenn Sie die ausgelösten Kanäle nach der Fehlerbehebung automatisch starten möchten, sollten Sie das Attribut TRIGINT des Warteschlangenmanagers auf einen anderen Wert als den Standardwert setzen. Wenn Sie das Attribut TRIGINT auf einen anderen Wert als den Standardwert setzen, versucht der Kanalinitiator, den Kanal in regelmäßigen Abständen erneut zu starten, während Nachrichten in der Übertragungswarteschlange vorhanden sind.

### Synchronisationswarteschlange

DQM benötigt eine Warteschlange für die Verwendung mit Folgenummern und logischen Arbeitseinheiten (LUWID). Sie müssen sicherstellen, dass eine Warteschlange mit dem Namen SYSTEM.CHANNEL.SYNCQ (siehe Planung für z/OS ). Diese Warteschlange muss verfügbar sein, andernfalls kann der Kanalinitiator nicht gestartet werden.

Stellen Sie sicher, dass diese Warteschlange mit INDXTYPE (MSGID) definiert wird. Dieses Attribut verbessert die Geschwindigkeit, mit der auf sie zugegriffen werden kann.

### Kanalbefehlwarteschlangen

Sie müssen sicherstellen, dass eine Kanalbefehlswarteschlange für Ihr System mit dem Namen SYSTEM.CHANNEL.INITQ. vorhanden ist.

Wenn der Kanalinitiator ein Problem mit der SYSTEM.CHANNEL.INITQ feststellt, kann es nicht normal fortgesetzt werden, bis das Problem behoben ist. Das Problem könnte eine der folgenden sein:

- Die Warteschlange ist voll.
- Die Warteschlange ist für das put nicht aktiviert.
- Die Seitengruppe, auf der sich die Warteschlange befindet, ist voll.
- Der Kanalinitiator verfügt nicht über die richtige Sicherheitsberechtigung für die Warteschlange.

Wenn die Definition der Warteschlange während der Ausführung des Kanalinitiators in GET (DISABLED) geändert wird, kann der Initiator keine Nachrichten aus der Warteschlange abrufen und wird beendet.

### Kanalinitiator starten

Die Triggerung wird mit Hilfe des Kanalinitiators implementiert. Unter IBM MQ for z/OS wird der Initiator mit dem MQSC-Befehl `START CHINIT` gestartet.

### Kanalinitiator stoppen

Der Kanalinitiator wird automatisch gestoppt, wenn Sie den WS-Manager stoppen. Wenn Sie den Kanalinitiator, aber nicht den Warteschlangenmanager stoppen müssen, verwenden Sie den MQSC-Befehl `STOP CHINIT`.

## z/OS Kanäle in z/OS überwachen und steuern

Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern.

Jeder WS-Manager von z/OS verfügt über ein DQM-Programm (den *Kanalinitiator*), um Verbindungen zu fernen Warteschlangenmanagern mit nativen z/OS-Funktionen zu steuern.

Die Implementierung dieser Anzeigen und Befehle unter z/OS ist in die Operationen und Steuerkonsolen sowie die MQSC-Befehle integriert. In der Organisation dieser beiden Gruppen von Anzeigen und Befehlen wird keine Differenzierung vorgenommen.

Sie können Befehle auch mit Hilfe von PCF-Befehlen (PCF = Programmable Command Format) eingeben. Weitere Informationen zur Verwendung dieser Befehle finden Sie im Abschnitt Verwaltungstasks automatisieren .

Die Informationen in diesem Abschnitt gelten in allen Fällen, in denen der Kanalinitiator für die verteilte Warteschlangensteuerung verwendet wird. Dies gilt unabhängig davon, ob Sie Gruppen mit gemeinsamer Warteschlange oder gruppeninterne Warteschlangensteuerung verwenden.

## Die DQM-Kanalsteuerfunktion

Eine Übersicht über das Managementmodell für verteilte Warteschlangen finden Sie unter „Senden und Empfangen von Nachrichten” auf Seite 231.

Die Kanalsteuerfunktion besteht aus Anzeigen, Befehlen und Programmen, zwei Synchronisationswarteschlangen, Kanalbefehlswarteschlangen und den Kanaldefinitionen. Dieser Abschnitt enthält eine kurze Beschreibung der Komponenten der Kanalsteuerfunktion.

- Die Kanaldefinitionen werden als Objekte in der Seitengruppe null oder in Db2 wie andere IBM MQ-Objekte in z/OS gespeichert.
- Sie verwenden die Operationen und Steuerkonsolen, MQSC-Befehle oder PCF-Befehle wie folgt:
  - Kanaldefinitionen erstellen, kopieren, anzeigen, ändern und löschen
  - Kanalinitiatoren und Empfangsprogramme starten und stoppen
  - Kanäle starten, stoppen und mit Ping überprüfen, Kanalfolgenummern zurücksetzen und unbestätigte Nachrichten auflösen, wenn Links nicht erneut aufgebaut werden können
  - Statusinformationen zu Kanälen anzeigen
  - Informationen zu DQM anzeigen

  Insbesondere können Sie die Initialisierungseingabedatei CSQINPX verwenden, um Ihre MQSC-Befehle auszugeben. Diese Gruppe kann jedes Mal verarbeitet werden, wenn Sie den Kanalinitiator starten. Weitere Informationen finden Sie in Initialisierungsbefehle .
- Es gibt zwei Warteschlangen (SYSTEM.CHANNEL.SYNCQ und SYSTEM.QSG.CHANNEL.SYNCQ), die für die Kanalwiedersynchronisation verwendet werden. Definieren Sie diese Warteschlangen aus Leistungsgründen mit INDXTYPE (MSGID).
- Die Kanalbefehlswarteschlange (SYSTEM.CHANNEL.INITQ) wird zum Speichern von Befehlen für Kanalinitiatoren, Kanäle und Empfangsprogramme verwendet.
- Das Kanalsteuerungsfunktionsprogramm wird in einem eigenen Adressraum ausgeführt, getrennt vom Warteschlangenmanager, und umfasst den Kanalinitiator, Empfangsprogramme, MCAs, Auslösemonitor und Befehlshandler.
- Informationen zu Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzten Kanälen finden Sie unter Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange.
- Informationen zur gruppeninternen Warteschlangensteuerung finden Sie unter Gruppeninterne Warteschlangensteuerung.

## Kanäle unter z/OS verwalten

Verwenden Sie die Links in der folgenden Tabelle, um Informationen zur Verwaltung Ihrer Kanäle, Kanalinitiatoren und Empfangsprogramme zu erhalten:

*Tabelle 64. Kanaltasks*

| **Task, die ausgeführt werden soll** | **MQSC-Befehl** |
|---|---|
| Kanal definieren | DEFINE CHANNEL |

| *Tabelle 64. Kanaltasks (Forts.)* | |
|---|---|
| **Task, die ausgeführt werden soll** | **MQSC-Befehl** |
| Kanaldefinition ändern | ALTER CHANNEL |
| Kanaldefinition anzeigen | ANZEIGEN CHANNEL |
| Kanaldefinition löschen | DELETE CHANNEL |
| Kanalinitiator starten | START CHINIT |
| Kanalinitiator stoppen | STOP CHINIT |
| Informationen zum Kanalinitiator anzeigen | ANZEIGEN CHINIT |
| Kanallistener starten | START LISTENER |
| Kanallistener stoppen | STOP LISTENER |
| Kanal starten | START CHANNEL |
| Kanal testen | Pingkanal |
| Nachrichtenfolgenummern für einen Kanal zurücksetzen | Kanal zurücksetzen |
| Unbestätigte Nachrichten in einem Kanal auflösen | Auflösungskanal |
| Kanal stoppen | STOP CHANNEL |
| Kanalstatus anzeigen | ANZEIGEN CHSTATUS |
| Clusterkanäle anzeigen | DISPLAY CLUSQMGR |

**Zugehörige Konzepte**

„Verwenden der Anzeigen und der Befehle” auf Seite 1058
Sie können die MQSC-Befehle, die PCF-Befehle oder die Operationen und Steuerkonsolen verwenden, um DQM zu verwalten.

„IBM MQ for z/OS einrichten” auf Seite 972
Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

„Kommunikation für z/OS konfigurieren” auf Seite 1073
Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Um erfolgreich zu sein, ist es erforderlich, dass die Verbindung definiert und verfügbar ist. In diesem Abschnitt wird erläutert, wie eine Verbindung definiert wird.

„IBM MQ for z/OS für DQM mit Gruppen mit gemeinsamer Warteschlange vorbereiten” auf Seite 1078
Verwenden Sie die Anweisungen in diesem Abschnitt, um die verteilte Steuerung mit Warteschlangen für Gruppen mit gemeinsamer Warteschlange unter IBM MQ for z/OS zu konfigurieren.

„Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten” auf Seite 1083
Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dieser Versuch erfolgreich ist, ist es erforderlich, dass die Verbindung definiert und verfügbar ist.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten” auf Seite 1052

In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

### z/OS *Verwenden der Anzeigen und der Befehle*

Sie können die MQSC-Befehle, die PCF-Befehle oder die Operationen und Steuerkonsolen verwenden, um DQM zu verwalten.

Informationen zu MQSC-Befehlen finden Sie unter IBM MQ mit MQSC-Befehlen verwalten. Informationen zu PCF-Befehlen finden Sie im Abschnitt Automatische Verwaltung mit Hilfe von Befehlen des Befehls 'Programmable Command Formats' .

## Eingangsanzeige verwenden

Eine Einführung in den Aufruf der Betriebs- und Steuerkonsolen, die Verwendung der Funktionstasten und das Abrufen von Hilfe finden Sie unter IBM MQ for z/OS verwalten.

**Anmerkung:** Um die Betriebs- und Steuerkonsolen verwenden zu können, müssen Sie über die korrekte Sicherheitsberechtigung verfügen. Weitere Informationen hierzu finden Sie im Abschnitt verwaltenIBM MQ for z/OS und in den untergeordneten Abschnitten. Abbildung 106 auf Seite 1058 zeigt die Anzeige, die angezeigt wird, wenn Sie eine Anzeigensitzung starten. Der Text nach der Anzeige erläutert die Aktionen, die Sie in dieser Anzeige ausführen.

```
IBM MQ for z/OS - Main Menu

Complete fields. Then press Enter.

Action . . . . . . . . . . 1   0. List with filter  4. Manage
1. List or Display  5. Perform
2. Define like    6. Start
3. Alter       7. Stop
8. Command
Object type . . . . . . . . CHANNEL    +
Name . . . . . . . . . . . . *
Disposition . . . . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
S=Shared, A=All

Connect name . . . . . . . MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time . . . . 10  5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2024. All rights reserved.

Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Prompt  F9=SwapNext F10=Messages
F12=Cancel
```

*Abbildung 106. Die Eingangsanzeige der Operationen und Steuerelemente*

In dieser Anzeige können Sie folgende Schritte ausführen:

- Wählen Sie die Aktion aus, die Sie ausführen möchten, indem Sie die entsprechende Zahl in das Feld **Aktion** eingeben.
- Geben Sie den Objekttyp an, mit dem gearbeitet werden soll. Drücken Sie die Taste F4, um eine Liste der Objekttypen zu erhalten, wenn Sie sich nicht sicher sind, was sie sind.
- Zeigt eine Liste der Objekte des angegebenen Typs an. Geben Sie einen Stern (*) in das Feld **Name** ein, und drücken Sie die Eingabetaste, um eine Liste der Objekte (der angegebenen Art) anzuzeigen, die bereits auf diesem Subsystem definiert wurden. Anschließend können Sie ein oder mehrere Objekte auswählen, mit der in der Sequenz gearbeitet werden soll. Abbildung 107 auf Seite 1059 zeigt eine Liste der Kanäle, die auf diese Weise erzeugt wurden.

- Geben Sie im Feld **Disposition** die Disposition in der Gruppe mit gemeinsamer Warteschlange der Objekte an, mit denen Sie arbeiten möchten. Die Disposition bestimmt, wo das Objekt aufbewahrt wird und wie sich das Objekt verhält.
- Wählen Sie im Feld **Connect name** (Verbindungsname) den lokalen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange aus, zu dem bzw. zu der eine Verbindung hergestellt werden soll. Wenn die Befehle auf einem fernen Warteschlangenmanager ausgegeben werden sollen, wählen Sie das Feld **Target queue manager** (Zielwarteschlangenmanager) oder das Feld **Action queue manager** (Aktionswarteschlangenmanager) aus, abhängig davon, ob es sich beim fernen Warteschlangenmanager um ein Mitglied einer Gruppe mit gemeinsamer Warteschlange handelt. Wenn der ferne Warteschlangenmanager kein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, wählen Sie das Feld **Target queue manager** aus. Wenn der ferne Warteschlangenmanager Mitglied in einer Gruppe mit gemeinsamer Warteschlange ist, wählen Sie das Feld **Action queue manager** aus.
- Wählen Sie die Wartezeit für Antworten aus, die im Feld **Antwortzeitdauer** empfangen werden sollen.

```
List Channels - MQ25          Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.
1=Display  2=Define like  3=Alter  4=Manage  5=Perform
6=Start   7=Stop

Name         Type     Disposition  Status
<> *           CHANNEL    ALL   MQ25
_  SYSTEM.DEF.CLNTCONN  CLNTCONN   QMGR  MQ25
_  SYSTEM.DEF.CLUSRCVR  CLUSRCVR   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.CLUSSDR  CLUSSDR    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.RECEIVER  RECEIVER   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.REQUESTER REQUESTER   QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SENDER   SENDER    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SERVER   SERVER    QMGR  MQ25 INACTIVE
_  SYSTEM.DEF.SVRCONN  SVRCONN    QMGR  MQ25 INACTIVE
******** End of list ********



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Status  F12=Cancel
```

*Abbildung 107. Kanäle auflisten*

### z/OS *Kanal unter z/OS definieren*

Unter z/OSkönnen Sie einen Kanal mithilfe von MQSC-Befehlen oder über die Operationen und Steuerkonsolen definieren.

## Prozedur

- Verwenden Sie den Befehl **DEFINE CHANNEL** , um einen Kanal mit den MQSC-Befehlen zu definieren.
- Um die Betriebs-und Steuerkonsolen zu verwenden, füllen Sie ab der Eingangsanzeige die folgenden Felder aus und drücken Sie die Eingabetaste:

*Tabelle 65. Betriebs-und Steuerkonsolen: Felder der Eingangsanzeige*

| **Feld** | **Wert für Eingabe im Feld** |
|---|---|
| Action | 2 (Definieren wie) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | |
| Disposition | Die Position des neuen Objekts. |

Es werden einige Anzeigen angezeigt, in denen Sie Informationen zu dem Namen und den Attributen angeben können, die für den zu definierenden Kanal verwendet werden sollen. Sie werden mit den Standardattributwerten initialisiert. Die gewünschten Änderungen vornehmen, bevor die Eingabetaste gedrückt wird.

**Anmerkung:** Wenn Sie CHANNEL im Feld **Objekttyp** eingegeben haben, wird zuerst die Anzeige **Gültigen Kanaltyp auswählen** angezeigt.

Wenn Sie einen Kanal mit denselben Attributen wie ein vorhandener Kanal definieren möchten, geben Sie den Namen des Kanals, den Sie kopieren möchten, in das Feld **Name** in der Eingangsanzeige ein. Die Anzeigen werden mit den Attributen des vorhandenen Objekts initialisiert.

Informationen zu den Kanalattributen finden Sie unter Kanalattribute .

**Anmerkung:**

1. Nennen Sie alle Kanäle in Ihrem Netzwerk eindeutig. Wie in Netzdiagramm mit allen Kanälen dargestellt, ist die Angabe der Quellen-und Zielwarteschlangenmanagernamen im Kanalnamen eine gute Möglichkeit, diese Benennung zu tun.

## Nächste Schritte

Nachdem Sie Ihren Kanal definiert haben, müssen Sie Ihren Kanal schützen. Weitere Informationen finden Sie unter „Kanal schützen” auf Seite 1062.

### z/OS *Ändern einer Kanaldefinition*

Sie können eine Kanaldefinition mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen ändern.

Um eine Kanaldefinition mit den MQSC-Befehlen zu ändern, verwenden Sie ALTER CHANNEL.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 3 (Ändern) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | CHANNEL.TO.ALTER |
| Disposition | Die Position des gespeicherten Objekts. |

Sie werden mit einigen Anzeigen angezeigt, die Informationen zu den aktuellen Attributen des Kanals enthalten. Ändern Sie alle ungeschützten Felder, die Sie mit dem neuen Wert eingeben wollen, und drücken Sie die Eingabetaste, um die Kanaldefinition zu ändern.

Informationen zu den Kanalattributen finden Sie unter Kanalattribute .

### z/OS *Kanaldefinition anzeigen*

Sie können eine Kanaldefinition mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen anzeigen.

Verwenden Sie DISPLAY CHANNEL, um eine Kanaldefinition mit den MQSC-Befehlen anzuzeigen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 1 (Liste oder Anzeige) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | CHANNEL.TO.DISPLAY |

| Feld | Wert |
|---|---|
| Disposition | Die Position des Objekts. |

Sie werden mit einigen Anzeigen aufgerufen, in denen Informationen zu den aktuellen Attributen des Kanals angezeigt werden.

Informationen zu den Kanalattributen finden Sie unter Kanalattribute .

### z/OS *Kanaldefinition löschen*

Sie können eine Kanaldefinition mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen löschen.

Verwenden Sie DELETE CHANNEL, um eine Kanaldefinition mit den MQSC-Befehlen zu löschen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 4 (Verwalten) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | CHANNEL.TO.DELETE |
| Disposition | Die Position des Objekts. |

Sie werden mit einer anderen Anzeige dargestellt. Wählen Sie in dieser Anzeige Funktionstyp 1 aus.

Drücken Sie die Eingabetaste, um die Kanaldefinition zu löschen. Sie werden aufgefordert, zu bestätigen, dass Sie die Kanaldefinition löschen möchten, indem Sie erneut die Eingabetaste drücken.

**Anmerkung:** Der Kanalinitiator muss aktiv sein, bevor eine Kanaldefinition gelöscht werden kann (außer bei Clientverbindungskanälen).

### z/OS *Informationen zum Kanalinitiator anzeigen*

Sie können Informationen über den Kanalinitiator mit Hilfe von MQSC-Befehlen oder über die Operationen und Steuerkonsolen anzeigen.

Verwenden Sie DISPLAY CHINIT, um Informationen über den Kanalinitiator mit Hilfe der MQSC-Befehle anzuzeigen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 1 (Anzeigen) |
| Objekttyp | SYSTEM |
| Name | Leer |

Sie werden mit einer anderen Anzeige dargestellt. Wählen Sie in dieser Anzeige Funktionstyp 1 aus.

**Anmerkung:**

1. Das Anzeigen von Informationen zur verteilten Steuerung von Warteschlangen kann einige Zeit in Anspruch nehmen, wenn Sie über viele Kanäle verfügen.
2. Der Kanalinitiator muss aktiv sein, bevor Informationen zum verteilten Warteschlangensteuerung angezeigt werden können.

### z/OS *Kanal schützen*

Sie können einen Kanal mit Hilfe von MQSC-Befehlen oder über die Operationen und Steuerkonsolen sichern.

Verwenden Sie SET CHLAUTH , um einen Kanal mit den MQSC-Befehlen zu sichern.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 8 |

Sie werden mit einem Editor dargestellt, in dem Sie einen MQSC-Befehl angeben können, in diesem Fall ein Befehl CHLAUTH, siehe Abbildung 108 auf Seite 1062. Wenn Sie die Eingabe im Befehl abgeschlossen haben, sind die Pluszeichen (+) erforderlich. Geben Sie PF3 ein, um den Editor zu verlassen und den Befehl an den Befehlsserver zu übergeben.

```
****************************** Top of Data ******************************
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
****** **************************** Bottom of Data ****************************

Command ===>                          Scroll ===> PAGE
F1=Help   F3=Exit   F4=LineEdit F12=Cancel
```

*Abbildung 108. Befehlseingabe*

Die Ausgabe des Befehls wird dann an Sie übergeben, siehe Abbildung 109 auf Seite 1062.

```
****** **************************** Top of Data ******************************
000001 CSQU000I CSQUTIL IBM MQ for z/OS V7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMSCA ' SET CHLAUTH' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ===>                          Scroll ===> PAGE
F1=Help   F3=Exit   F5=Rfind   F6=Rchange  F9=SwapNext F12=Cancel
```

*Abbildung 109. Befehlsausgabe*

### z/OS *Kanalinitiator starten*

Sie können einen Kanalinitiator mit Hilfe von MQSC-Befehlen oder mit den Operations-und Steuerkonsolen starten.

Verwenden Sie START CHINIT, um einen Kanalinitiator unter Verwendung der MQSC-Befehle zu starten.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 6 (Start) |

| Feld | Wert |
|---|---|
| Objekttyp | SYSTEM |
| Name | Leer |

Die Anzeige "Systemfunktion starten" wird aufgerufen. Der Text im Anschluss an die folgende Anzeige erläutert, welche Aktion zu ergreifen ist:

```
Start a System Function

Select function type, complete fields, then press Enter to start system
function.

Function type . . . . . _   1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
JCL substitution . . . . . ________________________________________________
________________________________________________

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP
LU name (LU6.2) . . . . . ________________
Port number (TCP/IP) . . . 1414
IP address (TCP/IP) . . . ________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

*Abbildung 110. Systemfunktion starten*

Wählen Sie Funktionsart 1 (Kanalinitiator) aus, und drücken Sie die Eingabetaste.

### z/OS *Kanalinitiator stoppen*

Sie können einen Kanalinitiator mit Hilfe von MQSC-Befehlen oder mit den Operations-und Steuerkonsolen stoppen.

Verwenden Sie STOP CHINIT, um einen Kanalinitiator unter Verwendung der MQSC-Befehle zu stoppen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 7 (Stoppen) |
| Objekttyp | SYSTEM |
| Name | Leer |

Die Anzeige "Systemfunktion stoppen" wird angezeigt. Der Text im Anschluss an die Anzeige erläutert, wie Sie diese Anzeige verwenden:

```
Stop a System Function

Select function type, complete fields, then press Enter to stop system
function.

Function type . . . . . _   1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
Restart shared channels  Y Y=Yes, N=No

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP

Port number (TCP/IP) . . . _____
IP address (TCP/IP) . . . ________________________________________________



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

*Abbildung 111. Funktionssteuerung stoppen*

Wählen Sie Funktionsart 1 (Kanalinitiator) aus, und drücken Sie die Eingabetaste.

Der Kanalinitiator wartet darauf, dass alle aktiven Kanäle im Quiescemodus gestoppt werden, bevor er stoppt.

**Anmerkung:** Wenn einige der Kanäle Empfänger-oder Requesterkanäle sind, die aktiv sind, aber nicht aktiv sind, bewirkt eine Stoppanforderung, die entweder an den Empfänger oder an den Sender-Kanalinitiator abgesetzt wird, die sofortige Stopp-Anforderung.

Wenn Nachrichten jedoch fließen, wartet der Kanalinitiator darauf, dass der aktuelle Stapel von Nachrichten beendet wird, bevor er stoppt.

### z/OS *Kanallistener starten*

Sie können einen Kanal-Listener mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen starten.

Verwenden Sie START LISTENER, um einen Kanal-Listener mit den MQSC-Befehlen zu starten.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| **Feld** | **Wert** |
|---|---|
| Action | 6 (Start) |
| Objekttyp | SYSTEM |
| Name | Leer |

Die Anzeige "Systemfunktion starten" wird angezeigt (siehe Abbildung 110 auf Seite 1063 ).

Wählen Sie den Funktionstyp 2 (Kanallistener) aus. Wählen Sie Eingehende Disposition aus. Wählen Sie Transporttyp aus. Wenn der Transporttyp L ist, wählen Sie den LU-Namen aus. Wenn der Transporttyp T ist, wählen Sie Portnummer und (optional) IP-Adresse aus. Drücken Sie die Eingabetaste.

**Anmerkung:** Für den TCP/IP-Listener können Sie mehrere Kombinationen von Port und IP-Adresse starten.

## z/OS *Kanallistener stoppen*

Sie können einen Kanallistener mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen stoppen.

Verwenden Sie STOP LISTENER, um einen Kanal-Listener mit den MQSC-Befehlen zu stoppen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 7 (Stoppen) |
| Objekttyp | SYSTEM |
| Name | Leer |

Die Anzeige "Systemfunktion stoppen" wird angezeigt (siehe Abbildung 111 auf Seite 1064 ).

Wählen Sie den Funktionstyp 2 (Kanallistener) aus. Wählen Sie Eingehende Disposition aus. Wählen Sie Transporttyp aus. Wenn der Transporttyp 'T' ist, wählen Sie Portnummer und (optional) IP-Adresse aus. Drücken Sie die Eingabetaste.

**Anmerkung:** Für ein TCP/IP-Empfangsprogramm können Sie bestimmte Kombinationen von Port und IP-Adresse stoppen, oder Sie können alle Kombinationen stoppen.

## z/OS *Kanal starten*

Sie können einen Kanal mit Hilfe von MQSC-Befehlen oder mit den Operations-und Steuerkonsolen starten.

Verwenden Sie START CHANNEL, um einen Kanal mit den MQSC-Befehlen zu starten.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 6 (Start) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | CHANNEL.TO.USE |
| Disposition | Die Disposition des Objekts. |

Daraufhin wird die Anzeige "Kanal starten" aufgerufen. Im folgenden Text wird erläutert, wie die Anzeige verwendet wird:

```
Start a Channel

Select disposition, then press Enter to start channel.


Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager









Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

*Abbildung 112. Kanal starten*

Wählen Sie die Disposition der Kanalinstanz aus und geben Sie an, auf welchem WS-Manager sie gestartet werden soll.

Drücken Sie die Eingabetaste, um den Kanal zu starten.

### z/OS *Gemeinsam genutzten Kanal starten*

Um einen gemeinsam genutzten Kanal zu starten und ihn auf einem benannten Kanalinitiator zu halten, verwenden Sie Disposition = S (im Befehl START CHANNEL, geben Sie CHLDISP (FIXSHARED)) an.

Es kann immer nur eine Instanz des gemeinsam genutzten Kanals aktiv sein. Versuche, eine zweite Instanz des Kanals zu starten, schlagen fehl.

Wenn Sie einen Kanal auf diese Weise starten, gelten die folgenden Regeln für diesen Kanal:

- Sie können den Kanal von jedem Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange stoppen. Sie können sie auch dann stoppen, wenn der Kanalinitiator, auf dem der Kanal gestartet wurde, nicht aktiv ist, wenn Sie die Anforderung zum Stoppen des Kanals absetzen. Wenn der Kanal gestoppt wurde, können Sie ihn erneut starten, indem Sie Disposition = S (CHLDISP (FIXSHARED)) auf dem gleichen oder einem anderen Kanalinitiator angeben. Sie können sie auch starten, indem Sie Disposition = A (CHLDISP (SHARED)) angeben.
- Wenn sich der Kanal im Status "Starten" oder "Wiederholung" befindet, können Sie ihn erneut starten, indem Sie Disposition = S (CHLDISP (FIXSHARED)) auf demselben oder einem anderen Kanalinitiator angeben. Sie können sie auch starten, indem Sie Disposition = A (CHLDISP (SHARED)) angeben.
- Der Kanal kann ausgelöst werden, wenn er gestartet wird, wenn er in den inaktiven Status wechselt. Gemeinsam genutzte Kanäle, die ausgelöst werden, haben immer eine gemeinsame Disposition (CHLDISP (SHARED)).
- Der Kanal kann mit CHLDISP (FIXSHARED), auf jedem Kanalinitiator gestartet werden, wenn er in den inaktiven Status wechselt. Sie können sie auch starten, indem Sie Disposition = A (CHLDISP (SHARED)) angeben.
- Der Kanal wird von keinem anderen aktiven Kanalinitiator in der Gruppe mit gemeinsamer Warteschlange wiederhergestellt, wenn der Kanalinitiator, auf dem er gestartet wurde, mit SHARED (RESTART) gestoppt wird, oder wenn der Kanalinitiator abnormal beendet wird. Der Kanal wird nur wiederhergestellt, wenn der Kanalinitiator, auf dem er gestartet wurde, nächsten erneut gestartet wird. Dadurch werden fehlgeschlagene Versuche zur Kanalwiederherstellung gestoppt, die an andere Kanalinitiatoren in der Gruppe mit gemeinsamer Warteschlange übergeben werden, die zu ihrer Auslastung beitragen würden.

### z/OS *Kanal testen*

Sie können einen Kanal mit Hilfe von MQSC-Befehlen oder mit den Operations-und Steuerkonsolen testen.

Verwenden Sie PING CHANNEL, um einen Kanal mit den MQSC-Befehlen zu testen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| **Feld** | **Wert** |
|---|---|
| Action | 5 (Ausführen) |
| Objekttyp | SENDER, SERVER oder CHANNEL |
| Name | CHANNEL.TO.USE |
| Disposition | Die Disposition des Kanalobjekts. |

Die Anzeige 'Kanalfunktion ausführen' wird angezeigt. Im folgenden Text wird erläutert, wie die Anzeige verwendet wird:

```
Perform a Channel Function

Select function type, complete fields, then press Enter.


Function type . . . . . _   1. Reset  3. Resolve with commit
2. Ping  4. Resolve with backout

Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Sequence number for reset . . 1     1 - 999999999
Data length for ping . . . . 16   16 - 32768



Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

*Abbildung 113. Kanal testen*

Wählen Sie den Funktionstyp 2 (Pingsignal) aus.

Wählen Sie die Disposition des Kanals aus, für den der Test ausgeführt werden soll, und auf welchem WS-Manager getestet werden soll.

Die Datenlänge wird anfänglich auf 16 gesetzt. Ändern Sie ihn, wenn Sie möchten, und drücken Sie die Eingabetaste.

### z/OS *Nachrichtenfolgenummern für einen Kanal zurücksetzen*

Sie können Nachrichtenfolgenummern für einen Kanal mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen zurücksetzen.

Verwenden Sie RESET CHANNEL, um Kanalfolgenummern mit den MQSC-Befehlen zurückzusetzen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 5 (Ausführen) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | CHANNEL.TO.USE |
| Disposition | Die Disposition des Kanalobjekts. |

Die Anzeige "Kanalfunktion ausführen" wird angezeigt (siehe Abbildung 113 auf Seite 1067 ).

Wählen Sie Funktionstyp 1 (Zurücksetzen) aus.

Wählen Sie die Disposition des Kanals aus, für den die Zurücksetzung durchgeführt werden soll, und auf welchem WS-Manager sie ausgeführt werden soll.

Das Feld **Folgenummer** wird zunächst auf eins gesetzt. Ändern Sie diesen Wert, wenn Sie möchten, und drücken Sie die Eingabetaste.

### z/OS *Unbestätigte Nachrichten in einem Kanal auflösen*

Sie können unbestätigte Nachrichten in einem Kanal mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen auflösen.

Verwenden Sie RESOLVE CHANNEL, um unbestätigte Nachrichten in einem Kanal mit den MQSC-Befehlen aufzulösen.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 5 (Ausführen) |
| Objekttyp | SENDER, SERVER oder CHANNEL |
| Name | CHANNEL.TO.USE |
| Disposition | Die Disposition des Objekts. |

Die Anzeige "Kanalfunktion ausführen" wird angezeigt (siehe Abbildung 113 auf Seite 1067 ).

Wählen Sie Funktionstyp 3 oder 4 aus (Auflösung mit Commit oder Backout). (Weitere Informationen finden Sie in „Handhabung unbestätigter Kanäle” auf Seite 252.)

Wählen Sie die Disposition des Kanals aus, für den die Auflösung ausgeführt werden soll, und den Warteschlangenmanager, auf dem der Kanal ausgeführt werden soll. Drücken Sie die Eingabetaste.

### z/OS *-Kanal stoppen*

Sie können einen Kanal mit Hilfe von MQSC-Befehlen oder mit den Bedienungs-und Steuerkonsolen stoppen.

Wenn Sie einen Kanal mit den MQSC-Befehlen stoppen möchten, verwenden Sie STOP CHANNEL.

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| Feld | Wert |
|---|---|
| Action | 7 (Stoppen) |
| Objekttyp | Kanaltyp (z. B. SENDER) oder CHANNEL |
| Name | CHANNEL.TO.USE |
| Disposition | Die Disposition des Objekts. |

Daraufhin wird die Anzeige "Kanal stoppen" angezeigt. Im folgenden Text wird erläutert, wie die Anzeige verwendet wird:

```
Stop a Channel

Complete fields, then press Enter to stop channel.


Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE


Disposition . . . . . P   P=Private on MQ25
A=Shared on any queue manager

Stop mode . . . . . 1   1. Quiesce  2. Force
Stop status . . . . . 1   1. Stopped  2. Inactive

Queue manager . . . . . _________________________________________________
Connection name . . . . . _________________________________________________




Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

*Abbildung 114. -Kanal stoppen*

Wählen Sie die Disposition des Kanals aus, für den der Stopp ausgeführt werden soll, und auf welchem Warteschlangenmanager er gestoppt werden soll.

Wählen Sie den erforderlichen Stoppmodus aus:

**Quiesce**
Der Kanal wird gestoppt, wenn die aktuelle Nachricht abgeschlossen ist und der Stapel dann beendet wird, selbst wenn der Wert für die Stapelgröße nicht erreicht wurde und es bereits Nachrichten in der Übertragungswarteschlange gibt. Es werden keine neuen Stapel gestartet. Dieser Modus ist der Standardwert.

**Erzwingen**
Der Kanal wird sofort gestoppt. Wenn ein Stapel von Nachrichten in Bearbeitung ist, kann eine 'unbestätigte' Situation entstehen.

Wählen Sie den WS-Manager und den Verbindungsnamen für den Kanal aus, den Sie stoppen möchten.

Wählen Sie den Status aus, den Sie benötigen:

**Gestoppt**
Der Kanal wird nicht automatisch erneut gestartet und muss manuell erneut gestartet werden. Dieser Modus ist der Standardwert, wenn kein WS-Manager oder Verbindungsname angegeben ist. Wenn ein Name angegeben wird, ist dies nicht zulässig.

**Inaktiv**
Der Kanal wird bei Bedarf automatisch erneut gestartet. Dieser Modus ist der Standardwert, wenn ein WS-Manager oder Verbindungsname angegeben wird.

Drücken Sie die Eingabetaste, um den Kanal zu stoppen.

Weitere Informationen finden Sie unter „Kanäle stoppen und in den Quiescemodus versetzt” auf Seite 250. Informationen zum erneuten Starten von gestoppten Kanälen finden Sie in „Gestoppte Kanäle erneut starten” auf Seite 251.

**Anmerkung:** Wenn sich ein gemeinsamer Kanal in einem Wiederholungsstatus befindet und der Kanalinitiator, auf dem er gestartet wurde, nicht aktiv ist, wird eine Anforderung STOP für den Kanal auf dem Warteschlangenmanager abgesetzt, in dem der Befehl eingegeben wurde.

## z/OS *Kanalstatus anzeigen*

Sie können den Kanalstatus mit Hilfe von MQSC-Befehlen oder über die Operationen und Steuerkonsolen anzeigen.

Verwenden Sie DISPLAY CHSTATUS, um den Status eines Kanals oder einer Gruppe von Kanälen mit den MQSC-Befehlen anzuzeigen.

**Anmerkung:** Das Anzeigen von Kanalstatusinformationen kann einige Zeit in Anspruch nehmen, wenn Sie über viele Kanäle verfügen.

Unter Verwendung der Operationen und Steuerkonsolen in der Anzeige "Listenkanal" (siehe Abbildung 107 auf Seite 1059 ) wird für jeden Kanal eine Zusammenfassung des Kanalstatus wie folgt angezeigt:

| | |
|---|---|
| INACTIVE | Es sind keine Verbindungen aktiv. |
| *status* | Eine Verbindung ist aktiv |
| *nnn status* | Mehr als eine Verbindung ist aktuell und alle aktuellen Verbindungen haben denselben Status. |
| *nnn* CURRENT | Es ist mehr als eine Verbindung vorhanden und die aktuellen Verbindungen haben nicht alle denselben Status. |
| Leer | IBM MQ kann nicht feststellen, wie viele Verbindungen aktiv sind (z. B., weil der Kanalinitiator nicht aktiv ist)<br><br>**Anmerkung:** Für Kanalobjekte mit der Disposition GROUP wird kein Status angezeigt. |

Dabei steht *nnn* für die Anzahl der aktiven Verbindungen und *status* für eine der folgenden Verbindungen:

| | |
|---|---|
| INIT | INITIALISIERUNG |
| BIND | BINDING |
| ANFANG | STARTING |
| AUSFUEF | RUNNING |
| STOPP | STOPPING oder STOPPED |
| WIEDERHO | RETRYING |
| REQST | REQUESTING |

Wenn Sie weitere Informationen zum Kanalstatus anzeigen möchten, drücken Sie die Statustaste (F11) im Listenkanal oder in der Anzeige "Anzeigen" oder "Kanalanzeigen ändern", um die Anzeige "Listenkanäle-Aktueller Status" anzuzeigen (siehe Abbildung 115 auf Seite 1071 ).

```
List Channels - Current Status - MQ25      Row 1 of 16

Type action codes, then press Enter. Press F11 to display saved status.
1=Display current status

Channel name     Connection name                 State
Start time      Messages Last message time  Type   Disposition
<> *                           CHANNEL  ALL   MQ25

_ RMA0.CIRCUIT.ACL.F  RMA1                       STOP
_  2005-03-21 10.22.36 557735  2005-03-24 09.51.11 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.ACL.N  RMA1
_  2005-03-21 10.23.09 378675  2005-03-24 09.51.10 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.CL.F  RMA2
_  2005-03-24 01.12.51 45544   2005-03-24 09.51.08 SENDER  PRIVATE MQ25
_ RMA0.CIRCUIT.CL.N  RMA2
_  2005-03-24 01.13.55 45560   2005-03-24 09.51.11 SENDER  PRIVATE MQ25
_ RMA1.CIRCUIT.CL.F  RMA1
_  2005-03-21 10.24.12 360757  2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
_ RMA1.CIRCUIT.CL.N  RMA1
_  2005-03-21 10.23.40 302870  2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
******** End of list ********
Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Saved  F12=Cancel
```

*Abbildung 115. Kanalverbindungen auflisten*

Die Werte für den Status lauten wie folgt:

| | |
|---|---|
| INIT | INITIALISIERUNG |
| BIND | BINDING |
| ANFANG | STARTING |
| AUSFUEF | RUNNING |
| STOPP | STOPPING oder STOPPED |
| WIEDERHO | RETRYING |
| REQST | REQUESTING |
| DOUBT | STOPPED und INDOUBT (YES) |

Weitere Informationen finden Sie unter „Kanalstatus” auf Seite 242.

Sie können F11 drücken, um eine ähnliche Liste der Kanalverbindungen mit dem gespeicherten Status anzuzeigen. Drücken Sie die Taste F11, um zur aktuellen Liste zurückzukehren. Der gespeicherte Status gilt erst dann, wenn mindestens ein Nachrichtenstachsatz auf dem Kanal übertragen wurde.

Verwenden Sie Aktionscode 1 oder einen Schrägstrich (/), um eine Verbindung auszuwählen, und drücken Sie die Eingabetaste. Die Anzeigen "Display Channel Connection Current Status" werden angezeigt.

### z/OS *Clusterkanäle anzeigen*

Sie können Clusterkanäle mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen anzeigen.

Verwenden Sie den MQSC-Befehl DISPLAY CLUSQMGR, um alle definierten Clusterkanäle anzuzeigen (explizit oder mit automatischer Definition).

Verwenden Sie die Operationen und Steuerkonsolen, beginnend mit der Eingangsanzeige, füllen Sie die folgenden Felder aus, und drücken Sie die Eingabetaste:

| **Feld** | **Wert** |
|---|---|
| Action | 1 (Liste oder Anzeige) |

| Feld | Wert |
|---|---|
| Objekttyp | CLUSCHL |
| Name | * |

Sie werden mit einer Anzeige wie Abbildung Abbildung 116 auf Seite 1072 dargestellt, in der die Informationen für jeden Clusterkanal drei Zeilen belegen und deren Kanal-, Cluster- und Warteschlangenmanager-Namen enthalten. Für Clustersenderkanäle wird der Gesamtstatus angezeigt.

```
List Cluster queue manager Channels - MQ25    Row 1 of 9

Type action codes, then press Enter. Press F11 to display connection status.
1=Display  5=Perform  6=Start  7=Stop

Channel name     Connection name                State
Type     Cluster name                   Suspended
Cluster queue manager name            Disposition
<>   *                          -    MQ25
_ TO.MQ90.T      HURSLEY.MACH90.COM(1590)
_  CLUSRCVR   VJH01T                        N
_   MQ90                       -    MQ25
_ TO.MQ95.T      HURSLEY.MACH95.COM(1595)             RUN
_  CLUSSDRA   VJH01T                        N
_   MQ95                       -    MQ25
_ TO.MQ96.T      HURSLEY.MACH96.COM(1596)             RUN
_  CLUSSDRB   VJH01T                        N
_   MQ96                       -    MQ25
******** End of list ********


Command ===> ____________________________________________________________
F1=Help   F2=Split   F3=Exit   F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd    F9=SwapNext F10=Messages F11=Status  F12=Cancel
```

*Abbildung 116. Clusterkanäle auflisten*

Wenn Sie vollständige Informationen zu einem oder mehreren Kanälen anzeigen möchten, geben Sie den Aktionscode 1 für ihre Namen ein, und drücken Sie die Eingabetaste. Verwenden Sie die Aktionscodes 5, 6 oder 7, um Funktionen auszuführen (z. B. Ping, Auflösen und Zurücksetzen), und starten oder stoppen Sie einen Clusterkanal.

Um weitere Informationen zum Kanalstatus anzuzeigen, drücken Sie die Taste "Status" (F11).

## z/OS *IBM MQ for z/OS für die Verwendung der Funktion zEnterprise Data Compression Express vorbereiten*

Die Funktion zEnterprise Data Compression (zEDC) Express ist ab IBM zEC12 GA2für bestimmte Modelle von IBM Z -Maschinen mit mindestens z/OS Version z/OS 2.1verfügbar.

Weitere Informationen finden Sie unter zEnterprise Data Compression (zEDC) .

### Voraussetzungen

Für IBM z15 und höher wurde die Funktion zEnterprise Data Compression (zEDC) Express aus einem optionalen Feature im PCIe-E/A-Einschub des Hardwaresystems als integrierter Akzelerator für zEDCversetzt. Mit dieser Änderung werden die Konfigurationsvoraussetzungen aktualisiert und hängen von Ihrem Hardwaresystem ab.

#### IBM z15 oder höher

Wenden Sie eine der folgenden PTFs entsprechend Ihrer Version von z/OSan:

- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635

- z/OS 2.2: UJ00638
- z/OS 2.1: UJ00639

Es gibt keine Hardwarevoraussetzungen für Systeme mit z15 oder höher. Die Integrated Accelerator for zEDC -Lösung in diesen Systemen bietet integrierte Datenbeschleunigung, sodass ein separater Adapter nicht mehr benötigt wird.

**IBM zEC12 GA2 zu IBM z14**

Ihr System muss außerdem die folgenden Voraussetzungen erfüllen:

- Ein zEDC Express ® -Adapter, der in den PCIe-E/A-Einschüben des Hardwaresystems installiert ist.
- Die zEDC -Softwarefunktionalität (eine optionale, kostenpflichtige Funktion) muss in einem parmlib-Member IFAPRDxx aktiviert sein.

## Verfahren

**IBM zEC12 GA2 zu IBM z14**

Stellen Sie sicher, dass die Benutzer-ID des Kanalinitiators die Berechtigung READ für das Profil FPZ.ACCELERATOR.COMPRESSION in der RACF FACILITY CLASS oder das Äquivalent in dem externen Sicherheitsmanager (ESM) hat, den Ihr Unternehmen verwendet.

**Achtung:** Nicht erforderlich für IBM z15 oder höher.

**IBM zEnterprise zEC12 GA2 oder höher**

Konfigurieren Sie den Kanal mit der Option COMPMSG (ZLIBFAST) sowohl auf der sendenden als auch auf der empfangenden Seite. Nach der Konfiguration wird die zlib-Komprimierung verwendet, um Nachrichten zu komprimieren und zu dekomprimieren, die über den Kanal fließen.

Die Komprimierung wird im zEDC ausgeführt, wenn die Größe der zu komprimierenden Daten den Mindestschwellenwert überschreitet. Der Schwellenwert hängt von der verwendeten IBM z-Hardware ab.

- IBM zEC12 GA2 für IBM z14 hat einen Mindestschwellenwert von 4KB
- IBM z15 oder höher hat einen Mindestschwellenwert von 1KB

Bei Nachrichten unterhalb des Schwellenwerts erfolgt die Komprimierung oder Inflation in der Software.

# z/OS Kommunikation für z/OS konfigurieren

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Um erfolgreich zu sein, ist es erforderlich, dass die Verbindung definiert und verfügbar ist. In diesem Abschnitt wird erläutert, wie eine Verbindung definiert wird.

DQM ist eine ferne Warteschlangenfunktion für IBM MQ. Es stellt Kanalsteuerprogramme für den Warteschlangenmanager zur Verfügung, die die Schnittstelle zu Kommunikationsverbindungen bilden. Diese Links sind durch den Systembediener steuerbar. Die Kanaldefinitionen, die von der verteilten Steuerung der Warteschlangensteuerung gehalten werden, verwenden diese Verbindungen.

Wählen Sie eine der beiden Formen des Kommunikationsprotokolls, die für z/OS verwendet werden können:

- „TCP-Verbindung unter z/OS definieren” auf Seite 1074
- „Definieren einer LU6.2-Verbindung für z/OS mit APPC/MVS” auf Seite 1077

MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat

oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

Jede Kanaldefinition muss nur ein Protokoll als Attribut "Übertragungsprotokoll" (Transporttyp) angeben. Ein WS-Manager kann mehr als ein Protokoll für die Kommunikation verwenden.

Möglicherweise finden Sie es auch hilfreich, sich Beispielkonfiguration - IBM MQ for z/OS anzusehen. Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, lesen Sie den Abschnitt „Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten” auf Seite 1083.

**Zugehörige Konzepte**

„Verwenden der Anzeigen und der Befehle” auf Seite 1058
Sie können die MQSC-Befehle, die PCF-Befehle oder die Operationen und Steuerkonsolen verwenden, um DQM zu verwalten.

„IBM MQ for z/OS einrichten” auf Seite 972
Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

„Kanäle in z/OS überwachen und steuern” auf Seite 1055
Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern.

„IBM MQ for z/OS für DQM mit Gruppen mit gemeinsamer Warteschlange vorbereiten” auf Seite 1078
Verwenden Sie die Anweisungen in diesem Abschnitt, um die verteilte Steuerung mit Warteschlangen für Gruppen mit gemeinsamer Warteschlange unter IBM MQ for z/OS zu konfigurieren.

„Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten” auf Seite 1083
Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dieser Versuch erfolgreich ist, ist es erforderlich, dass die Verbindung definiert und verfügbar ist.

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren” auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten” auf Seite 1052
In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

## z/OS *TCP-Verbindung unter z/OS definieren*

Um eine TCP-Verbindung zu definieren, gibt es eine Reihe von Einstellungen für die Konfiguration.

Der Name des TCP-Adressraums muss in der Datei mit den TCP-Systemparametern angegeben werden, *tcpip* .TCPIP.DATA. In der Datei muss eine Anweisung " TCPIPJOBNAME *TCPIP_proc* " enthalten sein.

Wenn Sie eine Firewall verwenden, müssen Sie `allow` -Verbindungen vom Kanalinitiator zu den Adressen in den Kanälen und von den fernen Verbindungen in den Warteschlangenmanager konfigurieren.

Gewöhnlich konfiguriert die Definition für eine Firewall die sendende IP-Adresse und den Port für die Ziel-IP-Adresse und den Zielport:

- Ein z/OS-Image kann mehr als einen Hostnamen haben; evtl. müssen Sie die Firewall mit mehreren Hostadressen als Quellenadresse konfigurieren.

  Sie können den Befehl NETSTAT HOME verwenden, um diese Namen und Adressen anzuzeigen.
- Ein Kanalinitiator kann über mehrere Empfangsprogramme in verschiedenen Ports verfügen, so dass Sie diese Ports konfigurieren müssen.

- Wenn Sie einen gemeinsam genutzten Port für eine Gruppe mit gemeinsamer Warteschlange verwenden, müssen Sie den gemeinsam genutzten Port ebenfalls konfigurieren.

Der Adressraum des Kanalinitiators muss die Berechtigung zum Lesen der Datei haben. Die folgenden Verfahren können für den Zugriff auf die Datei TCPIP.DATA verwendet werden, je nachdem, welches TCP/IP-Produkt und welche Schnittstelle Sie verwenden:

- Umgebungsvariable, RESOLVER_CONFIG
- /etc/resolv.conf im Dateisystem
- // DD-Anweisung SYSTCPD
- // DD-Anweisung SYSTCPDD
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

Sie müssen auch vorsichtig sein, um das übergeordnete Qualifikationsmerkmal für TCP/IP korrekt anzugeben.

Sie benötigen einen entsprechend konfigurierten DNS-Server (DNS = Domain Name System), der in der Lage ist, die Umsetzung von Name zu IP-Adresse und die IP-Adresse in die Namensumsetzung zu übersetzen.

**Anmerkung:** Einige Änderungen an der Resolver-Konfiguration erfordern ein Stoppen und erneutes Starten von Anwendungen, die sie verwenden, z. B. IBM MQ.

Weitere Informationen finden Sie in den folgenden Informationen:

- TCP/IP-Basissystem
- z/OS UNIX System Services.

Jeder TCP-Kanal, der gestartet wird, verwendet TCP-Ressourcen. Möglicherweise müssen Sie die folgenden Parameter in der Konfigurationsdatei PROFILE.TCPIP anpassen:

**ACBPOOLSIZE**
Fügen Sie einen pro gestarteten TCP-Kanal hinzu, plus eins.

**CCBPOOLSIZE**
Fügen Sie einen pro gestarteten TCP-Kanal hinzu, plus einen pro DQM-Dispatcher plus eine

**DATABUFFERPOOLSIZE**
Fügen Sie zwei pro gestarteten TCP-Kanal hinzu, plus eins.

**MAXFILEPROC**
Steuert, wie viele Kanäle jeder Dispatcher in dem Kanalinitiator verarbeiten kann.

Dieser Parameter wird im Member BPXPRMxx von SYSI.PARMLIB angegeben. Stellen Sie sicher, dass Sie einen Wert angeben, der groß genug für Ihre Anforderungen ist.

Standardmäßig ist der Kanalinitiator nur in der Lage, an IP-Adressen zu binden, die dem im Warteschlangenmanager TCPNAME-Warteschlangenmanager angegebenen Stack zugeordnet sind. Um dem Kanalinitiator die Kommunikation mit zusätzlichen TCP/IP-Stacks auf dem System zu ermöglichen, ändern Sie das Attribut TCPSTACK queue manager in MULTIPLE.

Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird in Red Hat oder Ubuntu Linuxoder Windowsimplementiert. Weitere Informationen finden Sie unter Aspera gateway -Verbindung unter Linux oder Windowsdefinieren.

**Zugehörige Konzepte**

„Sendende Beendigung” auf Seite 1076

Am sendenden Ende der TCP/IP-Verbindung gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

„Empfang auf TCP" auf Seite 1076
Am empfangenden Ende der TCP/IP-Verbindung gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

„Verwenden der Backlog-Option für den TCP-Listener unter z/OS" auf Seite 1077
Beim Empfang über TCP/IP wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Diese ausstehenden Anforderungen können als *Rückstand* von Anforderungen betrachtet werden, die auf den TCP/IP-Port warten, bis der Listener die Anforderung akzeptiert hat.

z/OS *Sendende Beendigung*

Am sendenden Ende der TCP/IP-Verbindung gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

Das Feld für den Verbindungsnamen (CONNAME) in der Kanaldefinition muss entweder auf den Hostnamen (z. B. MVSHUR1) oder die TCP-Netzadresse des Ziels gesetzt sein. Die TCP-Netzadresse kann in Dezimalschreibweise mit Trennzeichen gemäß IPv4 (z. B. 127.0.0.1) oder im Hexadezimalformat nach IPv6 (z. B. 2001:DB8:0:0:0:0:0:0) angegeben werden. Wenn es sich bei dem Verbindungsnamen um einen Hostnamen handelt, ist ein TCP-Namensserver erforderlich, um den Hostnamen in eine TCP-Hostadresse zu konvertieren. (Diese Anforderung ist eine Funktion von TCP, nicht von IBM MQ.)

Am einleitenden Ende einer Verbindung (Sender-, Requester-und Serverkanaltypen) ist es möglich, eine optionale Portnummer für die Verbindung bereitzustellen, z. B.:

**Verbindungsname**
192.0.2.0 (1555)

In diesem Fall versucht das einleitende Ende, eine Verbindung zu einem empfangenden Programm herzustellen, das an Port 1555 empfangsbereit ist.

**Anmerkung:** Die Standardportnummer 1414 wird verwendet, wenn eine optionale Portnummer nicht angegeben ist.

Der Kanalinitiator kann einen beliebigen TCP/IP-Stack verwenden, der aktiv und verfügbar ist. Standardmäßig bindet der Kanalinitiator seine abgehenden Kanäle an die Standard-IP-Adresse für den TCP/IP-Stack, der im Attribut TCPNAME des Warteschlangenmanagers angegeben ist. Wenn Sie eine Verbindung über einen anderen Stack herstellen möchten, müssen Sie entweder den Hostnamen oder die IP-Adresse des Stacks im Attribut LOCLADDR des Kanals angeben.

z/OS *Empfang auf TCP*

Am empfangenden Ende der TCP/IP-Verbindung gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet. Dazu muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Sie starten dieses Listenerprogramm mit dem Befehl START LISTENER oder verwenden die Operationen und Steuerkonsolen.

Die Standardposition lautet wie folgt:

- Das TCP-Listener-Programm verwendet Port 1414 und ist an allen Adressen empfangsbereit, die für den TCP-Stack verfügbar sind.
- TCP/IP-Listener können nur an Adressen gebunden werden, die dem TCP/IP-Stack zugeordnet sind, der im Attribut TCPNAME queue manager angegeben ist.

Wenn Sie Empfangsprogramme für andere Adressen oder alle verfügbaren TCP-Stacks starten möchten, setzen Sie Ihr TCPSTACK-WS-Manager-Attribut auf 'MULTIPLE'.

Sie können das TCP-Empfangsprogramm starten, um nur an einer bestimmten Adresse oder einem bestimmten Hostnamen empfangsbereit zu sein, indem Sie IPADDR im Befehl START LISTENER angeben. Weitere Informationen finden Sie unter Listener.

z/OS *Verwenden der Backlog-Option für den TCP-Listener unter z/OS*

Beim Empfang über TCP/IP wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Diese ausstehenden Anforderungen können als *Rückstand* von Anforderungen betrachtet werden, die auf den TCP/IP-Port warten, bis der Listener die Anforderung akzeptiert hat.

Der Standardwert für den Listenerrückstand unter z/OS ist 10000. Wenn der Rückstand diese Werte erreicht, wird die TCP/IP-Verbindung zurückgewiesen, und der Kanal kann nicht gestartet werden.

Bei MCA-Kanälen führt dies dazu, dass der Kanal in einen RETRY-Status eingeht und die Verbindung zu einem späteren Zeitpunkt erneut versucht.

Für Clientverbindungen empfängt der Client einen Ursachencode MQRC_Q_MGR_NOT_AVAILABLE von MQCONN und kann die Verbindung zu einem späteren Zeitpunkt wiederholen.

**Zugehörige Konzepte**

„Verwenden der Backlog-Option für den TCP-Listener unter IBM MQ for Multiplatforms” auf Seite 286

In TCP werden die Verbindungen nur unvollständig behandelt, wenn zwischen dem Server und dem Client ein Dreiwege-Handshake nicht stattfindet. Diese Verbindungen werden als ausstehende Verbindungsanforderungen bezeichnet. Für diese ausstehenden Verbindungsanforderungen wird ein Maximalwert festgelegt und kann als Rückstand von Anforderungen betrachtet werden, die auf den TCP-Port warten, damit der Listener die Anforderung akzeptiert.

## z/OS *Definieren einer LU6.2-Verbindung für z/OS mit APPC/MVS*

Um eine LU6.2-Verbindung definieren zu können, müssen Sie eine Reihe von Einstellungen konfigurieren.

### APPC/MVS-Konfiguration

Jede Instanz des Kanalinitiators muss den Namen der LU haben, die für APPC/MVS definiert ist, im Member APPCPMxx von SYS1.PARMLIB, wie im folgenden Beispiel:

```
LUADD ACBNAME( luname ) NOSCHED TPDATA(CSQ.APPCTP)
```

*luname* ist der Name der logischen Einheit, die verwendet werden soll. NOSCHED ist erforderlich; TPDATA wird nicht verwendet. Es sind keine Hinzufügungen für das Member ASCHPMxx oder die Datei APPC/MVS TP-Profil erforderlich.

Das Seitendaten-Set muss erweitert werden, um die von DQM verwendeten Verbindungen zu definieren. Weitere Informationen zur Verwendung des APPC-Dienstprogramms ATBSDFMU enthält das mitgelieferte Beispiel CSQ4SIDE. Ausführliche Informationen zu den zu verwendenden TPNAME-Werten finden Sie in der folgenden Tabelle:

*Tabelle 66. Einstellungen auf dem lokalen z/OS-System für eine ferne Warteschlangenmanagerplattform*

| **Ferne Plattform** | **TPNAME** |
|---|---|
| z/OS oder MVS | Dasselbe gilt für TPNAME in den entsprechenden Nebeninformationen zum fernen Warteschlangenmanager. |
| IBM i | Entsprechendes gilt für den Vergleichswert im Routing-Eintrag auf dem IBM i-System. |
| Systeme mit AIX and Linux | Dasselbe gilt für TPNAME in den entsprechenden Nebeninformationen zum fernen Warteschlangenmanager. |
| Windows | Wie im Windows-Befehl 'Listener ausführen' angegeben, oder dem aufrufbaren Transaktionsprogramm, das mit TpSetup unter Windows definiert wurde. |

Wenn mehrere WS-Manager auf derselben Maschine vorhanden sind, stellen Sie sicher, dass die TPNamen in den Kanaldefinitionen eindeutig sind.

In einer Umgebung, in der der Warteschlangenmanager über APPC mit einem Warteschlangenmanager auf demselben oder einem anderen z/OS-System kommuniziert, stellen Sie sicher, dass entweder die VTAM-Definition für die kommunizierende LU SECACPT(ALREADYV) angibt oder dass ein RACF APPCLU-Profil für die Verbindung zwischen LUs vorhanden ist, das CONVSEC(ALREADYV) angibt.

Der z/OS-Befehl VARY ACTIVE muss für die Basis- und die Listener-LUs abgesetzt werden, bevor versucht wird, die eingehende oder abgehende Kommunikation zu starten.

**Achtung:** Zusätzlich zu der APPC-Konfiguration müssen Sie den folgenden Befehl eingeben:

```
ALTER QMGR LUNAME(luname)
```

und starten Sie den Kanalinitiator erneut.

Weitere Informationen finden Sie in LUNAME.

**Zugehörige Konzepte**

„Verbindung zu LU 6.2 wird hergestellt" auf Seite 1078
Um eine Verbindung zu LU 6.2 herzustellen, gibt es eine Reihe von Einstellungen für die Konfiguration.

„Empfangen auf LU 6.2" auf Seite 1078
Für den Empfang von LU 6.2 gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

z/OS *Verbindung zu LU 6.2 wird hergestellt*

Um eine Verbindung zu LU 6.2 herzustellen, gibt es eine Reihe von Einstellungen für die Konfiguration.

Das Feld für den Verbindungsnamen (CONNAME) in der Kanaldefinition muss auf den symbolischen Zielnamen gesetzt werden, wie in der Nebeninformationsdatei für APPC/MVS angegeben.

Der zu verwendende LU-Name (definiert in APPC/MVS wie zuvor beschrieben) muss auch in den Kanalinitiatorparametern angegeben werden. Sie muss auf dieselbe LU gesetzt werden, die vom Empfangsprogramm empfangen wird.

Der Kanalinitiator verwendet die Option " SECURITY (SAME) " APPC/MVS, daher ist es die Benutzer-ID des Kanalinitiatoradressraums, der für abgehende Übertragungen verwendet wird, und wird dem Empfänger angezeigt.

z/OS *Empfangen auf LU 6.2*

Für den Empfang von LU 6.2 gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

Empfangs-MCAs werden als Antwort auf eine Startanforderung vom sendenden Kanal gestartet. Dazu muss ein Empfangsprogramm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Das Listenerprogramm ist ein APPC/MVS-Server. Sie starten ihn mit dem Befehl START LISTENER oder verwenden die Operationen und Steuerkonsolen. Sie müssen den LU-Namen angeben, der mit einem symbolischen Zielnamen verwendet werden soll, der in der Seitendaten-Datei definiert ist. Die so identifizierte lokale LU muss mit der lokalen LU identisch sein, die für abgehende Übertragungen verwendet wird, wie in den Kanalinitiatorparametern festgelegt.

## z/OS IBM MQ for z/OS für DQM mit Gruppen mit gemeinsamer Warteschlange vorbereiten

Verwenden Sie die Anweisungen in diesem Abschnitt, um die verteilte Steuerung mit Warteschlangen für Gruppen mit gemeinsamer Warteschlange unter IBM MQ for z/OS zu konfigurieren.

Eine Beispielkonfiguration unter Verwendung von Gruppen mit gemeinsamer Warteschlange finden Sie im Abschnitt Beispielkonfiguration - IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange. Ein Beispiel für eine Nachrichtenkanalplanung unter Verwendung von Gruppen mit gemeinsamer Warteschlange finden Sie im Abschnitt Beispiel für Nachrichtenkanalplanung für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange.

Sie müssen die folgenden Komponenten erstellen und konfigurieren, um die verteilte Steuerung von Warteschlangen mit Gruppen mit gemeinsamer Warteschlange zu aktivieren:

- LU 6.2 und TCP/IP-Listener
- Übertragungswarteschlangen und Triggering
- Nachrichtenkanalagenten (MCAs)
- Synchronisationswarteschlange

Nachdem Sie die Komponenten erstellt haben, die Sie für die Konfiguration der Kommunikation benötigen, lesen Sie die Informationen im Abschnitt „Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten" auf Seite 1083.

Informationen zur Überwachung und Steuerung von Kanälen bei der Verwendung von Gruppen mit gemeinsamer Warteschlange finden Sie unter „Kanäle in z/OS überwachen und steuern" auf Seite 1055.

In den folgenden Abschnitten finden Sie Konzepte für die Gruppe mit gemeinsamer Warteschlange und die Vorteile bei deren Verwendung.

## Serviceklasse

Eine gemeinsam genutzte Warteschlange ist eine Art von lokaler Warteschlange, die eine andere Serviceklasse bietet. Nachrichten in einer gemeinsam genutzten Warteschlange werden in einer Coupling-Facility (CF) gespeichert, damit von allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange darauf zugegriffen werden kann. Eine Nachricht in einer gemeinsam genutzten Warteschlange muss eine Nachricht mit einer Länge von nicht mehr als 100 MB sein.

## Generische Schnittstelle

Eine Gruppe mit gemeinsamer Warteschlange verfügt über eine generische Schnittstelle, mit der das Netz die Gruppe als eine einzelne Entität anzeigen kann. Diese Sicht wird durch die Verwendung einer einzigen generischen Adresse erreicht, die verwendet werden kann, um eine Verbindung zu einem beliebigen Warteschlangenmanager in der Gruppe herzustellen.

Jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange ist für eingehende Sitzungsanforderungen an einer Adresse empfangsbereit, die logisch mit der generischen Adresse verknüpft ist. Weitere Informationen finden Sie unter „LU 6.2- und TCP/IP-Listener für Gruppen mit gemeinsamer Warteschlange" auf Seite 1081.

## Start des Lastausgleichkanals

Eine gemeinsam genutzte Übertragungswarteschlange kann von einem abgehenden Kanal bedient werden, der auf einem beliebigen Kanalinitiator in der Gruppe mit gemeinsamer Warteschlange ausgeführt wird. Der Lastausgleichkanalstart bestimmt, wo ein Startkanalbefehl zielgerichtet ist. Es wird ein geeigneter Kanalinitiator ausgewählt, der Zugriff auf das erforderliche DFV-Subsystem hat. Ein Kanal, der mit TRPTYPE (LU6.2) definiert ist, kann beispielsweise nicht auf einem Kanalinitiator gestartet werden, der nur Zugriff auf ein TCP/IP-Subsystem hat.

Die Wahl des Kanalinitiators hängt von der Kanalbelastung und dem Kopfraum des Kanalinitiators ab. Bei der Kanalauslastung handelt es sich um die Anzahl der aktiven Kanäle als Prozentsatz der maximal zulässigen Anzahl aktiver Kanäle, die in den Kanalinitiatorparametern definiert sind. Der Headroom ist die Differenz zwischen der Anzahl der aktiven Kanäle und der maximal zulässigen Anzahl.

Eingehende gemeinsam genutzte Kanäle können durch Verwendung einer generischen Adresse, wie in „LU 6.2- und TCP/IP-Listener für Gruppen mit gemeinsamer Warteschlange" auf Seite 1081 beschrieben, über die Gruppe mit gemeinsamer Warteschlange hinweg geladen werden.

## Wiederherstellung des gemeinsam genutzten Kanals

In der folgenden Tabelle werden die Typen des Fehlers im gemeinsam genutzten Kanal und die Art und Weise, wie die einzelnen Typen gehandhabt werden, angezeigt

| Typ des Fehlers: | Was passiert: |
|---|---|

| | |
|---|---|
| Fehler am DFV-Subsystem des Kanalinitiators | Die Kanäle, die vom Kommunikationssubsystem abhängig sind, wiederholen eine Kanaloperation und werden mit einem lastausgleichsbasierten Startbefehl auf einem geeigneten Kanalinitiator der Gruppe mit gemeinsamer Warteschlange erneut gestartet. |
| Kanalinitiatorfehler | Der Kanalinitiator schlägt fehl, aber der zugehörige Warteschlangenmanager bleibt aktiv. Der Warteschlangenmanager überwacht den Fehler und leitet die Wiederherstellungsverarbeitung ein. |
| Warteschlangenmanagerfehler | Der Warteschlangenmanager schlägt fehl (der zugeordnete Kanalinitiator wird nicht ausgeführt). Andere Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange überwachen das Ereignis und leiten eine Peerwiederherstellung ein. |
| Shared-Status-Fehler | Die Kanalstatusinformationen werden in Db2 gespeichert, sodass ein Verlust der Konnektivität zu Db2 einen Fehler verursacht, wenn eine Änderung des Kanalstatus auftritt. Aktive Kanäle können ohne Zugriff auf diese Ressourcen ausgeführt werden. Wenn der Zugriff auf Db2 fehlschlägt, dann wird der Kanal in den Wiederholungsmodus versetzt. |

Zur Wiederherstellung eines gemeinsamen Kanals für ein fehlgeschlagenes System wird Konnektivität zu Db2 auf dem System benötigt, auf dem die Wiederherstellung verwaltet wird, um den Status des gemeinsamen Kanals abzurufen.

## Clientkanäle

Clientverbindungskanäle können von der hohen Verfügbarkeit von Nachrichten in Gruppen mit gemeinsamer Warteschlange profitieren, die mit der generischen Schnittstelle und nicht mit einem bestimmten Warteschlangenmanager verbunden sind. Weitere Informationen finden Sie unter Clientverbindungskanäle.

**Zugehörige Konzepte**

Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange

„IBM MQ for z/OS einrichten" auf Seite 972
Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

„Cluster und Gruppen mit gemeinsamer Warteschlange" auf Seite 1083
Sie können die gemeinsam genutzte Warteschlange für einen Cluster in einer einzigen Definition verfügbar machen. Geben Sie dazu den Namen des Clusters an, wenn Sie die gemeinsam genutzte Warteschlange definieren.

„Kanäle und Serialisierung" auf Seite 1083
Bei der Peer-Recovery der gemeinsamen Warteschlange werden Nachrichtenkanalagenten, die Nachrichten in gemeinsam genutzten Warteschlangen verarbeiten, ihren Zugriff auf die Warteschlangen serialisiert.

Einreihung in Warteschlange innerhalb von Gruppen

**Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren" auf Seite 206
Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

„Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten" auf Seite 1052
In diesem Abschnitt werden die Vorbereitungen für IBM MQ for z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

### z/OS *LU 6.2- und TCP/IP-Listener für Gruppen mit gemeinsamer Warteschlange*

Die Gruppen LU 6.2 und TCP/IP-Listener sind an einer Adresse empfangsbereit, die logisch mit der generischen Adresse verbunden ist.

Die angegebene LUGROUP wird für den LU 6.2-Listener der generischen VTAM-Ressource zugeordnet, die zur Gruppe mit gemeinsamer Warteschlange gehört. Ein Beispiel für die Einrichtung dieser Technologie finden Sie unter „Definieren einer LU6.2-Verbindung für z/OS mit APPC/MVS" auf Seite 1077.

Für den TCP/IP-Listener kann der angegebene Port auf eine der folgenden Arten mit der generischen Adresse verbunden werden:

- Für einen Front-End-Router wie den IBM Network Dispatcher werden eingehende Verbindungsanforderungen vom Router an die Mitglieder der Gruppe mit gemeinsamer Warteschlange weitergeleitet.
- Für den TCP/IP-Sysplex-Distributor wird jeder Listener, der ausgeführt wird und auf einer bestimmten Adresse empfangsbereit ist, die als verteilte DVIPA konfiguriert ist, einen Teil der eingehenden Anforderungen zugeordnet. Ein Beispiel für die Einrichtung dieser Technologie finden Sie im Abschnitt Sysplex Distributor verwenden

### z/OS *Übertragungswarteschlangen und Triggering für Gruppen mit gemeinsamer Warteschlange*

In einer gemeinsam genutzten Übertragungswarteschlange werden Nachrichten gespeichert, bevor Sie von der Gruppe mit gemeinsamer Warteschlange in das Ziel verschoben werden.

Es handelt sich um eine gemeinsam genutzte Warteschlange, auf die alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange zugreifen können.

#### Auslösefunktion

Eine ausgelöste gemeinsam genutzte Warteschlange kann für eine erfüllte Auslöserbedingung mehr als eine Auslösenachricht generieren. Es wird eine Auslösenachricht für jede lokale Initialisierungswarteschlange generiert, die in einem Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange definiert ist, die der ausgelösten gemeinsam genutzten Warteschlange zugeordnet ist.

Bei der verteilten Steuerung von Warteschlangen empfängt jeder Kanalinitiator eine Auslösenachricht für eine erfüllte Auslöserbedingung für die gemeinsame Übertragungswarteschlange. Tatsächlich verarbeitet jedoch nur ein Kanalinitiator den ausgelösten Start, und die anderen können sicher nicht ausgeführt werden. Der ausgelöste Kanal wird anschließend mit einem Start mit Lastausgleich gestartet (siehe „IBM MQ for z/OS für DQM mit Gruppen mit gemeinsamer Warteschlange vorbereiten" auf Seite 1078 ). die ausgelöst wird, um Kanal `QSG.TO.QM2`zu starten. Verwenden Sie die IBM MQ-Befehle (MQSC), um eine gemeinsam genutzte Übertragungswarteschlange zu erstellen, wie im folgenden Beispiel gezeigt:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

**Anmerkung:** Wenn eine gemeinsam genutzte Warteschlange für das Auslösen konfiguriert ist und die Verbindung zu der Coupling-Facility, die die gemeinsam genutzte Warteschlange hostet, verloren geht, wird möglicherweise ein Auslöserereignis generiert und eine Nachricht in die Initialisierungswarteschlange eingereiht. Dies kann auch auftreten, wenn keine Nachricht zur Auslösung in die ursprüngliche Konfiguration der gemeinsam genutzten Warteschlange eingereiht wurde. Dies wird durch die Überanzeige von Bits durch das Makro IXLVECTR verursacht, wie in The List Notification Vectordokumentiert.

### z/OS *Nachrichtenkanalagenten für Gruppen mit gemeinsamer Warteschlange*

Ein Kanal kann nur auf einem Kanalinitiator gestartet werden, wenn er Zugriff auf eine Kanaldefinition für einen Kanal mit diesem Namen hat.

Ein Nachrichtenkanalagent ist ein IBM MQ-Programm, das das Senden und Empfangen von Nachrichten steuert. Nachrichtenkanalagenten verschieben Nachrichten von einem WS-Manager in einen anderen; es gibt jeweils einen Nachrichtenkanalagenten an jedem Ende eines Kanals.

Eine Kanaldefinition kann so definiert werden, dass sie für einen WS-Manager privat ist oder im gemeinsam genutzten Repository gespeichert ist und überall verfügbar ist (eine Gruppendefinition). Das bedeutet, dass ein von einer Gruppe definierter Kanal in jedem Kanalinitiator in der Gruppe mit gemeinsamer Warteschlange verfügbar ist.

**Anmerkung:** Die private Kopie der Gruppendefinition kann geändert oder gelöscht werden.

Verwenden Sie die IBM MQ-Befehle (MQSC) wie in den folgenden Beispielen gezeigt, um Gruppenkanaldefinitionen zu erstellen:

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

Die Nachrichtenkanalagenten, die für die verteilte Steuerung von Warteschlangen für Gruppen mit gemeinsamer Warteschlange verwendet werden, können aus zwei Perspektiven betrachtet werden:

## Eingehend

Ein eingehender Kanal ist ein gemeinsam genutzter Kanal, wenn er über den Gruppenlistener mit dem Warteschlangenmanager verbunden ist. Er wird über die generische Schnittstelle mit der Gruppe mit gemeinsamer Warteschlange verbunden und anschließend an einen Warteschlangenmanager in der Gruppe weitergeleitet oder an den Gruppenport eines bestimmten Warteschlangenmanagers oder den vom Gruppenlistener verwendeten LU-Namen übertragen.

## Ausgehend

Ein abgehender Kanal ist ein gemeinsam genutzter Kanal, wenn er Nachrichten aus einer gemeinsam genutzten Übertragungswarteschlange verschiebt. In den Beispielbefehlen ist der Senderkanal `QSG.TO.QM2` ein gemeinsam genutzter Kanal, da die Übertragungswarteschlange QM2 mit QSGDISP (SHARED) definiert ist.

### z/OS *Synchronisationswarteschlange für Gruppen mit gemeinsamer Warteschlange*

Gemeinsam genutzte Kanäle haben ihre eigene Synchronisierungswarteschlange mit dem Namen SYSTEM.QSG.CHANNEL.SYNCQ.

Jedes Mitglied der Gruppe mit gemeinsamer Warteschlange kann auf diese Synchronisationswarteschlange zugreifen. (Private Kanäle verwenden weiterhin die private Synchronisationswarteschlange. Siehe „IBM MQ -Objekte unter z/OS definieren” auf Seite 1054 ). Dies bedeutet, dass der Kanal auf einem anderen Warteschlangenmanager und einer Kanalinitiatorinstanz in der Gruppe mit gemeinsamer Warteschlange erneut gestartet werden kann, wenn das Kommunikationssubsystem, der Kanalinitiator oder der Warteschlangenmanager fehlschlagen. Weitere Informationen finden Sie unter „IBM MQ for z/OS für DQM mit Gruppen mit gemeinsamer Warteschlange vorbereiten” auf Seite 1078.

Für DQM mit Gruppe mit gemeinsamer Warteschlange muss eine gemeinsam genutzte Warteschlange mit dem Namen SYSTEM.QSG.CHANNEL.SYNCQ verfügbar sein. Diese Warteschlange muss verfügbar sein, damit ein Gruppenlistener erfolgreich gestartet werden kann.

Wenn ein Gruppenlistener fehlschlägt, weil die Warteschlange nicht verfügbar war, kann die Warteschlange definiert werden und der Listener kann erneut gestartet werden, ohne dass der Kanalinitiator erneut gestartet werden muss. Die nicht gemeinsam genutzten Kanäle sind davon nicht betroffen.

Stellen Sie sicher, dass diese Warteschlange mit INDXTYPE (MSGID) definiert wird. Diese Definition verbessert die Geschwindigkeit, mit der auf die Nachrichten in der Warteschlange zugegriffen werden kann.

### z/OS *Cluster und Gruppen mit gemeinsamer Warteschlange*

Sie können die gemeinsam genutzte Warteschlange für einen Cluster in einer einzigen Definition verfügbar machen. Geben Sie dazu den Namen des Clusters an, wenn Sie die gemeinsam genutzte Warteschlange definieren.

Benutzer im Netz sehen die gemeinsam genutzte Warteschlange als Host der einzelnen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange. (Die gemeinsam genutzte Warteschlange wird nicht als Host für die Gruppe mit gemeinsamer Warteschlange zugänglich gemacht). Clients können Sitzungen mit allen Mitgliedern der Gruppe mit gemeinsamer Warteschlange starten, um Nachrichten in dieselbe gemeinsam genutzte Warteschlange einzureihen.

Weitere Informationen finden Sie unter „WS-Manager-Cluster konfigurieren” auf Seite 312.

### z/OS *Kanäle und Serialisierung*

Bei der Peer-Recovery der gemeinsamen Warteschlange werden Nachrichtenkanalagenten, die Nachrichten in gemeinsam genutzten Warteschlangen verarbeiten, ihren Zugriff auf die Warteschlangen serialisiert.

Wenn ein Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange fehlschlägt, während ein Nachrichtenkanalagent sich mit nicht festgeschriebenen Nachrichten in einer oder mehreren gemeinsam genutzten Warteschlangen befasst, wird der Kanal und der zugehörige Kanalinitiator beendet, und die Peerwiederherstellung der gemeinsam genutzten Warteschlange wird für den Warteschlangenmanager ausgeführt.

Da die Peerwiederherstellung für gemeinsam genutzte Warteschlangen eine asynchrone Aktivität ist, kann die Peerkanalwiederherstellung versuchen, den Kanal in einem anderen Teil der Gruppe mit gemeinsamer Warteschlange gleichzeitig erneut zu starten, bevor die Peerwiederherstellung der gemeinsam genutzten Warteschlange abgeschlossen ist. Wenn dieses Ereignis eintritt, können festgeschriebene Nachrichten vor den Nachrichten, die noch wiederhergestellt werden, verarbeitet werden. Um sicherzustellen, dass Nachrichten auf diese Weise nicht aus der Sequenz verarbeitet werden, serialisieren Nachrichtenkanalagenten, die Nachrichten in gemeinsam genutzten Warteschlangen verarbeiten, ihren Zugriff auf diese Warteschlangen.

Ein Versuch, einen Kanal zu starten, für den die Peer-Recovery der gemeinsam genutzten Warteschlange noch in Bearbeitung ist, kann zu einem Fehler führen. Es wird eine Fehlernachricht ausgegeben, die darauf hinweist, dass die Wiederherstellung in Bearbeitung ist, und der Kanal wird in den Wiederholungsstatus versetzt. Sobald die Peer-Wiederherstellung des Warteschlangenmanagers abgeschlossen ist, kann der Kanal zum Zeitpunkt der nächsten Wiederholung erneut gestartet werden.

Der Versuch, einen Kanal zu RESOLVE, PING oder DELETE zu versuchen, kann aus demselben Grund fehlschlagen.

### z/OS *Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten*

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dieser Versuch erfolgreich ist, ist es erforderlich, dass die Verbindung definiert und verfügbar ist.

Wählen Sie aus einem der beiden Formen des Kommunikationsprotokolls, das verwendet werden kann:

- TCP
- LU 6.2 über APPC/MVS

Die Informationen unter Beispielkonfiguration - IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange können hilfreich sein.

z/OS *TCP-Verbindung für Gruppen mit gemeinsamer Warteschlange definieren*
Zum Definieren einer TCP-Verbindung für eine Gruppe mit gemeinsamer Warteschlange müssen bestimmte Attribute auf der Sende- und Empfangsseite konfiguriert werden.

Informationen zum Einrichten Ihres TCP finden Sie unter „TCP-Verbindung unter z/OS definieren” auf Seite 1074.

### Sendende Beendigung

Das Feld CONNAME (Verbindungsname) in der Kanaldefinition, mit dem eine Verbindung zu Ihrer Gruppe mit gemeinsamer Warteschlange hergestellt werden soll, muss auf die generische Schnittstelle Ihrer Gruppe mit gemeinsamer Warteschlange gesetzt werden (siehe Gruppen mit gemeinsamer Warteschlange ). Weitere Informationen hierzu finden Sie im Abschnitt Sysplex Distributor verwenden .

### TCP mithilfe einer Gruppe mit gemeinsamer Warteschlange empfangen

Das Empfangen von gemeinsam genutzten Kanalprogrammen wird als Antwort auf eine Startanforderung von dem sendenden Kanal gestartet. Dazu muss ein Listener gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Sie starten dieses Listenerprogramm mit dem Befehl START LISTENER unter Verwendung der Eingangsdisposition der Gruppe oder verwenden die Operationen und Steuerkonsolen.

Alle Gruppenlistener in der Gruppe mit gemeinsamer Warteschlange müssen an demselben Port empfangsbereit sein. Wenn Sie mehr als einen Kanalinitiator in einem einzelnen MVS-Image ausführen, können Sie virtuelle IP-Adressen definieren und das TCP-Empfangsprogramm starten, um nur auf eine bestimmte Adresse oder einen bestimmten Hostnamen zu hören, indem Sie IPADDR im Befehl START LISTENER angeben. (Weitere Informationen finden Sie in START LISTENER .)

z/OS *LU 6.2-Verbindung unter z/OS definieren*
Um eine LU 6.2-Verbindung für eine Gruppe mit gemeinsamer Warteschlange zu definieren, müssen bestimmte Attribute auf dem sendenden und empfangenden Ende konfiguriert werden.

Informationen zum Einrichten von APPC/MVS finden Sie im Abschnitt Kommunikation für z/OS konfigurieren.

### Verbindung zu APPC/MVS wird hergestellt (LU 6.2)

Das Feld für den Verbindungsnamen (CONNAME) in der Kanaldefinition für die Verbindung mit Ihrer Gruppe mit gemeinsamer Warteschlange muss auf den symbolischen Zielnamen gesetzt werden, wie in der Nebeninformationsdatei für APPC/MVS angegeben. Die Partner-LU, die in dieser symbolischen Destination angegeben ist, muss der generische Ressourcenname sein. Weitere Informationen hierzu finden Sie im Abschnitt Sich mit generischen Ressourcen in das Netz definieren .

### Empfangen von LU 6.2 über eine generische Schnittstelle

Die empfangenen gemeinsam genutzten MCAs werden als Antwort auf eine Startanforderung vom sendenden Kanal gestartet. Dazu muss ein Gruppen-Listener-Programm gestartet werden, um eingehende Netzanforderungen zu erkennen und den zugehörigen Kanal zu starten. Das Listenerprogramm ist ein APPC/MVS-Server. Sie starten ihn mit dem Befehl START LISTENER unter Verwendung einer ankommenden Dispositionsgruppe oder verwenden Sie die Operationen und Steuerkonsolen. Sie müssen den LU-Namen angeben, um einen symbolischen Zielnamen zu verwenden, der in der Seitendaten-Datei definiert ist. Weitere Informationen hierzu finden Sie im Abschnitt Sich mit generischen Ressourcen in das Netz definieren .

## z/OS IBM MQ mit IMS verwenden

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die es IBM MQ ermöglichen, mit IMS zu interagieren.

Wenn Sie IBM MQ und IMS für die Zusammenarbeit konfigurieren möchten, müssen Sie die folgenden Tasks ausführen:

- „IMS-Adapter einrichten” auf Seite 1085
- „IMS-Bridge konfigurieren” auf Seite 1092

**Zugehörige Konzepte**

IBM MQ und IMS

„IBM MQ mit CICS verwenden” auf Seite 1093
Um IBM MQ mit CICS verwenden zu können, müssen Sie den IBM MQ CICS-Adapter und optional die Komponenten der IBM MQ CICS bridge konfigurieren.

„OTMA-Exits in IMS verwenden” auf Seite 1096
Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

IMS- und IMS-Brückenanwendungen unter IBM MQ for z/OS

**Zugehörige Tasks**

„Warteschlangenmanager unter z/OS erstellen” auf Seite 966
Verwenden Sie diese Anweisungen zum Konfigurieren von Warteschlangenmanagern unter IBM MQ for z/OS.

**Zugehörige Verweise**

„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen” auf Seite 1094
Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

## z/OS IMS-Adapter einrichten

Für die Verwendung von IBM MQ in IMS ist der Adapter IBM MQ - IMS erforderlich (im Allgemeinen als IMS -Adapter bezeichnet).

In diesem Abschnitt erfahren Sie, wie Sie den IMS -Adapter für Ihr IMS-Subsystem verfügbar machen. Wenn Sie mit der Anpassung eines IMS -Subsystems nicht vertraut sind, lesen Sie die IMS -Dokumentation .

Gehen Sie wie folgt vor, um den IMS-Adapter für IMS-Anwendungen zur Verfügung zu stellen:

1. Definieren Sie IBM MQ mithilfe der IMS-Funktion External Subsystem Attach Facility (ESAF) als ein externes Subsystem für IMS.

   Weitere Informationen finden Sie unter „IBM MQ für IMS definieren” auf Seite 1087.
2. Schließen Sie die IBM MQ-Ladebibliothek thlqual.SCSQAUTH in die JOBLIB- oder STEPLIB-Verknüpfung in der Jobsteuersprache für Ihre IMS-Steuerregion und jede abhängige Region ein, die eine Verbindung mit IBM MQ herstellt (falls nicht im Link-Pack-Bereich oder in der Linkliste enthalten). Wenn Ihre JOBLIB oder STEPLIB nicht berechtigt ist, fügen Sie sie auch in die DFSESL-Verknüpfung hinter der Bibliothek mit den IMS-Modulen (normalerweise IMS RESLIB) ein.

   Geben Sie außerdem thlqual.SCSQANLx an (wobei x für den Sprachenbuchstaben steht).

   Wenn DFSESL vorhanden ist, müssen SCSQAUTH und SCSQANLx in die Verkettung eingeschlossen oder der LNKLIST hinzugefügt werden. Das Hinzufügen zur STEPLIB-oder JOBLIB-Verkettung in der JCL ist nicht ausreichend.
3. Kopieren Sie das IBM MQ-Assemblerprogramm CSQQDEFV aus thlqual.SCSQASMS in eine Benutzerbibliothek.
4. Das mitgelieferte Programm CSQQDEFV enthält ein einziges Subsystem mit dem Namen CSQ1, das standardmäßig mit dem IMS-Sprachschnittstellentoken (LIT) MQM1 gekennzeichnet ist. Sie können diesen Namen für Test-und Installationsprüfung beibehalten.

   Für Produktionssubsysteme ändern Sie NAME=CSQ1 in Ihren eigenen Subsystemnamen oder verwenden Sie CSQ1. Sie können bei Bedarf weitere Subsystemdefinitionen hinzufügen. Weitere In-

formationen zu LITs finden Sie unter „Definieren von IBM MQ-Warteschlangenmanagern für den IMS-Adapter” auf Seite 1090.

5. Assemblieren und verknüpfen Sie das Programm, um das Lademodul CSQQDEFV zu erstellen. Geben Sie für die Assemblierung die Bibliothek thlqual.SCSQMACS in Ihrer SYSLIB-Verkettung an; verwenden Sie den Parameter 'link-edit' RENT. Dies wird in der Beispiel-JCL in thlqual.SCSQPROC (CSQ4DEFV) angezeigt.
6. Schließen Sie die Benutzerbibliothek mit dem Modul CSQQDEFV, das Sie erstellt haben, in die JOBLIB- oder STEPLIB-Verknüpfung in der Jobsteuersprache für jede abhängige Region ein, die eine Verbindung mit IBM MQ herstellt. Stellen Sie diese Bibliothek vor den Wert SCSQAUTH, da SCSQAUTH ein Standardlademodul hat. Wenn Sie dies nicht tun, erhalten Sie von IMS die Benutzerabbruchnachricht 3041.
7. Wenn der IMS-Adapter einen unerwarteten IBM MQ-Fehler feststellt, gibt er einen z/OS-SNAP-Speicherauszug mit dem Datendefinitionsnamen CSQSNAP aus und gibt den Ursachencode MQRC_UNERWARTETEN Fehler an die Anwendung aus. Wenn die CSQSNAP-DD-Anweisung nicht in der Jobsteuersprache der abhängigen IMS-Region enthalten war, wird kein Speicherauszug erstellt. In diesem Fall könnten Sie die DD-Anweisung CSQSNAP in die JCL aufnehmen und die Anwendung erneut ausführen. Da einige Probleme jedoch gelegentlich auftreten, wird empfohlen, dass Sie die DD-Anweisung CSQSNAP einschließen, um die Ursache für einen Fehler zu dem Zeitpunkt zu erfassen, zu dem sie aufgetreten ist.
8. Wenn Sie dynamische IBM MQ-Aufrufe (beschrieben in IBM MQ-Stub dynamisch aufrufen ) verwenden möchten, erstellen Sie den dynamischen Stub, wie in Abbildung 117 auf Seite 1087 dargestellt.
9. Wenn Sie den IMS-Auslösemonitor verwenden möchten, definieren Sie die IMS-Auslösemonitoranwendung CSQQTRMN und führen Sie PSBGEN und ACBGEN aus. Siehe „IMS-Auslösemonitor konfigurieren” auf Seite 1092.
10. Wenn Sie RACF zum Schutz von Ressourcen in der Klasse OPERCMDS verwenden, müssen Sie sicherstellen, dass die Benutzer-ID, die Ihrem Adressraum des IBM MQ-Warteschlangenmanagers zugeordnet ist, über die Berechtigung verfügt, den Befehl MODIFY für ein beliebiges IMS-System auszugeben, zu dem es eine Verbindung herstellen kann.

```
//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD  SYSOUT=*
//ACSQMOD  DD  DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB   DD  DISP=SHR,DSN=ims.reslib
//SYSLMOD  DD  DISP=SHR,DSN=private.load[1]
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN   DD  *
  INCLUDE ACSQMOD(CSQQSTUB)
  INCLUDE IMSLIB(DFSLI000)
  ALIAS MQCONN,MQCONNX,MQDISC             MQI entry points
  ALIAS MQGET,MQPUT,MQPUT1                MQI entry points
  ALIAS MQOPEN,MQCLOSE                    MQI entry points
  ALIAS MQBACK,MQCMIT                     MQI entry points
  ALIAS CSQBBAK,CSQBCMT                   MQI entry points
  ALIAS MQINQ,MQSET                       MQI entry points
  ALIAS DFSPLI,PLITDLI                    IMS entry points
  ALIAS DFSCOBOL,CBLTDLI                  IMS entry points
  ALIAS DFSFOR,FORTDLI                    IMS entry points
  ALIAS DFSASM,ASMTDLI                    IMS entry points
  ALIAS DFSPASCL,PASTDLI                  IMS entry points
  ALIAS DFHEI01,DFHEI1                    IMS entry points
  ALIAS DFSAIBLI,AIBTDLI                  IMS entry points
  ALIAS DFSESS,DSNWLI,DSNHLI              IMS entry points
  ALIAS MQCRTMH,MQDLTMH,MQDLTMP           IMS entry points
  ALIAS MQINQMP,MQSETMP,MQMHBUF,MQBUFMH   IMS entry points
  MODE AMODE(31),RMODE(24)                Note RMODE setting
  NAME CSQQDYNS(R)
/*

[1]Specify the name of a library accessible to IMS applications that
want to make dynamic calls to IBM MQ.
```

*Abbildung 117. Beispiel-JCL zum Verlinken-Bearbeiten des Stubs für dynamische Verbindungen*

**Zugehörige Konzepte**

IBM MQ und IMS

„IMS-Bridge konfigurieren” auf Seite 1092

Die IBM MQ - IMS-Bridge ist eine optionale Komponente, die IBM MQ die Eingabe und Ausgabe von vorhandenen Programmen und Transaktionen, die nicht IBM MQ-fähig sind, ermöglicht.

IMS- und IMS-Brückenanwendungen unter IBM MQ for z/OS

## z/OS *IBM MQ für IMS definieren*

IBM MQ muss für die IMS-Steuerregion und für jede abhängige Region, die auf diesen IBM MQ-Warteschlangenmanager zugreift, definiert sein. Dazu müssen Sie ein Subsystem-Member (SSM) in der IMS.PROCLIB-Bibliothek erstellen und den SSM in den entsprechenden IMS-Regionen identifizieren.

### Das Subsystem-Member-Eintrag in IMS wird platziert.PROCLIB

Jeder SSM-Eintrag in IMS.PROCLIB definiert eine Verbindung von einer IMS-Region zu einem anderen Warteschlangenmanager.

Um ein SSM zu benennen, verketten Sie den Wert (ein bis vier alphanumerische Zeichen) im Feld IMS-ID des Makros IMS IMSCTRL mit einem beliebigen Namen (ein bis vier alphanumerische Zeichen), der von Ihrer Site definiert wurde.

Ein SSM kann von allen IMS-Regionen gemeinsam genutzt werden, oder es kann ein bestimmtes Mitglied für jede Region definiert werden. Dieses Member enthält so viele Einträge, wie Verbindungen zu externen Subsystemen vorhanden sind. Jeder Eintrag ist ein 80-stelliger Datensatz.

**Positionsgebundene Parameter**

Die Felder in diesem Eintrag lauten wie folgt:

```
SSN,LIT,ESMT,RTT,REO,CRC
```

Dabei gilt:

**SSN**
Gibt den Namen des IBM MQ-Warteschlangenmanagers an. Es ist erforderlich und muss ein bis vier Zeichen enthalten.

**LIT**
Gibt das Sprachschnittstellentoken (LIT) an, das IMS bereitgestellt wird. Dieses Feld ist erforderlich, sein Wert muss mit einem Wert im Modul CSQQDEFV übereinstimmen.

**ESMT**
Gibt die Modultabelle des externen Subsystems (ESMT) an. In dieser Tabelle wird angegeben, welche Anhangsmodule von IMS geladen werden müssen. CSQQESMT ist der erforderliche Wert für dieses Feld.

**RTT**
Diese Option wird von IBM MQ nicht unterstützt.

**REO**
Gibt die Regionsfehleroption (REO) an, die verwendet werden soll, wenn eine IMS-Anwendung auf ein inaktives externes Subsystem verweist oder wenn Ressourcen zur Zeit der Thread-Erstellung nicht verfügbar sind. Dieses Feld ist optional und enthält ein einzelnes Zeichen, das Folgendes sein kann:

**R**
Übergibt einen Rückkehrcode an die Anwendung und gibt an, dass die Anforderung für IBM MQ-Services fehlgeschlagen ist.

**Q**
Beendet die Anwendung mit dem Code für abnormale Beendigung U3051, setzt die Aktivität auf den letzten Commitpunkt zurück, führt einen PSTOP der Transaktion aus und stellt die Eingabenachricht erneut in die Warteschlange. Diese Option gilt nur, wenn eine IMS-Anwendung versucht, auf ein inaktives externes Subsystem zu verweisen, oder wenn die Ressourcen zur Zeit der Thread-Erstellung nicht verfügbar sind.

IBM MQ-Beendigungscodes und Ursachencodes werden an die Anwendung zurückgegeben, wenn das IBM MQ-Problem auftritt, während IBM MQ die Anforderung verarbeitet. Dies ist der Fall, nachdem der Adapter die Anforderung an IBM MQ übergeben hat.

**A**
Beendet die Anwendung mit dem Code für abnormale Beendigung U3047 und löscht die Eingabenachricht. Diese Option gilt nur, wenn eine IMS-Anwendung auf ein inaktives externes Subsystem verweist oder wenn die Ressourcen zur Zeit der Thread-Erstellung nicht verfügbar sind.

IBM MQ-Beendigungscodes und Ursachencodes werden an die Anwendung zurückgegeben, wenn das IBM MQ-Problem auftritt, während IBM MQ die Anforderung verarbeitet. Dies ist der Fall, nachdem der Adapter die Anforderung an IBM MQ übergeben hat.

**CRC**
Diese Option kann angegeben werden, wird aber von IBM MQ nicht verwendet.

**Anmerkung:** Ausführliche Informationen zu allen positionsgebundenen Parametern finden Sie in Die Angabe von externen Subsystemen für IMS.

Ein Beispiel für einen SSM-Eintrag ist:

```
CSQ1,MQM1,CSQQESMT,,R,
```

Dabei gilt:

| | |
|---|---|
| **CSQ1** | Der Standardsubsystemname, der im Lieferumfang von IBM MQ enthalten ist. Sie können diese Änderung an Ihre Installation anpassen. |
| **MQM1** | Der Standardwert LIT wird in CSQQDEFV angegeben. |
| **CSQQESMT** | Der Name des externen Subsystemmoduls. Sie müssen diesen Wert verwenden. |
| **R** | REO-Option. |

**Schlüsselwortparameter**

IBM MQ-Parameter können im Schlüsselwortformat angegeben werden. Der Parameter SST kann einen Wert von DB2 oder MQ haben. Die Unterstützung für den MQ-Wert wurde in IMS 14 hinzugefügt. Die Verwendung von MQ trägt zur Klarheit bei, und der IMS-Subsystembefehl enthält jetzt den SST-Wert, hat aber ansonsten keine nennenswerten Auswirkungen. Ein Wert von DB2 kann bei Bedarf weiterhin verwendet werden. Weitere Parameter sind wie in positionsgebundene Parameter beschrieben und im folgenden Beispiel dargestellt:

```
SST=MQ,SSN=SYS3,LIT=MQM3,ESMT=CSQQESMT
```

Dabei gilt:

| | |
|---|---|
| **SYS3** | Der Subsystemname |
| **MQM3** | Die LIT wie in CSQQDEFV angegeben |
| **CSQQESMT** | Der Modulname des externen Subsystems. |

## Geben Sie den Parameter SSM EXEC an.

Geben Sie den Parameter SSM EXEC in der Startprozedur der Steuerregion von IMS an. Dieser Parameter gibt das aus einem Zeichen zu vierstelligee Subsystemmember-Name (SSM) an.

Wenn Sie den SSM für die Steuerregion IMS angeben, kann jede abhängige Region, die unter dem Steuerbereich ausgeführt wird, dem IBM MQ-Warteschlangenmanager zugeordnet werden, der in der IMS benannt ist.PROCLIB-Member durch den SSM-Parameter angegeben. Der IMS.Der PROCLIB-Membername ist die IMS-ID (IMSID=*xxxx*), die mit einem bis zu vier Zeichen verknüpft ist, die im Parameter SSM EXEC angegeben sind. Die IMS-ID ist der Parameter für die IMS-ID des IMS-CTRL-Generierungsmakros.

In IMS können Sie so viele externe Subsystemverbindungen definieren, wie dies erforderlich ist. Für verschiedene IBM MQ-Warteschlangenmanager kann mehr als eine Verbindung definiert werden. Alle IBM MQ-Verbindungen müssen sich innerhalb desselben z/OS-Systems befinden. Für eine abhängige Region können Sie eine abhängige Region SSM angeben oder die Region verwenden, die für die Steuerregion angegeben wurde. Sie können im SSM der abhängigen Region und im SSM der Steuerregion unterschiedliche Regionsfehleroptionen (REOs) angeben. Tabelle 67 auf Seite 1089 zeigt die verschiedenen Möglichkeiten von SSM-Spezifikationen.

*Tabelle 67. Optionen für SSM-Spezifikationen*

| SSM für Steuerregion | SSM für abhängige Region | Action | Kommentare |
|---|---|---|---|
| Nein | Nein | -- | Es kann kein externes Subsystem angeschlossen werden. |
| Nein | Ja | -- | Es kann kein externes Subsystem angeschlossen werden. |

| *Tabelle 67. Optionen für SSM-Spezifikationen (Forts.)* | | | |
|---|---|---|---|
| **SSM für Steuerregion** | **SSM für abhängige Region** | **Action** | **Kommentare** |
| Ja | Nein | SSM der Steuerregion verwenden | Anwendungen, die in der Region geplant sind, können auf externe Subsysteme zugreifen, die in der Steuerregion SSM angegeben sind. Exits und Steuerblöcke für jeden Anhang werden in den Steuerbereich und die Adressräume der abhängigen Region geladen. |
| Ja | Ja (leer) | Für die abhängige Region wird kein SSM verwendet. | Anwendungen, die in dieser Region geplant sind, können nur auf DL/I-Datenbanken zugreifen. Exits und Steuerblöcke für jeden Anhang werden in den Adressraum der Steuerregion geladen. |
| Ja | Ja (nicht leer) | Überprüfen Sie die abhängige Region SSM mit der Steuerregion SSM. | Anwendungen, die in dieser Region geplant sind, können nur auf externe Subsysteme zugreifen, die in beiden SSMs angegeben sind. Exits und Steuerblöcke für jeden Anhang werden in den Steuerbereich und die Adressräume der abhängigen Region geladen. |

Es gibt keinen bestimmten Parameter, um die maximale Anzahl von SSM-Spezifikationsmöglichkeiten zu steuern.

## IMS-Adapter vorinstallieren

Die Leistung des IMS-Adapters kann verbessert werden, wenn er von IMS vorinstalliert wird. Die Vorinstallation wird durch das Member DFSMPLxx von IMS.PROCLIB gesteuert: Weitere Informationen finden Sie im Handbuch "IMS Administration Guide: System". Die anzugebenden IBM MQ-Modulnamen lauten wie folgt:

| | | | | | |
|---|---|---|---|---|---|
| CSQACLST | CSQAMLST | CSQAPRH | CSQAVICM | CSQFSALM | CSQQDEFV |
| CSQQCONN | CSQQDISC | CSQQTERM | CSQQINIT | CSQQBACK | CSQQCMMT |
| CSQQESMT | CSQQPREP | CSQQTTHD | CSQQWAIT | CSQQNORM | CSQQSSOF |
| CSQQSSON | CSQFSTAB | CSQQRESV | CSQQSNOP | CSQQCMND | CSQQCVER |
| CSQQTMID | CSQQTRGI | CSQQCON2 | CSQBPAPI | CSQBCRMH | CSQBAPPL |

Weitere Informationen zur Verwendung von IBM MQ classes for JMS finden Sie unter Using IBM MQ classes for JMS in IMS.

Aktuelle Releases von IMS unterstützen die Vorinstallation von IBM MQ-Modulen aus PDS-E-Formatbibliotheken nur in MPP-, BMP-, IFP-, JMP-und JBP-Regionen. Alle anderen IMS-Regionen unterstützen die Vorinstallation aus PDS-E-Bibliotheken nicht. Wenn die Vorinstallation für eine beliebige andere Region erforderlich ist, müssen die bereitgestellten IBM MQ-Module in eine PDS-Formatbibliothek kopiert werden.

### z/OS *Definieren von IBM MQ-Warteschlangenmanagern für den IMS-Adapter*

Die Namen der WS-Manager von IBM MQ und die zugehörigen Sprachenschnittstellentoken (LITs) müssen in der Definitionstabelle des Warteschlangenmanagers definiert sein.

Verwenden Sie das bereitgestellte Makro CSQQDEFX, um das Lademodul CSQQDEFV zu erstellen. Abbildung 118 auf Seite 1091 zeigt die Syntax dieses Assemblermakros.

```
CSQQDEFX TYPE=ENTRY|DEFAULT,NAME=qmgr-name,LIT=token
or
CSQQDEFX TYPE=END
```

*Abbildung 118. CSQQDEFX, Makrosyntax*

**Parameter**

**TYPE=ENTRY | STANDARD**
Geben Sie TYPE=ENTRY oder TYPE=DEFAULT wie folgt an:

**TYPE=ENTRY**
Gibt an, dass ein Tabelleneintrag, der einen IBM MQ-Warteschlangenmanager beschreibt, der für eine IMS-Anwendung verfügbar ist, generiert werden soll. Ist dies der erste Eintrag, wird auch der Tabellenheader generiert, einschließlich einer Anweisung CSQQDEFV CSECT.

**TYPE=DEFAULT**
Wie bei TYPE=ENTRY. Der angegebene WS-Manager ist der Standardwarteschlangenmanager, der verwendet werden soll, wenn MQCONN oder MQCONNX einen Namen angibt, der alle Leerzeichen enthält. Es darf nur einen solchen Eintrag in der Tabelle geben.

**NAME= *qmgr-name***
Gibt den Namen des Warteschlangenmanagers an, wie mit **MQCONN** oder **MQCONNX**angegeben.

**LIT = Token**
Gibt den Namen des Sprachenschnittstellentokens (LIT) an, das IMS für die Identifikation des Warteschlangenmanagers verwendet.

Ein MQCONN -oder MQCONNX -Aufruf verknüpft den Eingabeparameter *name* und den Ausgabeparameter *hconn* mit dem Namenskennsatz und damit die LIT im Eintrag CSQQDEFV. Weitere IBM MQ-Aufrufe, die den Parameter *hconn* übergeben, verwenden das LIT aus dem Eintrag CSQQDEFV, der im Aufruf MQCONN oder MQCONNX angegeben ist, um Aufrufe an den IBM MQ-Warteschlangenmanager zu leiten, der im Member IMS SSM PROCLIB mit dieser LIT-Datei definiert ist.

Der Parameter **name** im Aufruf MQCONN oder MQCONNX gibt somit eine LIT in CSQQDEFV an, und die gleiche LIT im SSM-Member gibt einen IBM MQ-Warteschlangenmanager an. (Weitere Informationen zum MQCONN -Aufruf finden Sie unter MQCONN-Verbindungswarteschlangenmanager. Informationen zum Aufruf MQCONNX finden Sie unter MQCONNX-Connect queue manager (extended) .)

**TYPE=ENDE**
Gibt an, dass die Tabelle vollständig ist. Wird dieser Parameter nicht angegeben, wird TYPE=ENTRY angenommen.

**Makro CSQQDEFX verwenden**

Abbildung 119 auf Seite 1091 zeigt das allgemeine Layout einer Warteschlangendefinitionstabelle.

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

*Abbildung 119. Layout einer Warteschlangenmanager-Definitionstabelle*

### z/OS *IMS-Auslösemonitor konfigurieren*

Sie können ein Stapelverarbeitungsprogramm für IMS einrichten, um eine Initialisierungswarteschlange für IBM MQ zu überwachen.

Definieren Sie die Anwendung für IMS mit Hilfe des Modells CSQQTAPL in der Bibliothek thlqual.SCSQPROC (siehe Beispiel einer Transaktionsdefinition für CSQQTRMN).

Generieren Sie die PSB und ACB mit Hilfe des Modells CSQQTPSB in der Bibliothek thlqual.SCSQPROC (siehe Beispiel-PSB-Definition für CSQQTRMN ).

```
*  This is the application definition *
*  for the IMS Trigger Monitor BMP    *

APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

*Abbildung 120. Beispieltransaktionsdefinition für CSQQTRMN*

```
PCB  TYPE=TP,      ALTPCB for transaction messages
MODIFY=YES,        To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB  TYPE=TP,      ALTPCB for diagnostic messages
MODIFY=YES,        To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN,  Runs program CSQQTRMN
CMPAT=YES
```

*Abbildung 121. Beispiel-PSB-Definition für CSQQTRMN*

Weitere Informationen zum Starten und Stoppen des IMS-Auslösemonitors finden Sie unter Auslösemonitor für IMS steuern.

## z/OS IMS-Bridge konfigurieren

Die IBM MQ - IMS-Bridge ist eine optionale Komponente, die IBM MQ die Eingabe und Ausgabe von vorhandenen Programmen und Transaktionen, die nicht IBM MQ-fähig sind, ermöglicht.

In diesem Abschnitt wird beschrieben, was Sie tun müssen, um die IBM MQ-IMS-Bridge anzupassen.

**Definieren Sie die XCF-und OTMA-Parameter für IBM MQ.**

In diesem Schritt werden die XCF-Gruppen- und -Membernamen für Ihr IBM MQ-System und andere OTMA-Parameter definiert. IBM MQ und IMS müssen zu derselben XCF-Gruppe gehören. Verwenden Sie das Schlüsselwort OTMACON des Makros CSQ6SYSP, um diese Parameter im Lademodul des Systemparameters anzupassen.

Weitere Informationen hierzu finden Sie im Abschnitt CSQ6SYSP verwenden .

**Definieren Sie die XCF-und OTMA-Parameter für IMS.**

In diesem Schritt werden die XCF-Gruppen-und -Membernamen für das IMS-System definiert. IMS und IBM MQ müssen zu derselben XCF-Gruppe gehören.

Fügen Sie die folgenden Parameter in Ihrer IMS-Parameterliste hinzu, entweder in der JCL oder in der Teildatei DFSPBxxx in IMS PROCLIB:

**OTMA=Y**

Dadurch wird OTMA automatisch gestartet, wenn IMS gestartet wird. (Optional: Wenn Sie OTMA=N angeben, können Sie OTMA auch starten, indem Sie den IMS-Befehl /START OTMA absetzen.)

**GRNAME=**
Dieser Parameter gibt den Namen der XCF-Gruppe an.

Es entspricht dem Gruppennamen, der in der Definition der Speicherklasse angegeben ist (siehe nächsten Schritt), und im Parameter **Group** des Schlüsselworts OTMACON des Makros CSQ6SYSP.

**OTMANM=**
Dieser Parameter gibt den XCF-Membernamen des IMS-Systems an.

Dies ist mit dem in der Speicherklassendefinition angegebenen Membernamen identisch (siehe nächsten Schritt).

**Informieren Sie IBM MQ über die XCF-Gruppe und den Membernamen des IMS-Systems.**
Dies wird durch die Speicherklasse einer Warteschlange angegeben. Wenn Sie Nachrichten über die IBM MQ - IMS-Bridge senden möchten, müssen Sie diese angeben, wenn Sie die Speicherklasse für die Warteschlange definieren. In der Speicherklasse müssen Sie die XCF-Gruppe und den Membernamen des IMS-Zielsystems definieren. Verwenden Sie dazu entweder die IBM MQ-Operationen und -Steuerkonsolen oder verwenden Sie die IBM MQ-Befehle wie im Abschnitt Einführung in Programmierbare Befehlsformate beschrieben.

**Richten Sie die Sicherheitsfunktion ein, die Sie benötigen.**
Der Befehl /SECURE OTMA IMS bestimmt die Sicherheitsstufe, die auf **jeden** IBM MQ-Warteschlangenmanager angewendet werden soll, der eine Verbindung zu IMS über OTMA herstellt. Weitere Informationen finden Sie unter Sicherheitsaspekte für die Verwendung von IBM MQ mit IMS.

## Zusätzliche IMS-Verbindung zum gleichen Warteschlangenmanager hinzufügen

Um eine IMS-Verbindung zu demselben Warteschlangenmanager hinzuzufügen, müssen Sie eine zweite Speicherklasse (STGCLASS) definieren, die auf die neue IMS verweist. Weitere Informationen finden Sie unter DEFINE STGCLASS.

**Wichtig:**

- Eine lokale Warteschlange kann nicht auf zwei Speicherklassen verweisen.
- Eine Speicherklasse kann nicht auf zwei IMS-Bridges verweisen.
- IBM MQ und IMS müssen zu derselben XCF-Gruppe gehören. Verwenden Sie das Schlüsselwort OTMACON des Makros CSQ6SYSP, um diese Parameter im Lademodul des Systemparameters anzupassen.

  Weitere Informationen hierzu finden Sie im Abschnitt CSQ6SYSP verwenden .

**Zugehörige Konzepte**
IBM MQ und IMS
„IMS-Adapter einrichten” auf Seite 1085
Für die Verwendung von IBM MQ in IMS ist der Adapter IBM MQ - IMS erforderlich (im Allgemeinen als IMS -Adapter bezeichnet).

IMS- und IMS-Brückenanwendungen unter IBM MQ for z/OS

# z/OS IBM MQ mit CICS verwenden

Um IBM MQ mit CICS verwenden zu können, müssen Sie den IBM MQ CICS-Adapter und optional die Komponenten der IBM MQ CICS bridge konfigurieren.

Weitere Informationen zum Konfigurieren des IBM MQ CICS-Adapters und der IBM MQ CICS bridge-Komponenten finden Sie im Abschnitt Verbindungen zu MQ konfigurieren in der Dokumentation zu CICS.

**Zugehörige Konzepte**
IBM MQ und CICS
„IBM MQ mit IMS verwenden” auf Seite 1084
Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die es IBM MQ ermöglichen, mit IMS zu interagieren.

**Zugehörige Verweise**

„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen” auf Seite 1094

Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

## z/OS Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen

Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

In der folgenden Tabellen ist dargestellt, was Sie mit IBM MQ for z/OS tun müssen, wenn Sie für die folgenden Produkte ein Upgrade oder Serviceaktualisierungendurchführen:

- Language Environment
- z/OS Callable Services (z. B. APPC und RRS)

*Tabelle 68. Der Service wurde angewendet, oder das Produkt wurde auf ein neues Release aktualisiert.*

| **Produkt** | **Aktion bei Verwendung von CALLLIBS und SMP/E V3r2 oder höher**<br><br>**Anmerkung: Es ist nicht erforderlich, separate Jobs für Language Environment und Callable Services auszuführen. Ein Job reicht aus.** | **Aktion bei Verwendung von LINK** |
|---|---|---|
| Language Environment | 1. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.<br>2. Geben Sie auf der SMPCNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie unter *z/OS SMP/E-Befehle* .<br>3. Führen Sie den SMP/E-Job aus. | Es ist keine Aktion erforderlich, vorausgesetzt, die SMP/E-Zonen wurden für die automatische Wiederverlinkung konfiguriert, und der Job CSQ8SLDQ wurde ausgeführt. |
| Aufrufbare Services | 1. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.<br>2. Geben Sie auf der SMPCNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie unter *z/OS SMP/E-Befehle* .<br>3. Führen Sie den SMP/E-Job aus. | Es ist keine Aktion erforderlich, vorausgesetzt, die SMP/E-Zonen wurden für die automatische Wiederverlinkung konfiguriert, und der Job CSQ8SLDQ wurde ausgeführt. |

*Tabelle 69. Eines der Produkte wurde in einer neuen SMP/E-Umgebung und Bibliotheken auf ein neues Release aktualisiert.*

| **Produkt** | **Aktion bei Verwendung von CALLLIBS und SMP/E V3r2 oder höher**<br><br>**Anmerkung: Sie müssen keine drei separaten Jobs für Language Environment und Callable Services ausführen. Für beide Produkte reicht ein Job aus.** | **Aktion bei Verwendung von LINK** |
|---|---|---|
| Language Environment | 1. Ändern Sie die DDDEFs für SCEELKED und SCEESPC so, dass sie auf die neue Bibliothek zeigen.<br>2. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.<br>3. Geben Sie auf der SMPCNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie unter *z/OS SMP/E-Befehle* .<br>4. Führen Sie den SMP/E-Job aus. | 1. Löschen Sie die XZMOD-Untereinträge für die folgenden LMOD-Einträge in der IBM MQ for z/OS -Zielzone:<br><br>CMQXDCST, CMQXRCTL, CMQXSUPR, CSQCBE00, CSQCBE30, CSQCBP00, CSQCBP10, CSQCBR00, CSQUCVX, CSQUDLQH, CSQVXPCB, CSQVXSPT, CSQXDCST, CSQXRCTL, CSQXSUPR, CSQXTCMI, CSQXTCP, CSQXTNSV, CSQ7DRPS, IMQB23IC, IMQB23IM, IMQB23IR, IMQS23IC, IMQS23IM, IMQS23IR, IMQS23IC<br>2. Richten Sie die entsprechenden ZONEINDEXs zwischen den IBM MQ-Zonen und den Language Environment-Zonen ein.<br>3. Geben Sie CSQ8SLDQ an, um auf die neue Zone im Parameter FROMZONE der LINK-Befehle zu verweisen. CSQ8SLDQ kann in der Bibliothek SCSQINST gefunden werden.<br>4. Führen Sie CSQ8SLDQ aus. |
| Aufrufbare Services | 1. DDDEF für CSSLIB so ändern, dass sie auf die neue Bibliothek verweist<br>2. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.<br>3. Geben Sie auf der SMPCNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie unter *z/OS SMP/E-Befehle* .<br>4. Führen Sie den SMP/E-Job aus. | 1. Löschen Sie die XZMOD-Untereinträge für die folgenden LMOD-Einträge in der IBM MQ for z/OS -Zielzone:<br><br>CMQXRCTL, CMQXSUPR, CSQBSRV, CSQILPLM, CSQXJST, CSQXRCTL, CSQXSUPR, CSQ3AMGP, CSQ3EPX, CSQ3REPL<br>2. Richten Sie die entsprechenden ZONEINDEXs zwischen den IBM MQ-Zonen und den Callable Services-Zonen ein.<br>3. Geben Sie CSQ8SLDQ an, um auf die neue Zone im Parameter FROMZONE der LINK-Befehle zu verweisen. CSQ8SLDQ kann in der Bibliothek SCSQINST gefunden werden.<br>4. Führen Sie CSQ8SLDQ aus. |

Ein Beispiel für einen Job, der bei Verwendung von CALLLIBS eine erneute Verbindung von Modulen durchführt, finden Sie im Abschnitt „Job LINK CALLLIBS ausführen” auf Seite 1095.

## z/OS Job LINK CALLLIBS ausführen

Ein Beispieljob zum Wiederverbinden von Modulen, wenn CALLLIBS verwendet wird.

Das folgende Beispiel zeigt, wie der Job bei Verwendung von CALLLIBs auf einem SMP/E-V3r2-System Module erneut verbinden kann. Sie müssen eine JOBCARD und den Namen der SMP/E-CSI-Datei angeben, die IBM MQ for z/OS enthält.

```
//*******************************************************************
//* RUN LINK CALLLIBS.
//*******************************************************************
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI  DD DSN=your.csi
//       DISP=SHR
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZONE).
LINK LMODS CALLLIBS .
/*
```

*Abbildung 122. Beispiel für einen SMP/E-LINK-CALLLIBS-Job*

# z/OS OTMA-Exits in IMS verwenden

Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

Wenn Sie eine Ausgabe von einer IMS-Transaktion an IBM MQ senden möchten und diese Transaktion nicht von IBM MQ stammt, müssen Sie einen oder mehrere IMS-OTMA-Exits codieren.

Wenn Sie die Ausgabe an ein Nicht-OTMA-Ziel senden möchten und die Transaktion ihren Ursprung in IBM MQ hat, müssen Sie auch einen oder mehrere IMS-OTMA-Exits codieren.

Die folgenden Exits sind in IMS verfügbar, damit Sie die Verarbeitung zwischen IMS und IBM MQ anpassen können:

- Ein OTMA-Exit vor der Weiterleitung
- Exit für Zielauflösungsbenutzer (DRU)

## OTMA-Exitnamen

Sie müssen den Vorleitwegausgang DFSYPRX0 benennen. Sie können den DRU-Exit so lange benennen, wie er nicht in Konflikt mit einem Modulnamen steht, der bereits in IMS enthalten ist.

### Angeben des Benutzerexitnamens für die Zielauflösung

Sie können den Parameter *Druexit* des Schlüsselworts OTMACON im Makro CSQ6SYSP verwenden, um den Namen des OTMA-DRU-Exits anzugeben, der von IMS ausgeführt werden soll.

Um die Objekt-ID zu vereinfachen, sollten Sie eine Namenskonvention von DRU0xxxx annehmen, wobei xxxx der Name Ihres IBM MQ-Warteschlangenmanagers ist.

Wenn Sie den Namen eines DRU-Exits im Parameter OTMACON nicht angeben, wird der Standardwert DFSYDRU0 verwendet. Weitere Informationen finden Sie in DFSYDRU0 .

### Namenskonvention für IMS-Zieladresse

Sie benötigen eine Namenskonvention für das Ziel, an das Sie die Ausgabe von Ihrem IMS-Programm senden. Dies ist die Destination, die im Aufruf CHNG Ihrer IMS-Anwendung festgelegt ist oder die in IMS PSB voreingestellt ist.

## Beispielszenario für einen OTMA-Exit

Verwenden Sie die folgenden Themen für ein Beispiel für einen Exit vor der Routing-Weiterleitung und einen Exit für Zielweiterleitung für IMS:

- „Exit-Exit DFSYPRX0 vor dem Routing” auf Seite 1097
- „Benutzerexit für Zielauflösung” auf Seite 1098

Um die Identifikation zu vereinfachen, machen Sie den OTMA-Zielnamen ähnlich wie der Name des IBM MQ-Warteschlangenmanagers, z. B. durch Wiederholung des Namens des IBM MQ-Warteschlangenmanagers. Wenn der Name des IBM MQ-Warteschlangenmanagers in diesem Fall "**VCPE**" lautet, dann heißt das Ziel, das durch den CHNG-Aufruf festgelegt wird, "**VCPEVCPE**".

**Zugehörige Konzepte**

IBM MQ und IMS

„IBM MQ mit IMS verwenden” auf Seite 1084
Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die es IBM MQ ermöglichen, mit IMS zu interagieren.

IMS- und IMS-Brückenanwendungen unter IBM MQ for z/OS

## z/OS Exit-Exit DFSYPRX0 vor dem Routing

Dieser Abschnitt enthält einen Beispielexit für die Vorweiterleitung für OTMA in IMS.

Sie müssen zuerst einen Vor-Routing-Exit DFSYPRX0 codieren. Weitere Informationen zu Parametern, die von IMSan diese Routine übergeben werden, finden Sie unter Benutzerexit für OTMA-Zielauflösung (DFSYPRX0 und andere OTMAYPRX-Exits) .

Dieser Exit prüft, ob die Nachricht für ein bekanntes OTMA-Ziel bestimmt ist (in unserem Beispiel VCPEVCPE). Ist dies der Fall, muss der Exit überprüfen, ob die Nachricht, die die Nachricht gesendet hat, in OTMA gesendet wurde. Wenn die Nachricht von OTMA stammt, hat sie einen OTMA-Header, so dass Sie von DFSYPRX0 mit dem Register 15 auf Null verlassen werden sollten.

- Wenn die Transaktion, die die Nachricht gesendet hat, nicht von OTMA stammt, müssen Sie den Clientnamen auf einen gültigen OTMA-Client setzen. Hierbei handelt es sich um den XCF-Member-Namen des IBM MQ-Warteschlangenmanagers, an den die Nachricht gesendet werden soll. Sie sollten Ihren Clientnamen (im Parameter OTMACON des Makros CSQ6SYSP ) auf den Namen des Warteschlangenmanagers setzen. Dies ist die Standardeinstellung. Anschließend sollte das DFSYPRX0-Einstellungsregister 15 bis 4 verlassen werden.
- Wenn die Transaktion, die die Nachricht sendet, von OTMA stammt und das Ziel nicht OTMA ist, sollten Sie die Register 15 bis 8 festlegen und den Vorgang beenden.
- In allen anderen Fällen sollte das Register 15 auf Null gesetzt werden.

Wenn Sie den OTMA-Clientnamen auf einen Namen setzen, der IMS nicht bekannt ist, gibt der CHNG oder ISRT-Aufruf Ihrer Anwendung einen A1-Statuscode zurück.

Für ein IMS-System, das mit mehr als einem IBM MQ-Warteschlangenmanager kommuniziert, sollten Sie die Logik für jeden WS-Manager von IBM MQ wiederholen.

Beispiel für Assemblercode wird in Abbildung 123 auf Seite 1098 angezeigt:

```
TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTIME
SPACE 2
LR  R12,R15        MODULE ADDRESSABILITY
USING DFSYPRX0,R12
*
L   R2,12(,R1)      R2 -> OTMA PREROUTE PARMS
*
LA  R3,48(,R2)      R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC  0(16,R3),=XL16'00'  OTMA ORIG?
BNE  OTMAIN        YES, GO TO THAT CODE
*
NOOTMAIN DS 0H            NOT OTMA INPUT
LA  R5,8(,R2)       R5 IS AT THE DESTINATION NAME
CLC  0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE  EXIT0         NO, NORMAL PROCESSING
*
L   R4,80(,R2)      R4 AT ADDR OF OTMA CLIENT
MVC  0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B   EXIT4         AND EXIT
*
OTMAIN  DS 0H           OTMA INPUT
LA  R5,8(,R2)       R5 IS AT THE DESTINATION NAME
CLC  0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE  EXIT8         NO, NORMAL PROCESSING
```

```
*
EXIT0  DS 0H
LA  R15,0         RC = 0
B   BYEBYE
*
EXIT4  DS 0H
LA  R15,4         RC = 4
B   BYEBYE
*
EXIT8  DS 0H
LA  R15,8         RC = 8
B   BYEBYE
*
BYEBYE  DS 0H
RETURN (14,12),,RC=(15)   RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END
```

*Abbildung 123. Assembler-Beispiel für OTMA-Exit vor Routing*

## z/OS Benutzerexit für Zielauflösung

Dieser Abschnitt enthält ein Beispiel für einen Zielauflösungs-Benutzerexit für IMS.

Wenn Sie die Register 15 bis 4 in DFSYPRX0 festgelegt haben oder wenn die Quelle der Transaktion OTMA ***und*** Sie Register 15 auf Null gesetzt haben, wird Ihr DRU-Exit aufgerufen. In diesem Beispiel ist der DRU-Exitname DRU0VCPE.

Der DRU-Exit prüft, ob das Ziel VCPEVCPE ist. Ist dies der Fall, werden die OTMA-Benutzerdaten (in dem OTMA-Präfix) wie folgt festgelegt:

**Offset**
**OTMA-Benutzerdaten**

**(dezimal)**

**0**
OTMA-Benutzerdatenlänge (in diesem Beispiel 334)

**2**
MQMD

**326**
Auf Format antworten

In diesen Offsets erwartet die IBM MQ-IMS-Brigde diese Informationen zu finden.

Der DRU-Exit sollte so einfach wie möglich sein. Daher werden in diesem Beispiel alle Nachrichten, die von IMS für einen bestimmten IBM MQ-Warteschlangenmanager stammen, in dieselbe IBM MQ-Warteschlange eingereiht.

Wenn die Nachricht persistent sein muss, muss IMS eine synchronisierte Transaktionspipe verwenden. Dazu muss der DRU-Exit die OUTPUT-Markierung setzen. Weitere Informationen finden Sie unter Synchronisierte Transaktionspipes für IBM MQangeben .

Schreiben Sie eine IBM MQ-Anwendung, um diese Warteschlange zu verarbeiten, und verwenden Sie Informationen aus der MQMD-Struktur, der MQIIH-Struktur (falls vorhanden) oder den Benutzerdaten, um die einzelnen Nachrichten an ihr Ziel weiterzuleiten.

Ein Beispielassembler-DRU-Exit wird in Abbildung 124 auf Seite 1100 angezeigt.

```
TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTIME
SPACE 2
LR  R12,R15            MODULE ADDRESSABILITY
USING DRU0VCPE,R12
*
L   R2,12(,R1)           R2 -> OTMA DRU PARMS
*
L   R5,88(,R2)           R5 ADDR OF OTMA USERDATA
LA  R6,2(,R5)           R6 ADDR OF MQMD
USING MQMD,R6            AS A BASE
*
LA  R4,MQMD_LENGTH+10       SET THE OTMA USERDATA LEN
STH  R4,0(,R5)           = LL + MQMD + 8
*                       CLEAR REST OF USERDATA
MVI  0(R6),X'00'          ...NULL FIRST BYTE
MVC  1(255,R6),0(R6)        ...AND PROPAGATE IT
MVC  256(MQMD_LENGTH-256+8,R6),255(R6) ...AND PROPAGATE IT
*
VCPE   DS  0H
CLC  44(16,R2),=CL16'VCPE'     IS DESTINATION VCPE?
BNE  EXIT4             NO, THEN DEST IS NON-OTMA
MVC  MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC  MQMD_REPLYTOQMGR,=CL48'VCPE'  SET QNAME AND QMGRNAME
MVC  MQMD_FORMAT,MQFMT_IMS     SET MQMD FORMAT NAME
MVC  MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*                       SET REPLYTO FORMAT NAME
B   EXIT0
*
EXIT0  DS  0H
LA  R15,0             SET RC TO OTMA PROCESS
B   BYEBYE             AND EXIT
*
EXIT4  DS  0H
LA  R15,4             SET RC TO NON-OTMA
B   BYEBYE             AND EXIT
*
BYEBYE  DS  0H
RETURN (14,12),,RC=(15)       RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA  EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END
```

*Abbildung 124. Beispiel-Assembler-DRU-Exit*

## z/OS Verwendung von IBM z/OSMF zur Automatisierung von IBM MQ

IBM z/OS Management Facility (z/OSMF) stellt Systemverwaltungsfunktionen in einer taskorientierten, browserbasierten Webbenutzerschnittstelle mit integrierter Benutzerunterstützung zur Verfügung, sodass Sie die Tagesoperationen und die Verwaltung Ihrer Großrechner-z/OS-Systeme einfacher verwalten können.

Durch die Optimierung einiger traditioneller Tasks und die Automatisierung anderer Tasks kann z/OSMF dazu beitragen, einige Bereiche des z/OS-Systemmanagements zu vereinfachen.

Ressourcen können von einem vom Benutzer bereitgestellten Portal auf Knopfdruck bereitgestellt oder mit einem Klick auf eine Schaltfläche bereitgestellt werden. z/OSMF stellt REST-APIs bereit, die Sie bei dieser Task unterstützen.

Das mit z/OSMF bereitgestellte Muster-Marktplatzportal kann auch zur Bereitstellung und Entbereitung von Ressourcen verwendet werden. Alternativ dazu können erfahrene Benutzer die z/OSMF-Webbenutzerschnittstelle (WUI) verwenden.

In diesem Abschnitt wird vorausgesetzt, dass Sie z/OSMFverstehen, aber wenn Sie mit z/OSMF nicht vertraut sind, sollten Sie Erste Schritte mit z/OSMF lesen. Alternativ können Sie in der Onlinehilfe von z/OSMF WUI auf diesen Abschnitt zugreifen.

Sie sollten sich mit der z/OS Cloud-Konfiguration vertraut machen, d. h.:

- Cloud-Bereitstellung- Resource Management Services
- Workload Management-Weitere Informationen finden Sie im Handbuch IBM z/OS Management Facility Programming Guide .
- Erste Schritte-siehe *Lernprogramm 'Einführung'-Cloud*

In z/OSMF 2.2 werden rollenbasierte Aktivitäten und Tasks eingeführt. Daher ist es wichtig, dass Sie Konzepte wie die folgenden verstehen:

Domänen
-Administratoren
Genehmiger
Tenants
Schablonen
Instanzen
Workflows

und so weiter.

Beispiele für IBM MQ z/OSMF-Workflows und zugehörige Dateien werden bereitgestellt und können als Teil der Komponenten von IBM MQ for z/OS UNIX System Services Components installiert werden. Der Installationsprozess für diese Funktion sowie die Verzeichnis- und Dateistruktur werden im IBM MQ for z/OS-Programmverzeichnis beschrieben. Download-Links für die Programmverzeichnisse finden Sie unter IBM MQ for z/OS Program Directory PDF files.

Die Beispielworkflows werden in XML geschrieben und veranschaulichen, wie die Bereitstellung (Erstellung) oder die Aufhebung der Bereitstellung (Vernichtung) von IBM MQ-Warteschlangenmanagern, Kanalinitiatoren und lokalen Warteschlangen automatisiert wird und wie Aktionen für die bereitgestellten IBM MQ-Ressourcen ausgeführt werden können. Schritte in den Workflows zum Übergeben von Jobs (JCL), zum Ausführen von REXX-Execs, zum Verarbeiten von Shell-Scripts oder zum Absetzen von REST API-Aufrufen.

Die Beispiele sollen die Typen von Funktionen veranschaulichen, die mit z/OSMF erreicht werden können. Es wird erwartet, dass z/OSMF-Workflows im Allgemeinen zum Bereitstellen von Ressourcen und Aktionen wie put- oder get-Nachrichten verwendet werden, die im Wesentlichen mithilfe von IBM MQ-Anwendungen ausgeführt werden.

Sie können die Beispielworkflows wie angegeben ausführen, vorausgesetzt, die Eigenschaften der Workflowvariablen wurden festgelegt (wie in den folgenden Abschnitten beschrieben), oder Sie können sie nach Bedarf anpassen. Sie können es vorziehen, eigene Workflows zu schreiben, um zusätzliche Funktionen auszuführen. Informationen zur Ausführung der Beispielworkflows finden Sie unter:

- „Voraussetzungen für z/OSMF” auf Seite 1102
- „Sicherheitseinstellungen” auf Seite 1103
- „Einschränkungen” auf Seite 1106

Beispielworkflowanwendungen werden bereitgestellt für:

- „Automatisieren Sie die Bereitstellung oder Löschung von IBM MQ-Warteschlangenmanagern und führen Sie Aktionen für die bereitgestellten Warteschlangenmanager aus” auf Seite 1107
- „Automatisieren Sie die Bereitstellung oder Löschung von lokalen IBM MQ-Warteschlangen und führen Sie Aktionen für die bereitgestellten Warteschlangen aus” auf Seite 1108.

**Zugehörige Konzepte**

„IBM MQ for z/OS einrichten” auf Seite 972

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

## z/OS Voraussetzungen für z/OSMF

Die Voraussetzungen, die Sie für die Ausführung von IBM z/OS Management Facility (z/OSMF) mit IBM MQ benötigen

Die in IBM MQ for z/OS 9.1.0 gelieferten Workflows nutzen die neue Funktion in z/OSMF, die über APARs sowohl in z/OS 2.1 als auch in Version 2.2 bereitgestellt wird. Weitere Einzelheiten finden Sie im folgenden Text.

1. Sie haben IBM z/OS Management Facility 2.2 ordnungsgemäß installiert und konfiguriert. Wenn Sie die Sicherheit für die Ausführung aktiviert haben, müssen Sie sicherstellen, dass alle Sicherheitseinstellungen, die von z/OSMF dokumentiert wurden, konfiguriert wurden.
2. Sie haben die folgenden APARs für installiert:

   **z/OS 2.1**

   - PI71068
   - PI71079
   - PI71082
   - PI71084
   - OA50130

   **z/OS 2.2**

   - PI70526
   - PI70521
   - PI70527
   - PI67839
   - PI70767
   - PI46315
   - OA49081
   - OA49802
   - OA50130
3. Die Angel-Prozesse (falls erforderlich) sowie die Serverprozesse der z/OSMF wurden konfiguriert.
4. Die z/OS-Cloudumgebung wurde konfiguriert (wie kurz oben erläutert und durch z/OSMF dokumentiert)
5. IBM MQ for z/OS 9.0.1 wurde installiert und die Bibliotheken mit den Produktlademodulen sind verfügbar.
6. Die folgenden Anpassungstasks für IBM MQ-Warteschlangenmanager wurden durchgeführt:

   | Task | Beschreibung |
   |---|---|
   | 1 | Angeben der z/OS-Systemparameter |
   | 2 | APF-Autorisierung der IBM MQ-Ladebibliotheken |
   | 3 | Aktualisieren der z/OS-Linkliste und LPA |
   | 4 | Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften |
7. Die Beispielworkflows und die zugehörigen Dateien sind in einem geeigneten Verzeichnis für z/OS UNIX System Services (z/OS UNIX) installiert.

8. Das Verzeichnis `/tmp` z/OS UNIX ist verfügbar, da der Workflow 'provision.xml' möglicherweise eine temporäre Datei in diesem Verzeichnis erstellt. Wenn eine Datei erstellt wird, löscht der Workflow im Allgemeinen die Datei nach der Verwendung.
9. Die Datei `deprovision.xml` enthält Schritte in dieser Datei, die die REXX-Execs CSQ4ZWS1.rexx und CSQ4ZWS2.rexx aufrufen. Diese Execs warten auf den Stopp des Warteschlangenmanagers und der Kanalinitiatorsubsysteme; die Execs rufen den z/OS UNIX-Befehl **SLEEP** als Systemaufruf auf.

   Abhängig von Ihrer z/OS UNIX-Konfiguration stellen Sie möglicherweise fest, dass der Befehl **SLEEP** nicht als codiert funktioniert. Wenn bei der Verarbeitung ein Fehler auftritt, der anzeigt, dass der Befehl **SLEEP** nicht gefunden wurde, können Sie versuchen, die folgenden Zeilen in den Execs CSQ4ZWS1.rexx und CSQ4ZWS2.rexx zu ersetzen:

```
CALL SYSCALLS('ON')            /* Enable z/OS UNIX calls    */
ADDRESS SYSCALL
"SLEEP" 10                          /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF'            /* Disable z/OS UNIX calls    */
```

   durch

```
'sleep' 10
```

   Setzen Sie dann den Befehl Open MVS (OMVS) **env** ab, um die Einstellung der Umgebungsvariablen PATH zu überprüfen. Stellen Sie sicher, dass das Verzeichnis, das den Befehl **`sleep`** enthält, für PATH definiert ist. Beachten Sie, dass sich der Befehl **`sleep`** normalerweise im Verzeichnis `/bin` befindet.
10. Stellen Sie sicher, dass z/OSMF gestartet wurde.

    Die Angel- und Serverprozesse der z/OSMF müssen gestartet worden sein und die z/OSMF Web User Interface (WUI) muss betriebsbereit sein. Weitere Informationen hierzu finden Sie im Abschnitt Liberty-Profil: Prozesstypen unter z/OS .

    Selbst wenn Sie die Workflows mit der REST API steuern möchten, muss die z/OSMF WUI gestartet werden. Die z/OSMF WUI kann nützlich sein, um die Erstellung und Ausführung von Workflows zu überwachen.

**Zugehörige Konzepte**

„Verwendung von IBM z/OSMF zur Automatisierung von IBM MQ” auf Seite 1100
IBM z/OS Management Facility (z/OSMF) stellt Systemverwaltungsfunktionen in einer taskorientierten, browserbasierten Webbenutzerschnittstelle mit integrierter Benutzerunterstützung zur Verfügung, sodass Sie die Tagesoperationen und die Verwaltung Ihrer Großrechner-z/OS-Systeme einfacher verwalten können.

## z/OS Sicherheitseinstellungen

Die Sicherheitseinstellungen, die für die Ausführung von z/OSMF erforderlich sind.

Die folgenden Eigenschaften für die Benutzer-ID-Variable sind in der Eigenschaftendatei definiert. Weitere Einzelheiten finden Sie unter „Workflows ausführen” auf Seite 1111.

| Benutzer-ID, Eigenschaft | Beschreibung |
|---|---|
| CSQ_USERID | Die Benutzer-ID, die zum Ausführen der Workflowschritte verwendet wird. Beachten Sie jedoch, dass ausgewählte Schritte (die in der Regel eine höhere Berechtigungsstufe erfordern) auf der Basis der Einstellung der **`CSQ_ADMIN_*`** -Benutzer-IDs, die im folgenden Text aufgeführt sind, mit unterschiedlichen Benutzer-IDs ausgeführt werden. Die Benutzer-ID, die verwendet wird, wird durch die Eigenschaft **`runAsUser`** auf dem jeweiligen Schritt in den Workflows angegeben. |
| CSQ_ADMIN_APF_USERID | Die Benutzer-ID, die verwendet werden soll, wenn APF die Ladebibliothek autorisiert, die das Systemparametermodul des Warteschlangenmanagers enthält. |

| Benutzer-ID, Eigenschaft | Beschreibung |
|---|---|
| CSQ_APF_APPROVAL_ID | Die Genehmigungs-ID, die es Benutzern ermöglicht, den APF-Berechtigungsschritt der Datei als Benutzer CSQ_ADMIN_APF_USERID auszuführen. |
| CSQ_ADMIN_CONSOLE_USERID | Die Benutzer-ID, die bei der Ausführung von Schritten unter der Ausführung verwendet wird, die z/OS-Konsol<br>**Achtung:** Dieser Benutzer-ID muss UPDATE-Zugriff auf das Profil der gestarteten Task (MVS.START.STC. *) erteilt werden. in der Klasse OPERCMDS. Weitere Informationen finden Sie unter Verwendung von Bedienerbefehlen steuern in der Dokumentation zu z/OS . |
| CSQ_CONSOLE_APPROVAL_ID | Die Genehmigungs-ID, mit der Benutzer Schritte ausführen können, die z/OS-Konsolbefehle bei der Ausführung als Benutzer CSQ_ADMIN_CONSOLE_USERID absetzen. |
| CSQ_ADMIN_SAF_USERID | Benutzer-ID, die beim Absetzen von SAF-Befehlen verwendet werden soll |
| CSQ_SAF_APPROVAL_ID | Die Genehmigungs-ID, die es Benutzern ermöglicht, die SAF-Befehlsschritte unter der Ausführung als Benutzer CSQ_ADMIN_SAF_USERID auszuführen. |
| CSQ_ADMIN_SSI_USERID | Die Benutzer-ID, die beim Absetzen des Befehls SETSSI zum Identifizieren des Subsystems verwendet werden soll, das für z/OS bereitgestellt wird. |
| CSQ_SSI_APPROVAL_ID | Die Genehmigungs-ID, mit der Benutzer den SETSSI-Befehlsschritt unter der Ausführung als Benutzer CSQ_ADMIN_SSI_USERID ausführen können. |

**Anmerkung:** Die Benutzer-ID, die zum Ausführen der Bereitstellungsworkflows und -bereitstellung verwendet wird, muss über eine ausreichende Berechtigung verfügen, wie im Folgenden aufgelistet:

1. Mit dem Befehl SETPROG können die Befehle SETPROG und APF für den Befehl SETPROG in den Warteschlangen für die Bereitstellung und den Entbereitungs Entweder ist die Benutzer-ID in der Eigenschaft CSQ_ADMIN_APF_USERID festgelegt, oder die Benutzer-ID, die zum Ausführen der Workflows verwendet wird, muss zum Ausführen dieses Befehls berechtigt sein. Sie können dies erreichen, indem Sie den folgenden Befehl ausgeben:

```
PERMIT MVS.SETPROG CLASS(OPERCMDS) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

   **Anmerkung:** Der Befehl SETPROG bleibt möglicherweise nach einem IPL eines z/OS-Systems nicht erhalten. Daher kann es erforderlich sein, den folgenden SETPROG-Befehl nach einem IPL manuell auszugeben:

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

   Weitere Informationen zum Befehl SETPROG finden Sie unter Verwenden von RACF zur Steuerung von APF-Listen.

   Darüber hinaus haben Sie möglicherweise die FACILITY-Klasse aktiviert, um zu steuern, welche Bibliotheken APF berechtigt sind. Daher müssen Sie den folgenden Befehl eingeben:

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID)
       ACCESS(UPDATE)
```

2. Ein Schritt im Bereitstellungsworkflow für den Warteschlangenmanager gibt den Befehl SETSSI aus, um das IBM MQ-Subsystem mit z/OS zu identifizieren. Die Benutzer-ID, die in der Eigenschaft

CSQ_ADMIN_SSI_USERID festgelegt ist, muss für die Verwendung dieses Befehls berechtigt sein. Sie können dies erreichen, indem Sie den folgenden Befehl ausgeben:

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMDS) ID(value of CSQ_ADMIN_SSI_USERID)
        ACCESS(CONTROL)
```

**Anmerkung:** Über den Befehl SETSSI für z/OS identifizierte Subsysteme bleiben nach einem IPL eines z/OS-Systems nicht erhalten. Daher kann es erforderlich sein, den folgenden SETSSI-Befehl nach einem IPL manuell auszugeben:

```
SETSSI  ADD,S='value of CSQ_SSID',I=CSQ3INI,
       P='CSQ3EPX,value of CSQ_CMD_PFX,S'
```

Weitere Informationen zum Befehl SETSSI finden Sie im Abschnitt SETSSI-Befehl.

3. Die Workflows setzen WS-Manager-Befehle ab. Wenn Sie also die Sicherheit aktivieren möchten, muss die in der Eigenschaft CSQ_ADMIN_RACF_USERID (oder die Benutzer-ID, die zum Ausführen der Workflows verwendete Benutzer-ID verwendet wird) die Berechtigung CLAUTH (Clientauthentifizierung) für die Klasse MQADMIN oder die Klasse MXADMIN (abhängig von der verwendeten Klasse) erteilt werden. Dies ist die Möglichkeit, diese Benutzer-ID zuzulassen, um Sicherheitsprofile für diese Klassen zu definieren. Sie können dies erreichen, indem Sie den folgenden Befehl ausgeben:

   ```
   ALTUSR value of CSQ_ADMIN_RACF_USERID CLAUTH(MQADMIN)
   ```

   Weitere Informationen zu **CLAUTH** finden Sie unter Attribut CLAUTH (Klassenberechtigung).
4. Der Workflow deprovision.xml gibt z/OS-Befehle aus, z. B. DISPLAY ACTIVE (für Jobs), CANCEL oder FORCE (für Subsysteme), so dass die Benutzer-ID, die in der Eigenschaft CSQ_ADMIN_CONSOLE_USERID festgelegt ist (oder die Benutzer-ID, die zum Ausführen der Workflows verwendet wird), über die geeignete Berechtigung zum Absetzen solcher Befehle verfügen muss.
5. Benutzer, die eine Warteschlangenmanagerinstanz mit Hilfe der Schablonentabelle der Task "Software-Services" anfordern, müssen über die Berechtigung für den Zugriff auf z/OSMF und den Konfigurationsassistenten, wie in z/OSMF definiert, verfügen.
6. Die Benutzer-ID der Konsumentenbereitstellung für einen Warteschlangenmanager ist berechtigt, Teildateien aus dem PROCLIB-Datensatz hinzuzufügen und zu löschen, die mit der Variablen CSQ_PROC_LIB definiert sind.
7. Vor Bereitstellungswarteschlangen muss ein Warteschlangenmanager bereitgestellt werden.
8. Um die Workflows `queueLoad.xml` und `queueOffload.xml` verwenden zu können, müssen die verwendeten Datensätze vorzeitig definiert werden. Außerdem muss die Benutzer-ID, die für die Ausführung dieser Workflows verwendet wird, über die Berechtigung UPDATE für die Dateien erteilt werden.
9. Ein Schritt im `provision.xml`-Workflow für Warteschlangenmanager inaktiviert die Subsystemsicherheit. Sie können Job `csq4znse.jcl` ändern, um die Subsystemsicherheit zu aktivieren, indem Sie die entsprechenden Sicherheitsbefehle zum Schützen von IBM MQ-Ressourcen hinzufügen. Beachten Sie jedoch, dass Sie, wenn Sie zusätzliche Befehle hinzufügen, auch Befehle zum Löschen von Sicherheitsberechtigungen in `csq4dse.jcl` hinzufügen müssen, die vom Workflow deprovision.xml übergeben werden.

   **Anmerkung:** In diesem Schritt werden RACF-Sicherheitsbefehle absetzt. Wenn Sie ein alternatives Sicherheitsprodukt verwenden, müssen Sie diesen Schritt ändern, um die entsprechenden Befehle für Ihr Sicherheitsprodukt auszugeben.

## Netzanforderungen

Wenn Sie eine WS-Manager-Schablone und Ressourcen für die Schablone hinzufügen, müssen Sie auf **Netzressourcenpool erstellen** klicken. Dadurch wird ein Ressourcenpool mit Netzressourcen für diese Schablone erstellt.

Wenn Sie den Konfigurationsassistenten verwenden, muss Ihr Netzadministrator diese Definition des Netzressourcenpools ausführen, indem Sie eine Begrenzung für die Anzahl der Ports definieren, die für diese Vorlage zugeordnet werden sollen.

Für jede Vorlageninstanz ordnet der `provision.xml`-Workflow einen Port in dem Bereich zu und startet eine Empfangsfunktion, die an diesem Port empfangsbereit ist.

## Klassifizieren mit IBM Workload Manager

Wenn Sie den Warteschlangenmanager und die Adressräume des Kanalinitiators mit WLM klassifizieren möchten, müssen Sie dies beim Hinzufügen einer Vorlage für die Bereitstellung eines Warteschlangenmanagers angeben.

Ob klassifiziert werden soll, wird durch die Flags **`CSQ_DEFINE_MSTR_WLM_RULE`** und **`CSQ_DEFINE_CHIN_WLM_RULE`**gesteuert, die in der Datei `workflow_variables.properties`festgelegt sind.

Weitere Informationen zum Klassifizieren mit WLM finden Sie im Handbuch *z/OSMF Configuration Guide* .

**Zugehörige Konzepte**

„Voraussetzungen für z/OSMF” auf Seite 1102
Die Voraussetzungen, die Sie für die Ausführung von IBM z/OS Management Facility (z/OSMF) mit IBM MQ benötigen

## z/OS Einschränkungen

Einschränkungen bei der Verwendung von z/OSMF mit IBM MQ.

1. Der `provision.xml`-Workflow automatisiert derzeit die folgenden hervorgehobenen Anpassungstasks für Warteschlangenmanager:

| Task | Beschreibung |
|---|---|
| 1 | Angeben der z/OS-Systemparameter |
| 2 | APF-Autorisierung der IBM MQ-Ladebibliotheken **(provision.xml führt APF-Berechtigung für einige Bibliotheken durch)** |
| 3 | Aktualisieren der z/OS-Linkliste und LPA |
| 4 | Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften |
| 5 | **Definieren des IBM MQ-Subsystems für z/OS** |
| 6 | **Erstellen von Prozeduren für den IBM MQ-Warteschlangenmanager** |
| 7 | **Prozeduren für den Kanalinitiator erstellen** |
| 8 | **Definieren des IBM MQ-Subsystems für eine z/OS-WLM-Serviceklasse** |
| 9 | Wählen Sie die Coupling Facility-Offload-Speicherumgebung aus, und konfigurieren Sie |
| 10 | Einrichten der Coupling-Facility |
| 11 | Implementieren Sie Ihre ESM-Sicherheitskontrollen. |
| 12 | SYS1.PARMLIB-Teildateien aktualisieren |
| 13 | **Eingabedatengruppen für die Initialisierung anpassen** |
| 14 | **Bootstrap-und Protokolldatengruppen erstellen** |
| 15 | **Seitengruppen definieren** |
| 16 | Hinzufügen der IBM MQ-Einträge zur Db2-Gruppe mit gemeinsamer Datennutzung |

| Task | Beschreibung |
|---|---|
| 17 | **Anpassen der Systemparametermodule (einige)** |
| 18 | **Kanalinitiatorparameter anpassen (einige)** |
| 19. | Batch-, TSO-und RRS-Adapter konfigurieren |
| 20 | Konfigurieren Sie die Operationen und Steuerkonsolen. |
| 21 | Teildatei für die Speicherauszugsformatierung in IBM MQ einschließen |
| 22 | Informationsnachrichten unterdrücken |
| 23 | Aktualisieren Ihres System-DIAG-Members für Advanced Message Security |
| 24 | Prozeduren für Advanced Message Security erstellen |
| 25 | Richten Sie die gestartete Task Benutzer Advanced Message Security ein. |
| 26 | Dem Sicherheitsadministrator RACDCERT-Berechtigungen für Advanced Message Security erteilen |
| 27 | Benutzern Ressourcenberechtigungen für Advanced Message Security erteilen |

2. Angepasste Anpassungstasks, die nicht in Fettschrift hervorgehoben sind, müssen bei Bedarf manuell ausgeführt werden.
3. Die Member INP1 und INP2 werden derzeit als Beispiel verwendet. Falls erforderlich, können zusätzliche Eigenschaften definiert werden, um die Ressourcen zu steuern, die von diesen Membern definiert werden.
4. Kommentare, die sich auf bestimmte Eigenschaften beziehen, die in der Eigenschaftendatei aufgeführt sind, geben alle Einschränkungen für die Verwendung dieser Eigenschaften an. Weitere Einzelheiten finden Sie unter „Workflows ausführen” auf Seite 1111.

**Zugehörige Konzepte**

„Sicherheitseinstellungen” auf Seite 1103
Die Sicherheitseinstellungen, die für die Ausführung von z/OSMF erforderlich sind.

## z/OS Bereitstellung von IBM MQ-Objekten automatisieren

Es werden Muster bereitgestellt, um die Bereitstellung von Warteschlangenmanagern und lokalen Warteschlangen zu automatisieren.

### Automatisieren Sie die Bereitstellung oder Löschung von IBM MQ-Warteschlangenmanagern und führen Sie Aktionen für die bereitgestellten Warteschlangenmanager aus

Die folgenden z/OSMF-Beispielworkflows für Warteschlangenmanager werden bereitgestellt:

| Workflowname | Beschreibung |
|---|---|
| provision.xml | Bereitstellung eines IBM MQ for z/OS-Warteschlangenmanagers<br>Dieser Beispielworkflow:<br>• Stellt die erforderlichen Systemressourcen für einen Warteschlangenmanager bereit.<br>• Stellt die erforderlichen Systemressourcen für einen Kanalinitiator bereit.<br>• Startet den Warteschlangenmanager (der auch den Kanalinitiator und den TCP/IP-Listener startet) |

| Workflowname | Beschreibung |
|---|---|
| | • Führt das Prüfprogramm für die Installation des Beispielwarteschlangenmanagers aus.<br>Eine Umgebungseigenschaft kann festgelegt werden, um die Bereitstellung von Warteschlangenmanagern mit unterschiedlichen Merkmalen zu steuern. Weitere Informationen finden Sie unter „Workflows ausführen" auf Seite 1111.<br>**Anmerkung:** Es wird eine Manifestdatei (`provision.mf`) bereitgestellt, die beim Hinzufügen einer Schablone für diesen Workflow unterstützt wird. Diese Datei enthält einen Verweis auf die Datei **qaas_readme.pdf** , die zusätzliche Informationen enthält. Sie können auf die Datei über einen Link zugreifen, sobald die Vorlage hinzugefügt wurde. |
| deprovision.xml | Löschung eines IBM MQ for z/OS-Warteschlangenmanagers<br>Dieser Beispielworkflow:<br>• Stoppt den Kanalinitiator (der auch den TCP/IP-Listener stoppt) und den Warteschlangenmanager.<br>• Wartet auf die zu stoppenden Subsysteme.<br>• Löscht alle Kanalinitiator- und Warteschlangenmanager-Systemressourcen. |
| startQMgr.xml | Start eines IBM MQ for z/OS-Warteschlangenmanagers<br>Dieser Beispielworkflow startet den Warteschlangenmanager (der auch den Kanalinitiator und den TCP/IP-Listener startet). |
| stopQMgr.xml | Stopp eines IBM MQ for z/OS-Warteschlangenmanagers<br>Dieser Beispielworkflow stoppt den Kanalinitiator (der auch den TCP/IP-Listener stoppt) und den Warteschlangenmanager. |

Jeder Workflow führt einen oder mehrere Schritte aus. Kommentare in den Workflows erläutern die Funktion, die von jedem Schritt ausgeführt wird. In einigen der Schritte wird nur die Dateneingabe angefordert, während einige Schritte JCL übergeben, REXX-Execs aufrufen, Shell-Scripts aufrufen oder REST API-Aufrufe ausführen, um die angegebene Funktion auszuführen.

Den exakten Namen der JCL-oder REXX-Exec-Dateien finden Sie in den einzelnen Schritten. Die Workflows und die zugehörigen JCL-oder REXX-Exec-Dateien verweisen auf Variablen, die in einer oder mehreren Variablen-XML-Dateien deklariert sind. Weitere Einzelheiten finden Sie unter „Workflowvariablendeklarationsdateien" auf Seite 1111.

**`deprovision`**, **`startQMgr`** und **`stopQMgr`** können als Aktionen für einen bereitgestellten IBM MQ for z/OS-Warteschlangenmanager ausgeführt werden.

## Automatisieren Sie die Bereitstellung oder Löschung von lokalen IBM MQ-Warteschlangen und führen Sie Aktionen für die bereitgestellten Warteschlangen aus

Die folgenden z/OSMF-Beispielworkflows für Warteschlangen werden bereitgestellt:

| Workflowname | Beschreibung |
|---|---|
| defineQueue.xml | Definieren Sie eine lokale Warteschlange.<br>Dieser Beispielworkflow veranschaulicht, wie z/OSMF-Workflows verwendet werden können, um kleine, mittlere oder große Warteschlangen auf der Basis von Eigenschafteneinstellungen zu definieren. |

| Workflowname | Beschreibung |
|---|---|
| | **Anmerkung:** Es wird eine Manifestdatei (`provision.mf`) bereitgestellt, die beim Hinzufügen einer Schablone für diesen Workflow unterstützt wird. Diese Datei enthält einen Verweis auf die Datei **qaas_readme.pdf** , die zusätzliche Informationen enthält. Sie können auf die Datei über einen Link zugreifen, sobald die Vorlage hinzugefügt wurde. |
| displayQueue.xml | Ausgewählte Attribute einer lokalen Warteschlange anzeigen<br><br>In diesem Beispielworkflow werden die ausgewählten Attribute einer lokalen Warteschlange angezeigt. Die Attribute werden in einer z/OSMF-Variablen zurückgegeben (siehe die Schritte im Workflow für den Namen der Variablen) und werden anschließend angezeigt. Falls erforderlich, kann auf den Inhalt der Variablen zugegriffen werden, indem ein REST API verwendet wird.<br><br>Weitere Informationen finden Sie im Artikel REST-APIs für Cloudbereitstellung. Weitere Informationen finden Sie unter z/OSMF-Workflowservices. |
| deleteQueue.xml | Lokale Warteschlange löschen<br><br>Dieser Beispielworkflow löscht eine lokale Warteschlange in einem angegebenen Warteschlangenmanager. |
| putQueue.xml | Fügen Sie eine oder mehrere Nachrichten in eine lokale Warteschlange ein.<br><br>Dieser Beispielworkflow reiht eine oder mehrere Nachrichten in eine lokale Warteschlange ein. Der Nachrichtentext kann zwar angegeben werden, aber wenn mehrere Nachrichten gleichzeitig in eine lokale Warteschlange gestellt werden, wird derselbe Nachrichtentext verwendet. |
| getQueue.xml | Ruft eine oder mehrere Nachrichten aus einer lokalen Warteschlange ab.<br><br>Dieser Musterworkflow ruft eine oder mehrere Nachrichten aus einer lokalen Warteschlange ab. Die Nachrichten werden in einer z/OSMF-Variablen zurückgegeben (siehe die Schritte im Workflow für den Namen der Variablen) und werden anschließend angezeigt. Falls erforderlich, können Sie mit einem REST API auf den Inhalt der Variablen zugreifen.<br><br>Weitere Informationen finden Sie im Artikel REST-APIs für Cloudbereitstellung. Weitere Informationen finden Sie unter z/OSMF-Workflowservices. |
| loadQueue.xml | Laden von Nachrichten aus einem Datensatz in eine lokale Warteschlange.<br><br>Dieser Musterworkflow lädt Nachrichten aus einer Datei, die in eine lokale Warteschlange gesetzt ist. Der Standardname der Datei wird durch Festlegen einer Eigenschaft angegeben. Weitere Einzelheiten finden Sie unter „Workflows ausführen” auf Seite 1111. |
| offloadQueue.xml | Versetzen von Nachrichten aus einer lokalen Warteschlange in eine Datei.<br><br>Bei diesem Beispielworkflow werden Nachrichten aus einer lokalen Warteschlange in einen Datensatz geladen. Der Standardname der Datei wird durch Festlegen einer Eigenschaft angegeben. Weitere Einzelheiten finden Sie unter „Workflows ausführen” auf Seite 1111. |
| clearQueue.xml | Nachrichten in einer lokalen Warteschlange löschen.<br><br>Dieser Beispielworkflow löscht (löscht) alle Nachrichten in einer lokalen Warteschlange. |

**Anmerkungen:**

1. Mit der Aktion **Aufnahmewarteschlange** können Sie einige Nachrichtendaten eingeben und eine oder mehrere Nachrichten in eine Warteschlange stellen. Wenn mehr als eine Nachricht während einer bestimmten Anforderung in eine Warteschlange gestellt werden soll, wird die gleiche Nachrichtendaten verwendet.
2. Die Workflows "loadQueue.xml" und "offloadQueue.xml" rufen das ausführbare Modul CSQUDMSG in der Bibliothek SCSQLOAD mit einem Aliasnamen von QLOAD auf. Dies entspricht der Dienstprogrammvariablen **dmpmqmsg** bei IBM MQ for Multiplatforms. Daher wird erwartet, dass Nachrichten aus einer Datei in eine Warteschlange oder aus einer Warteschlange in eine Datei im **dmpmqmsg** -Format geladen werden.

   Der JCL-Mustercode wird auch als Member CSQ4QLOD in SCSQPROC bereitgestellt.

   Die einfachste Möglichkeit, die Aktionen "loadQueue" und "offloadQueue" auszuprobieren, besteht darin, die folgenden Schritte zu tun:

   a. Geben Sie **putQueue** einige Male aus, um einige Nachrichten in eine Warteschlange einzureihen.
   b. Mit **offloadQueue** können Sie die Nachrichten aus der Warteschlange in eine Datei auslagern.
   c. Falls erforderlich, geben Sie **clearQueue** aus, um alle Nachrichten aus der Warteschlange zu entfernen.
   d. Verwenden Sie **loadQueue** , um die Nachrichten aus einer Datei in dieselbe oder eine andere Warteschlange zu laden.

   Wenn Sie am Format **dmpmqmsg** interessiert sind, können Sie den Inhalt der Datei durchsuchen, nachdem Sie eine Auslagerungsanforderung ausgegeben haben.
3. Sie können **displayQueue**, **deleteQueue**, **putQueue**, **getQueue**, **loadQueue**, **offloadQueue** und **clearQueue** als Aktionen für eine bereitgestellte lokale IBM MQ for z/OS-Warteschlange ausführen. Weitere Informationen zu Aktionen und Aktionsdateien finden Sie im Handbuch *z/OSMF Programming Guide* .
4. Alle aktionsbezogenen Workflows werden standardmäßig gelöscht. Der Grund dafür ist, dass die Benutzer die Workflows nicht bereinigen müssen.

   Das Problem besteht jedoch darin, dass bei einer Aktion in einer Ausgabe eine bestimmte Ausgabe entsteht. Die Aktionen **displayQueue** und **getQueue** erzeugen beispielsweise eine Ausgabe.

   Die Ausgabe kann nicht angezeigt werden, da der zugehörige Workflow gelöscht wird, sobald die Aktion ausgeführt wurde. Wenn Sie also die Workflowaktionen über die z/OS -Webbenutzerschnittstelle steuern, müssen Sie das Flag **cleanAfterComplete** im Tag **< workflow>** für jede Aktion, deren Ausgabe Sie anzeigen möchten, auf *false* setzen.

   Um beispielsweise die Ausgabe von **displayQueue**anzuzeigen, setzen Sie das Flag wie folgt:

```
<action name="displayQueue">
  <workflow cleanAfterComplete="false">
  ...

  ...
  </workflow>
</action>
```

   Dies bedeutet jedoch, dass Sie dann Aktionsworkflows manuell bereinigen müssen.

Jeder Beispielworkflow von z/OSMF führt einen oder mehrere Schritte aus. Kommentare in den Workflows erläutern die Funktion, die von jedem Schritt ausgeführt wird. Einige der Schritte fordern nur Dateneingabe an, während einige Schritte JCL übergeben und andere REXX-Execs aufrufen, um die angegebene Funktion auszuführen.

Den exakten Namen der JCL-oder REXX-Exec-Dateien finden Sie in den einzelnen Schritten. Die Workflows und die zugehörigen JCL-oder REXX-Exec-Dateien verweisen auf Variablen, die in einem oder mehreren „Workflowvariablendeklarationsdateien” auf Seite 1111 deklariert sind.

**Zugehörige Konzepte**

„Einschränkungen” auf Seite 1106

Einschränkungen bei der Verwendung von z/OSMF mit IBM MQ.

## z/OS Workflows ausführen

Eine Beschreibung der Dateien, auf die durch das Beispiel verwiesen wird. Die z/OSMF-Workflows und die Art und Weise, in der ein Workflow ausgeführt wird.

### Workflowvariablendeklarationsdateien

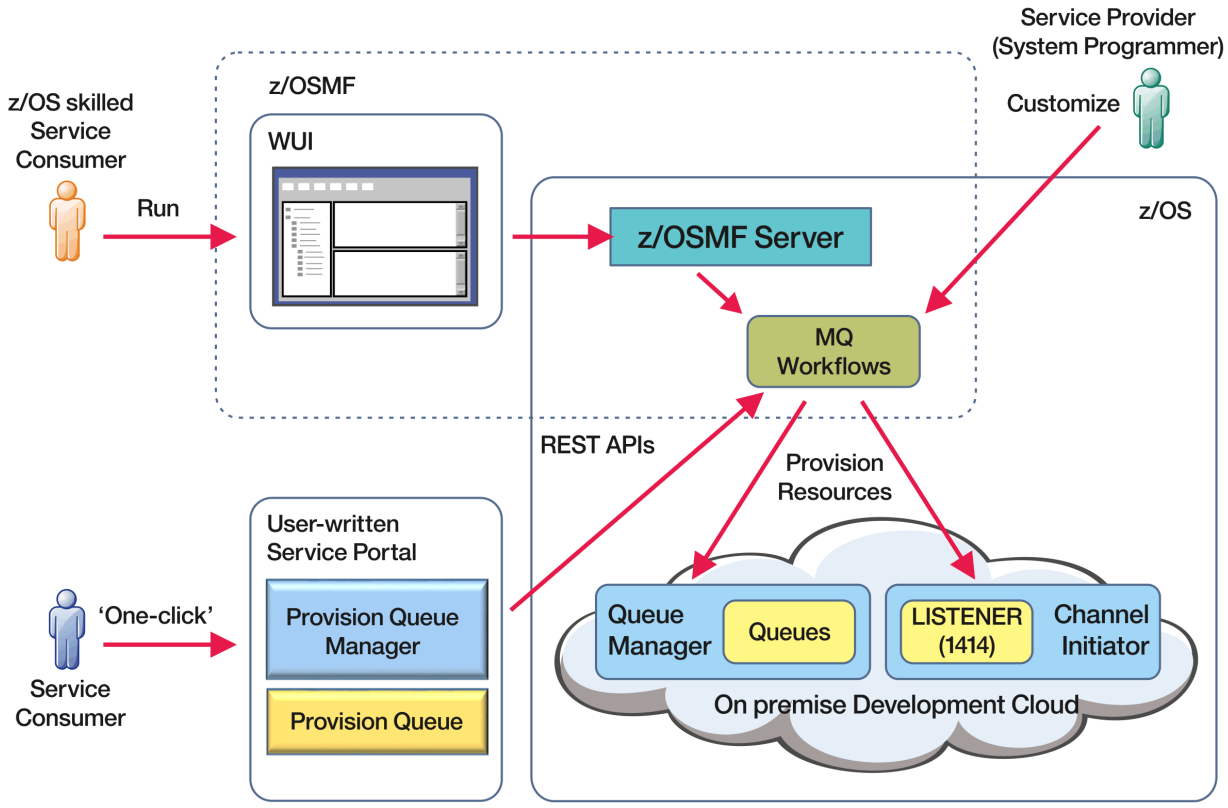
In den folgenden Dateien werden Variablen deklariert, die von den z/OSMF-Beispielworkflows und den zugehörigen JCL-oder REXX-Exec-Dateien referenziert werden:

| Name der Workflowvariablendeklarationsdatei | Beschreibung |
|---|---|
| common_variables.xml | Variablen, die sowohl für den Warteschlangenmanager (plus Kanalinitiator) als auch für die Warteschlangenworkflows gemeinsam sind. |
| qmgr_variables.xml | Variablen, die für die Workflows des Warteschlangenmanagers (plus Kanalinitiator) spezifisch sind. |
| queue_variables.xml | Spezifische Variablen für die Warteschlangenworkflows. |
| tcpip_variables.xml | Variablen, die für die Workflows des Warteschlangenmanagers (plus Kanalinitiator) spezifisch sind und für die Identifizierung von TCP/IP-Ressourcen verwendet werden. |

**Anmerkung:** Die Standardsichtbarkeit von Variablen ist *private*. Damit Variablen mithilfe der z/OSMF REST API abgefragt werden können, wurden die ausgewählten Variablen als *public* markiert. Sie können die Sichtbarkeit einer bestimmten Variablen jedoch bei Bedarf ändern.

### Workflows ausführen

*Abbildung 125. Bereitstellung von IBM MQ for z/OS-Ressourcen per Mausklick*




Bevor die Workflows ausgeführt werden können, müssen einige Eigenschaften in der folgenden Datei festgelegt werden:

| **Name der Eigenschaftendatei der Workflowvariablen** | **Beschreibung** |
|---|---|
| workflow_variables.properties | Anfängliche Eigenschaften für die Workflowvariablen. Kommentare in der Datei geben den Zweck der einzelnen Eigenschaften an.<br>• Eigenschaften in Meta-Klammern (< >) müssen auf benutzerspezifische Werte gesetzt werden.<br>• Eine Umgebungseigenschaft kann so eingestellt werden, dass Warteschlangenmanager für die Entwicklung (DEV) oder Test (TEST) oder Qualitätssicherungsumgebungen (Quality Assurance, QA) oder Produktionsumgebungen (PROD) bereitgestellt werden.<br>Zusätzliche Eigenschafteneinstellungen steuern die Merkmale des Warteschlangenmanagers, der für jede Umgebung bereitgestellt werden soll. Sie können z. B. die Anzahl der aktiven Protokolle oder die Anzahl der Seitengruppen für jeden Umgebungstyp ändern.<br>• Andere Eigenschaften sind auf IBM MQ-Standardwerte gesetzt, können aber bei Bedarf geändert werden, um die lokalen Konventionen zu erfüllen. |

Wenn die Eigenschaften festgelegt wurden, können die Workflows im Allgemeinen wie angegeben ausgeführt werden. Falls erforderlich, können Sie jedoch einen Workflow anpassen, um vorhandene Schritte zu ändern oder zu entfernen oder um neue Schritte hinzuzufügen.

Workflows können ausgeführt werden:

- Über die WUI von z/OSMF.

  Von Cloud Provisioning -> Software Services in der WUI können Workflows im Automatikmodus oder im manuellen Modus ausgeführt werden. Der manuelle Modus ist beim Testen nützlich, und in beiden Modi kann der Fortschritt jedes Schritts im Workflow überwacht werden.

  Weitere Informationen finden Sie unter Cloud-Bereitstellungsservices und Workflow erstellen.
- Verwenden Sie die REST-Workflow-Services von z/OSMF.

  Die REST-Workflow-Services können verwendet werden, um Workflows über eine REST API auszuführen. Dieser Modus ist nützlich für das Erstellen von Einmalklickoperationen aus einem benutzerdefinierten Portal.

  Weitere Informationen finden Sie im Artikel REST-APIs für Cloudbereitstellung. Weitere Informationen finden Sie unter z/OSMF-Workflowservices.
- Verwenden Sie das Marketplace-Musterportal, das mit z/OSMF bereitgestellt wird.

**Zugehörige Konzepte**

„Bereitstellung von IBM MQ-Objekten automatisieren" auf Seite 1107
Es werden Muster bereitgestellt, um die Bereitstellung von Warteschlangenmanagern und lokalen Warteschlangen zu automatisieren.

### z/OS MQ Adv. VUE MFT-Agentenkonnektivität zu fernen z/OS-Warteschlangenmanagern aktivieren

In einigen Fällen können Managed File Transfer -Agenten unter z/OS eine Verbindung zu einem fernen Warteschlangenmanager unter z/OS über eine Clientverbindung herstellen. Dies kann zu einfacheren IBM MQ Topologien führen.

Clientverbindungen zu fernen z/OS -Warteschlangenmanagern werden in den folgenden Fällen unterstützt:

- V 9.3.4 LTS Der MFT -Agent hat IBM MQ 9.3.4 oder höher oder Long Term Support mit angewendetem APAR PH56722 und wurde der Produkt-ID (PID) von IBM MQ Advanced for z/OS VUEoder IBM MQ Advanced for z/OSzugeordnet.
- Der MFT -Agent befindet sich unter IBM MQ 9.3.0 und wurde der PID IBM MQ Advanced for z/OS VUEzugeordnet.

Informationen zu den verschiedenen PIDs finden Sie unter IBM MQ -Produkt-IDs und Exportinformationen .

Informationen zum Festlegen der PID, die einer MFT -Installation zugeordnet ist, finden Sie unter **fteSetProductId** .

Die PID, unter der der Agent ausgeführt wird, wird im Protokoll beim Start des Agenten angezeigt.

Ein MFT -Agent unter z/OS, der unter einer anderen PID ausgeführt wird, kann nur über eine Verbindung im Bindungsmodus eine Verbindung zu einem lokalen Warteschlangenmanager herstellen.

Wenn ein Agent versucht, eine Verbindung zu einem Warteschlangenmanager herzustellen, der nicht unter z/OSausgeführt wird, wird die BFGQM1044E -Nachricht ausgegeben und der Agentenstart beendet.

**Zugehörige Tasks**

MFT-Agenten unter z/OS starten

## IBM MQ Internet Pass-Thru konfigurieren

In diesem Abschnitt werden die verschiedenen Funktionen beschrieben, die von IBM MQ Internet Pass-Thru (MQIPT) unterstützt werden, und wie diese konfiguriert werden.

Konfigurieren Sie MQIPT, indem Sie Änderungen an der Konfigurationsdatei `mqipt.conf` vornehmen. Die Struktur der MQIPT-Konfigurationsdatei und die Eigenschaften, die angegeben werden können, sind in der Referenz zur IBM MQ Internet Pass-Thru-Konfiguration beschrieben.

**Anmerkung:** Sie sollten sichere Dateiberechtigungen für das Verzeichnis festlegen, in dem sich die Datei `mqipt.conf` befindet, um zu verhindern, dass nicht berechtigte Benutzer gespeicherte Kennwörter sehen oder die Konfiguration ändern. Schützen Sie alle Kennwörter, die in der Konfigurationsdatei angegeben sind, indem Sie der Prozedur in „Gespeicherte Kennwörter in MQIPT verschlüsseln” auf Seite 1156 folgen.

Änderungen an der Konfigurationsdatei werden wirksam, wenn MQIPT gestartet oder aktualisiert wird. Durch das Aktualisieren einer aktiven Instanz von MQIPT werden Konfigurationsänderungen wirksam, ohne dass MQIPT erneut gestartet werden muss. Wenn MQIPT aktualisiert wird, wird die `mqipt.conf`-Konfigurationsdatei erneut gelesen, und MQIPT führt die folgenden Aktionen aus:

- Alle aktiven Routen, die als inaktiv gekennzeichnet oder in der Konfigurationsdatei nicht mehr angegeben sind, werden geschlossen und akzeptieren keine eingehenden Verbindungen mehr.
- Alle Routen, die in der Konfigurationsdatei als aktiv gekennzeichnet sind und derzeit nicht ausgeführt werden, werden gestartet.
- Alle Änderungen an den Konfigurationsparametern von aktiven Routen werden angewendet. Sofern möglich, werden diese Änderungen ohne eine Unterbrechung von aktiven Verbindungen wirksam. Bei einigen Parameteränderungen, beispielsweise bei einer Änderung des Routenziels, werden alle Verbindungen geschlossen, bevor die Änderung angewendet und die Route erneut gestartet wird.

Zum Aktualisieren von MQIPT verwenden Sie den Befehl **`mqiptAdmin`**. Weitere Informationen zur Verwaltung von MQIPT mit dem Befehl **`mqiptAdmin`** finden Sie im Abschnitt MQIPT über die Befehlszeile verwalten.

## HTTP -Unterstützung in MQIPT

MQIPT unterstützt HTTP-Tunnelung. MQIPT kann so konfiguriert werden, dass die weitergeleiteten Datenpakete als HTTP-Anforderungen codiert werden.

IBM MQ-Kanäle akzeptieren keine HTTP-Anforderungen. Daher ist ein zweites MQIPT erforderlich, um die HTTP-Anforderungen zu empfangen und sie wieder in Protokollpakete von IBM MQ zu konvertieren. Der zweite MQIPT entfernt den HTTP-Header, um das eingehende Paket zurück in ein Standardprotokollpaket von IBM MQ zu konvertieren, bevor es an den Zielwarteschlangenmanager übergeben wird.

Wenn HTTP zwischen zwei Instanzen von MQIPT verwendet wird, ist die TCP/IP-Verbindung, über die die HTTP-Anforderungen und -Antworten fließen, persistent und wird für die Laufzeit des Nachrichtenkanals offen gehalten. MQIPT schließt die TCP/IP-Verbindung zwischen Anforderungs-/Antwortpaaren nicht.

Wenn zwei Instanzen von MQIPT über HTTP kommunizieren, kann es sein, dass eine HTTP-Anforderung für einen längeren Zeitraum ausstehend bleibt. Dies könnte beispielsweise in einem Requester-/Serverkanal passieren, wenn die Serverseite auf die Ankunft neuer Nachrichten in ihrer Übertragungswarteschlange wartet. Das IBM MQ-Kanalprotokoll stellt einenÜberwachungssignalmechanismus ("Heartbeat"-Mechanismus) bereit, der verlangt, dass das wartende Ende regelmäßig Heartbeatnachrichten an seinen Partner sendet. Der Standardkanal-Überwachungssignalzeitraum beträgt 5 Minuten. MQIPT verwendet dieses Überwachungssignal als HTTP-Antwort. Inaktivieren Sie dieses Kanalüberwachungssignal nicht und legen Sie keinen zu hohen Wert für das Signal fest, um Probleme mit Zeitlimits in einigen Firewalls zu vermeiden.

MQIPT akzeptiert HTTP-Datenverkehr im aufgeteilten Format, der von einem HTTP-Proxy oder -Server generiert wird.

Ein Beispiel für die Verwendung von HTTP in MQIPT finden Sie unter HTTP-Tunneling konfigurieren.

### HTTP-Proxys

Ein HTTP-Proxy kann zwischen den beiden Instanzen von MQIPT platziert werden. Der HTTP-Proxy muss die folgenden Voraussetzungen erfüllen:

- Der Proxy muss das HTTP 1.1-Protokoll unterstützen.
- Die von MQIPT festgelegten **Connection** -oder **Proxy-Connection** HTTP -Header müssen vom Proxy berücksichtigt werden. Dadurch können Verbindungen zwischen den beiden Instanzen von MQIPT für die Lebensdauer des Nachrichtenkanals offen gehalten werden.
- Eine Eins-zu-eins-Zuordnung persistenter Verbindungen muss über den Proxy hinweg verwaltet werden. Dadurch wird sichergestellt, dass TCP/IP-Verbindungen vom Proxy zum Ziel MQIPT nicht zum Übertragen von Daten für mehr als einen Nachrichtenkanal verwendet werden.

Sie können Eigenschaften festlegen, um zu konfigurieren, wie persistente Verbindungen auf einigen HTTP-Proxys verwaltet werden. So können Sie beispielsweise die maximale Anzahl von Anforderungen festlegen, die für eine persistente Verbindung hergestellt werden können. Die folgenden Eigenschaften sollten festgelegt werden:

- Persistente Verbindungen sollten aktiviert werden.
- Die Wiederverwendung von TCP/IP-Verbindungen vom Proxy zu MQIPT durch mehr als eine HTTP-Sitzung sollte inaktiviert werden, um eine Eins-zu-eins-Zuordnung persistenter Verbindungen über den Proxy zu erhalten.
- Das Zeitlimit für Proxy-Anforderungen sollte auf einen hohen Wert gesetzt werden. Zum Beispiel 12 Stunden.
- Die maximale Anzahl an Anforderungen, die für eine persistente Verbindung erstellt werden können, sollte auf einen hohen Wert gesetzt werden. Beispiel: 5000

MQIPT verwendet HTTP-POST-Anforderungen, um Daten zwischen den beiden Instanzen von MQIPT zu senden. Wenn die Konfiguration von MQIPT den Hostnamen des Proxys unter Verwendung der Eigenschaft **HTTPProxy** angibt, stellt MQIPT eine Verbindung zum Proxy her und fordert die HTTP-CONNECT-Methode auf, um anzufordern, dass der Proxy einen Tunnel zum Ziel MQIPT herstellt. Dadurch können HTTPS-Verbindungen den Proxy passieren, ohne dass die TLS-Sitzung im Proxy beendet wird.

Wenn eine Lastausgleichsfunktion zwischen den MQIPT-Instanzen platziert wird, muss sie so konfiguriert werden, dass sie den Wert des *MQIPTSessionId*-HTTP-Cookies verwendet, um sicherzustellen, dass alle Anforderungen für jede Sitzung an dasselbe Ziel weitergeleitet werden.

## HTTPS in MQIPT

HTTPS kann in einer HTTP -Verbindung verwendet werden, indem die Routeneigenschaften **HTTPS** und **SSLClient** in der MQIPT aktiviert werden, die die Clientverbindung ausgibt.

MQIPT muss Zugriff auf das vertrauenswürdige CA-Zertifikat haben, das zur Authentifizierung des Ziel-HTTP-Proxys/-Servers verwendet wird. Die Eigenschaft **SSLClientCAKeyring** kann zur Definition der Schlüsselringdatei verwendet werden, die das vertrauenswürdige CA-Zertifikat enthält.

Bei einer gängigen Konfiguration für HTTPS wird ein lokaler HTTP-Proxy für die Übertragung im Tunnelungsverfahren über eine Firewall und für die Verbindung zu einem fernen HTTP-Server (oder einem anderen Proxy) verwendet, der wiederum eine Verbindung zum fernen MQIPT herstellen wird. Dieser MQIPT auf der Serverseite der Verbindung benötigt keine bestimmte Konfiguration, da die Verbindungsanforderung wie jede normale HTTP-Verbindung behandelt wird.

MQIPTverwendet die Eigenschaften **HTTPProxy** und **HTTPServer** zur Unterscheidung der lokalen und fernen Proxys. Die MQIPT-Route, in der die Eigenschaft **HTTPProxy** festgelegt ist, gilt als der lokale HTTP-Proxy, und die MQIPT-Route, in der die Eigenschaft **HTTPServer** festgelegt ist, wird als ferner Server (oder Proxy) betrachtet.

HTTPS -Verbindungen werden normalerweise zur Listener-Port-Adresse 443 auf dem HTTP -Proxy/-Server hergestellt, aber die Eigenschaften **HTTPProxyPort** und **HTTPServerPort** können zum Überschreiben dieses Standardwerts verwendet werden.

## SOCKS-Unterstützung in MQIPT

Ein SOCKS-Proxy ist ein Netzservice, der als kontrollierter Exitpunkt durch eine Firewall verwendet wird. Eine SOCKS-fähige Anwendung, die innerhalb der Firewall ausgeführt wird, kann über den SOCKS-Proxy eine Verbindung zu einer fernen Anwendung herstellen.

MQIPT kann als SOCKS-Proxy fungieren, indem die Eigenschaft **SocksServer** aktiviert wird. Dadurch kann eine SOCKS-fähige IBM MQ-Anwendung über MQIPT eine Verbindung zu einem fernen IBM MQ-Warteschlangenmanager herstellen. Wenn Sie diese Funktion verwenden, werden die Zieladresse und die Zielportadresse während des SOCKS-Handshakeverfahrens abgerufen. Daher werden die Routeneigenschaften **Destination** und **DestinationPort** überschrieben. Es handelt sich hier um eine Schlüsselfunktion für die Unterstützung von IBM MQ-Clustering.

MQIPT kann auch als SOCKS-Client für eine lokale IBM MQ-Anwendung fungieren, die nicht SOCKS-fähig ist. Dies ist bei Verwendung einer Firewall hilfreich, die abgehende Verbindungen nur über einen SOCKS-Proxy ermöglicht. Jede MQIPT-Route kann so konfiguriert werden, dass sie mit einem anderen SOCKS-Proxy kommuniziert.

In SOCKS-Proxy konfigurieren finden Sie ein Beispiel für die Verwendung von SOCKS.

## Clustering in MQIPT

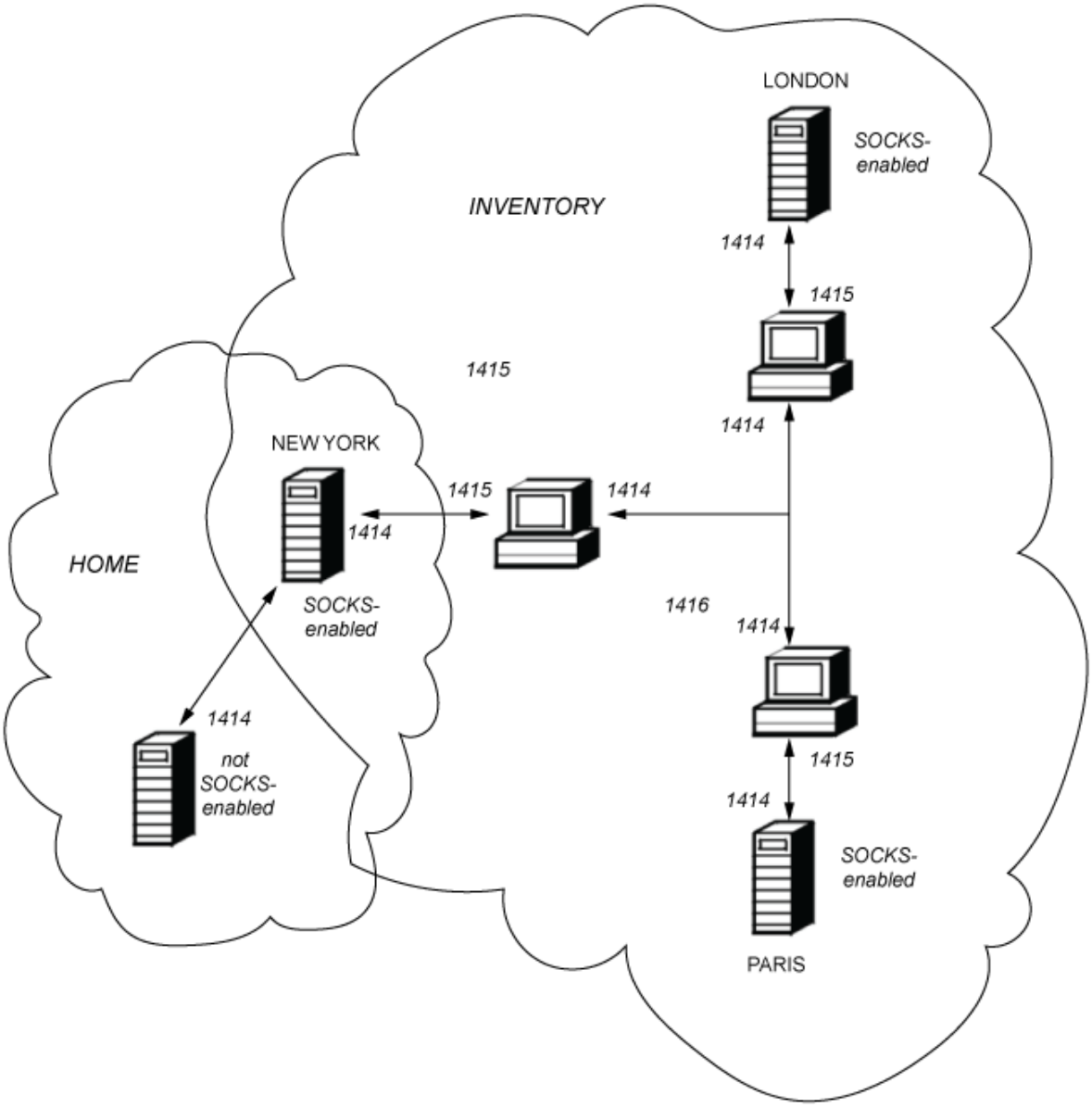
IBM MQ-Cluster können mit MQIPT verwendet werden, indem jeder Warteschlangenmanager in einem Cluster für die Verwendung im Internet für SOCKS aktiviert wird und indem MQIPT für Verwendung als SOCKS-Proxy aktiviert wird.

Im folgenden Diagramm befinden sich NEWYORK und CHICAGO in einem Cluster mit der Bezeichnung HOME und enthalten jeweils vollständige Repositorys. NEWYORK, LONDON und PARIS befinden sich in einem anderen Cluster mit der Bezeichnung INVENTORY. Beachten Sie, dass CHICAGO nicht für SOCKS aktiviert sein muss, da es sich in einem Cluster befindet, in dem MQIPT nicht benötigt wird.

Jeder Warteschlangenmanager im Cluster INVENTORY wird effektiv hinter MQIPT "versteckt". Da der Warteschlangenmanager SOCKS-fähig ist, wird die Anforderung beim Start eines Clustersenderkanals an das zugehörige Ziel gesendet, wobei MQIPT als SOCKS-Proxy verwendet wird. Normalerweise wird mit CONNAME in einem Clusterempfängerkanal der lokale Warteschlangenmanager ermittelt, aber bei der Verwendung mit MQIPT muss CONNAME das lokale MQIPT und die zugehörigen eingehenden Listener-Ports identifizieren. Im folgenden Diagramm haben alle eingehenden Listener-Ports die Adresse 1414 und alle ausgehenden Listener-Ports die Adresse 1415.

Ein SOCKS-fähiger Warteschlangenmanager kann auf zwei Arten ausgeführt werden. Der gesamte Computer, auf dem der Warteschlangenmanager ausgeführt wird, kann für SOCKS aktiviert werden. Es kann nur der Warteschlangenmanager für SOCKS aktiviert werden. Sie müssen mit einer dieser beiden Methoden den SOCKS-Client konfigurieren, damit dieser nur ferne Verbindungen mithilfe von MQIPT als SOCKS-Proxy herstellt und die Benutzerauthentifizierung inaktiviert ist. Es gibt eine Reihe von Produkten für die SOCKS-Unterstützung auf dem Markt. Sie müssen ein Produkt auswählen, dass das SOCKS V5-Protokoll unterstützt.

In Unterstützung für MQIPT-Clustering konfigurieren finden Sie ein Beispiel für die Konfiguration eines Clusternetzes.




## SSL/TLS-Unterstützung in MQIPT

Sichere Sockets können verwendet werden, um die Vertraulichkeit der Kommunikation, die Kommunikationsintegrität und die Authentifizierung zu gewährleisten.

**Datenschutz bei der Kommunikation**
Die Verbindung kann als nicht öffentliche Verbindung hergestellt werden. Die zwischen dem Client und dem Server auszutauschenden Daten können verschlüsselt werden, und nur der Sender und Empfänger können die Daten erfassen. Somit können private Informationen, wie z. B. Kreditkartennummern, sicher übertragen werden.

**Übertragungsintegrität**
Die Verbindung ist zuverlässig. Der Nachrichtentransport umfasst eine Nachrichtenintegritätsprüfung, die auf einer sicheren Hashfunktion basiert.

**Authentifizierung**
Der Client kann den Server authentifizieren, und ein authentifizierter Server kann den Client authentifizieren. Dies bedeutet, dass die Informationen garantiert nur zwischen den vorgesehenen Parteien

ausgetauscht werden können. Der Authentifizierungsmechanismus basiert auf dem Austausch von digitalen Zertifikaten (X.509v3-Zertifikate).

## Secure Sockets-Protokolle

In MQIPT werden sichere Sockets bereitgestellt, indem das Transport Layer Security-Protokoll (TLS) und das Secure Sockets Layer-Protokoll (SSL) verwendet werden. Die beiden sicheren Sockets-Protokolle sind sich ähnlich, können jedoch nicht miteinander zusammenarbeiten. In dieser Dokumentation sind die Begriffe SSL und TLS austauschbar, es sei denn, ein bestimmter Unterschied muss beachtet werden.

MQIPT unterstützt SSL 3.0, TLS 1.0, TLS 1.1 und TLS 1.2, die von der mitgelieferten Java runtime environment (JRE) bereitgestellt werden. V 9.3.0 Ab IBM MQ 9.3.0 unterstützt MQIPT auch TLS 1.3. Die IBM MQ-CipherSpec des fernen Kanals bestimmt, welches Protokoll MQIPT verwendet.

SSL 3.0, TLS 1.0und TLS 1.1 sind unsicher und in MQIPTstandardmäßig inaktiviert. Wenn Sie eines dieser inaktivierten Protokolle verwenden müssen, können sie mit der folgenden Prozedur in „Veraltete Protokolle und Cipher Suites in MQIPT aktivieren” auf Seite 1143 erneut aktiviert werden.

Die beiden Protokolle (SSL/TLS) können zur Authentifizierung der miteinander kommunizierenden Parteien unterschiedliche Algorithmen für digitale Signaturen verwenden. Die in SSL/TLS eingesetzten Verschlüsselungsoperationen, die Verschlüsselung für den Datenschutz und das sichere Hashverfahren für die Nachrichtenintegrität beruhen auf der Nutzung eines gemeinsamen Geheimschlüssels zwischen Client und Server. SSL/TLS stellt verschiedene Schlüsselaustauschmechanismen zur Verfügung, die die gemeinsame Nutzung von geheimen Schlüsseln ermöglichen. SSL/TLS kann für Verschlüsselung und Hashing verschiedene Algorithmen verwenden.

## FIPS-Modus in MQIPT aktivieren

Die SSL/TLS-Verschlüsselungskomponente der JRE enthält den `IBMJCEPlusFIPS` -Sicherheitsprovider, der gemäß dem FIPS 140-2-Standard zertifiziert ist. Wenn Sie nur FIPS-zertifizierte Verschlüsselung in MQIPTverwenden wollen, aktivieren Sie den FIPS-Modus im `IBMJSSE2` -Provider, indem Sie die folgenden Java -Systemeigenschaften festlegen, wenn MQIPT gestartet wird:

- `com.ibm.jsse2.usefipsprovider=true`
- V 9.3.0 `com.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS`

Sie können Java -Systemeigenschaften festlegen, wenn MQIPT mit der Umgebungsvariablen **`MQIPT_JVM_OPTIONS`** gestartet wird. Setzen Sie beispielsweise unter Linuxden folgenden Befehl ab, um die Umgebungsvariable festzulegen, bevor Sie den Befehl zum Starten von MQIPTabsetzen:

```
export MQIPT_JVM_OPTIONS="-Dcom.ibm.jsse2.usefipsprovider=true -Dcom.ibm.jsse2.usefipsProvider
Name=IBMJCEPlusFIPS"
```

Weitere Informationen zum Aktivieren des FIPS-Modus finden Sie unter FIPS-Modus im IBMJSSE2 -Provider aktivieren.

## SSL/TLS-Bridging-Modus

Wenn für eine Route sowohl SSLServer als auch SSLClient festgelegt sind, akzeptiert der MQIPT eine eingehende gesicherte SSL/TLS-Verbindung und stellt eine zweite gesicherte SSL/TLS-Verbindung zu einem anderen MQIPT oder zu einem Zielwarteschlangenmanager her. Die IBM MQ -Kanalinformationen werden zwischen diesen beiden SSL/TLS-Verbindungen entschlüsselt und erneut verschlüsselt. SSL/TLS-Bridging wird auch als *SSL/TLS-Terminierungsproxy*bezeichnet.

IBM MQ unterstützt SSL/TLS-Bridging über die MQIPT. Es wurde beobachtet, dass andere SSL/TLS-Beendigungsproxys mit IBM MQ unterbrochene Verbindungen verursachen, wenn der Proxy SSL/TLS-Datensätze mit anderen Größen als die von IBM MQgesendeten kombiniert oder rekonstruiert. Dies ist auf eine Interaktion zwischen der Art und Weise, wie Warteschlangenmanager Speicher für eingehende IBM MQ -Netzdaten zuordnen und verwalten, und der Art und Weise zurückzuführen, wie IBM MQ -Netzdaten in SSL/TLS-Datensätze gepackt werden.

MQIPT behält die Paketierung von IBM MQ -Netzdaten in SSL/TLS-Datensätzen bei, ohne sie aufzuteilen oder zu kombinieren. Wenn andere SSL/TLS-Brücken die SSL/TLS-Datensätze nicht exakt beibehalten, können sie dazu führen, dass IBM MQ -Kanäle mit Fehlernachrichten fehlschlagen:

```
AMQ9638: SSL communications error for channel
AMQ9208: Error on receive from host
```

## SSL/TLS-Proxy-Modus

Eine MQIPT -Route kann im SSL/TLS-Proxy-Modus als Alternative zur SSL/TLS-Überbrückung konfiguriert werden. In diesem Modus leitet die Route SSL/TLS-Daten nur zwischen zwei IBM MQ -Endpunkten; sie nimmt nicht am SSL-/TLS-Handshake teil und erfordert keine digitalen Zertifikate.

Sie können den SSL/TLS-Proxy-Modus in Fällen verwenden, in denen die IBM MQ -Kanäle, die über MQIPT kommunizieren, bereits für die SSL/TLS-Kommunikation konfiguriert sind und Sie MQIPT für einen anderen Zweck verwenden möchten, wie z. B. das Routing von Verbindungen über Firewalls oder das Beschränken der Gruppe zulässiger Verbindungen über einen Sicherheitsexit. Bei der Ausführung im SSL/TLS-Proxy-Modus überprüft MQIPT, ob die ersten SSL/TLS-Pakete, die von einer neuen Verbindung empfangen werden, gültig sind, bevor sie an an das Ziel übergeben werden.

IBM MQ unterstützt den SSL/TLS-Proxy-Modus mit dem MQIPT oder einem anderen SSL/TLS-Proxy

## IBM MQ-Unterstützung für mehrere Zertifikate mit MQIPT

IBM MQ 8.0 und höher unterstützen die Verwendung mehrerer Zertifikate auf demselben Warteschlangenmanager unter Verwendung einer kanalspezifischen Zertifikatsbezeichnung, die mit dem Attribut **CERTLABL** in der Kanaldefinition angegeben wird. Eingehende Kanäle zum Warteschlangenmanager (z. B. Serververbindung oder Empfänger) basieren auf der Erkennung des Kanalnamens unter Verwendung von TLS Server Name Indication (SNI), um das richtige Zertifikat vom WS-Manager zu präsentieren. Weitere Informationen zur Verwendung mehrerer Zertifikate auf einem Warteschlangenmanager finden Sie unter Funktionalität von IBM MQ für mehrere Zertifikate.

Wenn ein Kanal über MQIPTeine Verbindung zum Zielwarteschlangenmanager herstellt und für die MQIPT -Route sowohl **SSLServer** als auch **SSLClient** festgelegt ist, gibt es zwei separate TLS-Sitzungen zwischen den Endpunkten. In früheren Versionen als IBM MQ 9.3.0 werden die SNI-Daten nicht über den Sitzungsbruch verteilt. Dadurch wird verhindert, dass ein Kanalzertifikat auf dem Zielwarteschlangenmanager für die TLS-Verbindung zwischen MQIPT und dem Warteschlangenmanager verwendet wird. Zur Verwendung eines pro-Kanal-Zertifikats auf dem Zielwarteschlangenmanager für eine TLS-Verbindung, die über MQIPT in einer früheren Version als IBM MQ 9.3.0 führt, muss die MQIPT-Route den SSL/TLS-Proxy-Modus verwenden, der alle intakten TLS-Steuerungsflüsse einschließlich der SNI-Namen weiterleitet.

V 9.3.0 Ab IBM MQ 9.3.0 kann MQIPT mit der Routeneigenschaft SSLClientOutboundSNI konfiguriert werden, um entweder die SNI für TLS-Verbindungen auf einen bestimmten Wert zu setzen oder die SNI, die an der eingehenden Verbindung zur Route empfangen wurde, zu durchlaufen. Damit pro-Kanal-Zertifikate auf einem Zielwarteschlangenmanager verwendet werden können, muss die Route entweder so konfiguriert werden, dass die SNI auf den IBM MQ-Kanalnamen gesetzt oder die SNI, die an der eingehenden Verbindung empfangen wurde, an die Route übergeben wird. Wenn MQIPT für den Durchlauf durch die SNI konfiguriert ist, muss der Warteschlangenmanager oder Client, der eine Verbindung zu MQIPT herstellt, die SNI auf den Kanalnamen setzen.

Die Zertifikate, die für TLS-Verbindungen verwendet werden, die von MQIPT beendet oder initiiert werden, können für jede Route einzeln konfiguriert werden, z. B. mithilfe der Routeneigenschaften **SSLServerSiteLabel** oder **SSLClientSiteLabel**.

## Von MQIPT unterstützte CipherSuites

In der nachfolgenden Tabelle ist dargestellt, welche CipherSuites von MQIPT unterstützt werden und welche standardmäßig aktiviert sind.

Standardmäßig ist nur eine Untergruppe von CipherSuites aktiviert. CipherSuites, die auf einigen Algorithmen basieren, die als unsicher betrachtet werden, werden von der JRE inaktiviert. Wenn Sie sich der potenziellen Gefahren bewusst sind, aber immer noch eine dieser CipherSuites verwenden müssen, können Sie die Unterstützung für eine inaktivierte CipherSuite hinzufügen, indem Sie die Prozedur in „Veraltete Protokolle und Cipher Suites in MQIPT aktivieren” auf Seite 1143 befolgen.

*Tabelle 70. CipherSuites, die Sie mit MQIPT verwenden können*

| **CipherSuite** | **Standardmäßig aktiviert** |
|---|---|
| **CipherSuites für TLS 1.3** | |
| V 9.3.0 V 9.3.0 TLS_AES_128_GCM_SHA256 | Ja |
| V 9.3.0 V 9.3.0 TLS_AES_256_GCM_SHA384 | Ja |
| V 9.3.0 V 9.3.0 TLS_CHACHA20_POLY1305_SHA256 | Ja |
| **CipherSuites für SSL 3.0, TLS 1.0, TLS 1.1 und TLS 1.2** | |
| SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA | |
| SSL_DH_anon_EXPORT_WITH_RC4_40_MD5 | |
| SSL_DH_anon_WITH_3DES_EDE_CBC_SHA | |
| SSL_DH_anon_WITH_AES_128_CBC_SHA | |
| SSL_DH_anon_WITH_AES_128_CBC_SHA256 | |
| SSL_DH_anon_WITH_AES_128_GCM_SHA256 | |
| SSL_DH_anon_WITH_AES_256_CBC_SHA | |
| SSL_DH_anon_WITH_AES_256_CBC_SHA256 | |
| SSL_DH_anon_WITH_AES_256_GCM_SHA384 | |
| SSL_DH_anon_WITH_DES_CBC_SHA | |
| SSL_DH_anon_WITH_RC4_128_MD5 | |
| SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA | |
| SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA | |
| SSL_DHE_DSS_WITH_AES_128_CBC_SHA | Ja |
| SSL_DHE_DSS_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_DHE_DSS_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_DHE_DSS_WITH_AES_256_CBC_SHA | Ja |
| SSL_DHE_DSS_WITH_AES_256_CBC_SHA256 | Ja |
| SSL_DHE_DSS_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_DHE_DSS_WITH_DES_CBC_SHA | |
| SSL_DHE_DSS_WITH_RC4_128_SHA | |
| SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA | |
| SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA | |
| SSL_DHE_RSA_WITH_AES_128_CBC_SHA | Ja |

| *Tabelle 70. CipherSuites, die Sie mit MQIPT verwenden können (Forts.)* | |
|---|---|
| **CipherSuite** | **Standardmäßig aktiviert** |
| SSL_DHE_RSA_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_DHE_RSA_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_DHE_RSA_WITH_AES_256_CBC_SHA | Ja |
| SSL_DHE_RSA_WITH_AES_256_CBC_SHA256 | Ja |
| SSL_DHE_RSA_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_DHE_RSA_WITH_DES_CBC_SHA | |
| SSL_ECDH_anon_WITH_3DES_EDE_CBC_SHA | |
| SSL_ECDH_anon_WITH_AES_128_CBC_SHA | |
| SSL_ECDH_anon_WITH_AES_256_CBC_SHA | |
| SSL_ECDH_anon_WITH_NULL_SHA | |
| SSL_ECDH_anon_WITH_RC4_128_SHA | |
| SSL_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA | |
| SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA | Ja |
| SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA | Ja |
| SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 | Ja |
| SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_ECDH_ECDSA_WITH_NULL_SHA | |
| SSL_ECDH_ECDSA_WITH_RC4_128_SHA | |
| SSL_ECDH_RSA_WITH_3DES_EDE_CBC_SHA | |
| SSL_ECDH_RSA_WITH_AES_128_CBC_SHA | Ja |
| SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_ECDH_RSA_WITH_AES_256_CBC_SHA | Ja |
| SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384 | Ja |
| SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_ECDH_RSA_WITH_NULL_SHA | |
| SSL_ECDH_RSA_WITH_RC4_128_SHA | |
| SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA | |
| SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA | Ja |
| SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA | Ja |

| *Tabelle 70. CipherSuites, die Sie mit MQIPT verwenden können (Forts.)* | |
|---|---|
| **CipherSuite** | **Standardmäßig aktiviert** |
| SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 | Ja |
| SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_ECDHE_ECDSA_WITH_NULL_SHA | |
| SSL_ECDHE_ECDSA_WITH_RC4_128_SHA | |
| SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA | |
| SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA | Ja |
| SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA | Ja |
| SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384 | Ja |
| SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_ECDHE_RSA_WITH_NULL_SHA | |
| SSL_ECDHE_RSA_WITH_RC4_128_SHA | |
| SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5 | |
| SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA | |
| SSL_KRB5_EXPORT_WITH_RC4_40_MD5 | |
| SSL_KRB5_EXPORT_WITH_RC4_40_SHA | |
| SSL_KRB5_WITH_3DES_EDE_CBC_MD5 | |
| SSL_KRB5_WITH_3DES_EDE_CBC_SHA | |
| SSL_KRB5_WITH_DES_CBC_MD5 | |
| SSL_KRB5_WITH_DES_CBC_SHA | |
| SSL_KRB5_WITH_RC4_128_MD5 | |
| SSL_KRB5_WITH_RC4_128_SHA | |
| SSL_RSA_EXPORT_WITH_DES40_CBC_SHA | |
| SSL_RSA_EXPORT_WITH_RC4_40_MD5 | |
| SSL_RSA_WITH_3DES_EDE_CBC_SHA | |
| SSL_RSA_WITH_AES_128_CBC_SHA | Ja |
| SSL_RSA_WITH_AES_128_CBC_SHA256 | Ja |
| SSL_RSA_WITH_AES_128_GCM_SHA256 | Ja |
| SSL_RSA_WITH_AES_256_CBC_SHA | Ja |
| SSL_RSA_WITH_AES_256_CBC_SHA256 | Ja |
| SSL_RSA_WITH_AES_256_GCM_SHA384 | Ja |
| SSL_RSA_WITH_DES_CBC_SHA | |
| SSL_RSA_WITH_NULL_MD5 | |

| *Tabelle 70. CipherSuites, die Sie mit MQIPT verwenden können (Forts.)* | |
|---|---|
| **CipherSuite** | **Standardmäßig aktiviert** |
| SSL_RSA_WITH_NULL_SHA | |
| SSL_RSA_WITH_NULL_SHA256 | |
| SSL_RSA_WITH_RC4_128_MD5 | Ja |
| SSL_RSA_WITH_RC4_128_SHA | |
| V 9.3.0 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 | |
| V 9.3.0 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 | |
| V 9.3.0 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 | |

## CipherSpecs und MQIPT-CipherSuites

Die folgende Tabelle zeigt die Beziehung zwischen den CipherSpecs, die von IBM MQ unterstützt werden, und den von MQIPT unterstützten CipherSuites.

Die Tabelle zeigt auch die Protokollversion, deren Verwendung IBM MQ bei den einzelnen CipherSpecs erwartet.

Eine IBM MQ-CipherSpec bestimmt eindeutig sowohl den Verschlüsselungsalgorithmus als auch die sichere Socket-Protokollversion, die verwendet werden soll. Einige IBM MQ-CipherSpecs unterscheiden sich nur in der Protokollversion, so dass es nicht ausreicht, die CipherSuite allein zu konfigurieren. Beim SSL/TLS-Handshake wird die höchste sichere Socketprotokollversion vereinbart, die von beiden Seiten unterstützt wird, und wählt dann eine CipherSuite aus der Gruppe der beiderseits aktivierten Chiffrierwerte aus.

Eine SSLClient-Route mit `SSLClientCipherSuites=SSL_RSA_WITH_3DES_EDE_CBC_SHA` könnte beispielsweise entweder TLS_RSA_WITH_3DES_EDE_CBC_SHA (TLS 1.0) oder TRIPLE_DES_SHA_US (SSL 3.0) mit dem fernen Warteschlangenmanager vereinbaren. In der Tat ist es möglich, diese CipherSuite über TLS 1.2 auszuhandeln, aber IBM MQ unterstützt diese CipherSuite nicht über TLS 1.2. Aus diesem Grund ist es besonders wahrscheinlich, dass SSLClient-Routen AMQ9616- oder AMQ9631-Fehler beim Warteschlangenmanager verursachen.

Um solche Fehler auf SSLClient-Routen zu vermeiden, setzen Sie die Routeneigenschaft **`SSLClient-Protocols`** auf den entsprechenden Wert für die vorgesehene CipherSpec. In einigen Fällen kann es auch erforderlich werden, die serverseitige Protokollgruppe mithilfe der Routeneigenschaft **`SSLServer-Protocols`** einzuschränken. Ermitteln Sie anhand der in der Tabelle angezeigten Protokollversion die korrekte Einstellung für diese Routeneigenschaften.

Dieses Problem betrifft insbesondere die folgenden CipherSuites und CipherSpecs für SSLClient-Routen:

- SSL_RSA_WITH_3DES_EDE_CBC_SHA, was folgenden Werten entspricht:
  - SSL 3.0: MQ CipherSpec TRIPLE_DES_SHA_US
  - TLS 1.0: MQ CipherSpec TLS_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA, was folgenden Werten entspricht:
  - SSL 3.0: MQ CipherSpec DES_SHA_EXPORT
  - TLS 1.0: MQ CipherSpec TLS_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA, was folgenden Werten entspricht:
  - SSL 3.0: MQ CipherSpec RC4_SHA_US
  - TLS 1.2: MQ CipherSpec TLS_RSA_WITH_RC4_128_SHA256

Wenn Sie eine einzelne MQIPT SSLClient-Route verwenden möchten, um mehrere IBM MQ-Kanäle zu tunneln, die verschiedene CipherSpecs verwenden, stellen Sie sicher, dass alle Kanäle über CipherSpecs verfügen, die dieselbe Secure Sockets-Protokollversion verwenden, und dass Sie **SSLClientProtocols** für die Verwendung dieser einzigen Protokollversion festlegen.

Weitere Informationen zu IBM MQ-CipherSpecs finden Sie im Abschnitt CipherSpecs aktivieren.

| IBM MQ CipherSpec | MQIPT CipherSuite | Protokollversion |
|---|---|---|
| DES_SHA_EXPORT | SSL_RSA_WITH_DES_CBC_SHA | SSLv3 |
| DES_SHA_EXPORT1024 | nicht zutreffend | nicht zutreffend |
| ECDHE_ECDSA_3DES_EDE_CBC_SHA256 | SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA | TLSv1.2 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 | SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 | TLSv1.2 |
| ECDHE_ECDSA_AES_128_GCM_SHA256 | SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 | TLSv1.2 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 | SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 | TLSv1.2 |
| ECDHE_ECDSA_AES_256_GCM_SHA384 | SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 | TLSv1.2 |
| ECDHE_ECDSA_NULL_SHA256 | SSL_ECDHE_ECDSA_WITH_NULL_SHA | TLSv1.2 |
| ECDHE_ECDSA_RC4_128_SHA256 | SSL_ECDHE_ECDSA_WITH_RC4_128_SHA | TLSv1.2 |
| ECDHE_RSA_3DES_EDE_CBC_SHA256 | SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA | TLSv1.2 |
| ECDHE_RSA_AES_128_CBC_SHA256 | SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256 | TLSv1.2 |
| ECDHE_RSA_AES_128_GCM_SHA256 | SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | TLSv1.2 |
| ECDHE_RSA_AES_256_CBC_SHA384 | SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384 | TLSv1.2 |
| ECDHE_RSA_AES_256_GCM_SHA384 | SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384 | TLSv1.2 |
| ECDHE_RSA_NULL_SHA256 | SSL_ECDHE_RSA_WITH_NULL_SHA | TLSv1.2 |
| ECDHE_RSA_RC4_128_SHA256 | SSL_ECDHE_RSA_WITH_RC4_128_SHA | TLSv1.2 |
| NULL_MD5 | SSL_RSA_WITH_NULL_MD5 | SSLv3 |
| NULL_SHA | SSL_RSA_WITH_NULL_SHA | SSLv3 |
| RC2_MD5_EXPORT | nicht zutreffend | nicht zutreffend |
| RC4_56_SHA_EXPORT1024 | nicht zutreffend | nicht zutreffend |
| RC4_MD5_EXPORT | SSL_RSA_EXPORT_WITH_RC4_40_MD5 | SSLv3 |
| RC4_MD5_US | SSL_RSA_WITH_RC4_128_MD5 | SSLv3 |
| RC4_SHA_US | SSL_RSA_WITH_RC4_128_SHA | SSLv3 |
| V 9.3.0 V 9.3.0 TLS_AES_128_GCM_SHA256 | TLS_AES_128_GCM_SHA256 | TLSv1.3 |

| IBM MQ CipherSpec | MQIPT CipherSuite | Protokollversion |
|---|---|---|
| V 9.3.0 V 9.3.0 TLS_AES_256_GCM_SHA384 | TLS_AES_256_GCM_SHA384 | TLSv1.3 |
| V 9.3.0 V 9.3.0 TLS_CHACHA20_POLY1305_SHA256 | TLS_CHACHA20_POLY1305_SHA256 | TLSv1.3 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA | SSL_RSA_WITH_3DES_EDE_CBC_SHA | TLSv1 |
| TLS_RSA_WITH_AES_128_CBC_SHA | SSL_RSA_WITH_AES_128_CBC_SHA | TLSv1 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 | SSL_RSA_WITH_AES_128_CBC_SHA256 | TLSv1.2 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 | SSL_RSA_WITH_AES_128_GCM_SHA256 | TLSv1.2 |
| TLS_RSA_WITH_AES_256_CBC_SHA | SSL_RSA_WITH_AES_256_CBC_SHA | TLSv1 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 | SSL_RSA_WITH_AES_256_CBC_SHA256 | TLSv1.2 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 | SSL_RSA_WITH_AES_256_GCM_SHA384 | TLSv1.2 |
| TLS_RSA_WITH_DES_CBC_SHA | SSL_RSA_WITH_DES_CBC_SHA | TLSv1 |
| TLS_RSA_WITH_NULL_NULL | nicht zutreffend | nicht zutreffend |
| TLS_RSA_WITH_NULL_SHA256 | SSL_RSA_WITH_NULL_SHA256 | TLSv1.2 |
| TLS_RSA_WITH_RC4_128_SHA256 | SSL_RSA_WITH_RC4_128_SHA | TLSv1.2 |
| TRIPLE_DES_SHA_US | SSL_RSA_WITH_3DES_EDE_CBC_SHA | SSLv3 |

## SSL/TLS-Handshake in MQIPT

Das SSL/TLS-Handshakeverfahren erfolgt bei der einleitenden Verbindungsanforderung zwischen dem SSL/TLS-Client und -Server, und zwar während der Authentifizierung und der Festlegung der CipherSuites.

Alle unter „SSL/TLS-Unterstützung in MQIPT" auf Seite 1117 aufgeführten SSL/TLS-CipherSuites (mit Ausnahme der anonymen CipherSuites) erfordern die Serverauthentifizierung und gestatten die Clientauthentifizierung, d. h., der Server kann so konfiguriert werden, dass eine Clientauthentifizierung angefordert wird. Sie sollten keine anonymen CipherSuites verwenden, da mit diesen die Identität des fernen Peers nicht garantiert werden kann. Durch eine Man-in-the-Middle-Attacke können anonyme SSL/TLS-Verbindungen ohne Ihr Wissen abgefangen werden. Verwenden Sie anonyme CipherSuites nur in vertrauenswürdigen internen Netzen und nur, wenn Sie bereit sind, das Risiko des Datenabfangs zu akzeptieren.

Die Peerauthentifizierung für die Kommunikation in SSL/TLS basiert auf einer Verschlüsselung mit öffentlichen Schlüsseln und digitalen X.509v3-Zertifikaten. Für eine Site, die im SSL/TLS-Protokoll authentifiziert werden soll, ist ein privater Schlüssel, ein digitales Zertifikat (das den zugehörigen privaten Schlüssel mit den Informationen zur Identität der Site enthält) und die Gültigkeitsdauer des Zertifikats erforderlich. Die Zertifikate werden von einer Zertifizierungsstelle signiert und anschließend als Unterzeichnerzertifikate bezeichnet. Ein Zertifikat gefolgt von einem oder mehreren Unterzeichnerzertifikaten bildet eine Zertifikatskette. Eine Zertifikatskette wird dadurch gekennzeichnet, dass die Signatur jedes Zertifikats in der Kette ab dem ersten Zertifikat (Sitezertifikat) mithilfe des öffentlichen Schlüssels im nächsten Unterzeichnerzertifikat geprüft werden kann.

Wenn eine sichere Verbindung aufgebaut wird, für die eine Serverauthentifizierung erforderlich ist, sendet der Server eine Zertifikatskette an den Client, um die Identität zu prüfen. Der SSL/TLS-Client führt den Verbindungsaufbau mit den Server nur dann fort, wenn der Server authentifiziert werden kann, beispielsweise durch die Prüfung der Signatur des Sitezertifikats für den Server. Zur Überprüfung dieser Signatur muss der SSL/TLS-Client der Serversite selbst oder zumindest den Unterzeichnern in der vom Server be-

reitgestellten Zertifikatskette vertrauen. Die Zertifikate der vertrauenswürdigen Sites und Unterzeichner müssen auf der Clientseite beibehalten werde, damit diese Überprüfung ausgeführt werden kann.

Der SSL/TLS-Client untersucht die Zertifikatskette des Servers und beginnt dabei beim Sitezertifikat. Der Client geht in den folgenden Fällen davon aus, dass die Signatur des Sitezertifikats gültig ist:

- Das Sitezertifikat befindet sich im Repository der vertrauenswürdigen Site- oder Unterzeichnerzertifikate
- Ein Unterzeichnerzertifikat in der Kette kann auf Basis der zugehörigen Repositorys der vertrauenswürdigen Unterzeichnerzertifikate geprüft werden

Im letzteren Falls überprüft der SSL/TLS-Client, ob die Zertifikatskette vom vertrauenswürdigen Unterzeichnerzertifikat bis zum Sitezertifikat des Servers tatsächlich korrekt signiert ist. Jedes Zertifikat, das an diesem Prozess beteiligt ist, wird auch auf die Richtigkeit des Formats und des Gültigkeitsdatums geprüft. Wenn eine dieser Prüfungen fehlschlägt, wird die Verbindung zum Server abgelehnt. Nach der Überprüfung des Serverzertifikats verwendet der Client den öffentlichen Schlüssel, der in dieses Zertifikat integriert ist, in den nächsten Schritten für das SSL/TLS-Protokoll. Die SSL/TLS-Verbindung kann nur aufgebaut werden, wenn der Server tatsächlich über den zugehörigen privaten Schlüssel verfügt.

Bei der Clientauthentifizierung wird genauso vorgegangen: Wenn für einen SSL/TLS-Server eine Clientauthentifizierung erforderlich ist, sendet der Client eine Zertifikatskette an den Server, um seine Identität zu beweisen. Der Server überprüft die Kette auf Basis des Repositorys der vertrauenswürdigen Site- und Unterzeichnerzertifikate. Nach der Überprüfung des Clientzertifikats verwendet der Client den öffentlichen Schlüssel, der in dieses Zertifikat integriert ist, in den nächsten Schritten für das SSL/TLS-Protokoll. Die SSL/TLS-Verbindung kann nur aufgebaut werden, wenn der Client tatsächlich über den zugehörigen privaten Schlüssel verfügt.

Die aktuellen Versionen der TLS-Protokolle stellen eine Kommunikation mit einem hohen Sicherheitsniveau bereit (SSL- und ältere TLS-Protokolle werden als unsicher betrachtet). Allerdings wird das Protokoll auf Basis der Informationen ausgeführt, die von der Anwendung bereitgestellt werden. Nur wenn diese Informationsbasis ebenfalls auf sichere Weise verwaltet wird, kann eine sichere Kommunikation als Gesamtziel erreicht werden. Wenn beispielsweise Ihr Repository aus vertrauenswürdigen Site- und Unterzeichnerzertifikaten beeinträchtigt ist, wird möglicherweise eine sichere Verbindung zu einem nicht sicheren Kommunikationspartner hergestellt.

## MQIPT-Implementierung von SSL/TLS

SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2 und TLS 1.3 werden mit PKCS #12 -Tokens (PKCS = Public Key Cryptography Standards) implementiert, die in Schlüsselringdateien (mit den Dateitypen `.p12` oder `.pfx`) mit X509.V3 -Zertifikate. MQIPT kann auch Schlüsselspeicher für Verschlüsselungshardware verwenden, die den PKCS#11 Cryptographic Token Interface-Standard unterstützen. MQIPT verwendet das JSSE-Paket (IBM Java Secure Socket Extension).

MQIPT kann als SSL/TLS-Client oder SSL/TLS-Server fungieren, je nachdem, von welcher Seite die Verbindung eingeleitet wurde. Der Client startet eine Verbindung und der Server akzeptiert die Verbindungsanforderung. Eine MQIPT-Route kann als Client und als Server verwendet werden. In diesem Falls wird mit dem SSL/TLS-Proxy-Modus normalerweise eine bessere Leistung erreicht.

Wenn MQIPT für die Verwendung des SSL/TLS-Proxy-Modus konfiguriert ist, werden nur SSL/TLS-Daten zwischen den beiden Endpunkten übertragen; es findet keine Beteiligung am SSL/TLS-Handshake statt und es sind keine digitalen Zertifikate erforderlich.

In Versionen vor IBM MQ 9.3.0MQIPT übergibt keine TLS-SNI-Daten (SNI = Server Name Indication), die über eine eingehende TLS-Verbindung empfangen werden, an eine abgehende TLS-Verbindung. Dies bedeutet, dass Zertifikate pro Kanal, die über das Kanalattribut **CERTLABL** angegeben werden, für TLS-Verbindungen zwischen MQIPT und dem Zielwarteschlangenmanager nicht verwendet werden können. Zur Verwendung eines pro-Kanal-Zertifikats auf dem Zielwarteschlangenmanager für eine TLS-Verbindung, die über MQIPT in einer früheren Version als IBM MQ 9.3.0 führt, muss die MQIPT-Route den SSL/TLS-Proxy-Modus verwenden, der alle intakten TLS-Steuerungsflüsse einschließlich der SNI-Namen weiterleitet. V 9.3.0 Ab IBM MQ 9.3.0 kann MQIPT so konfiguriert werden, dass entweder die SNI für TLS-Verbindungen auf einen bestimmten Wert gesetzt wird oder dass die in der eingehenden Verbin-

dung empfangene SNI an die Route übergeben wird. Weitere Informationen zur Verwendung mehrerer Zertifikate auf einem Warteschlangenmanager mit MQIPT finden Sie unter „IBM MQ-Unterstützung für mehrere Zertifikate mit MQIPT” auf Seite 1119.

Jede MQIPT-Route kann unabhängig mit einer eigenen Gruppe von SSL/TLS-Eigenschaften konfiguriert werden. Weitere Informationen finden Sie im Abschnitt Routeneigenschaften für MQIPT.

## Schlüsselringkennwort in MQIPT verschlüsseln

Verschlüsseln Sie das Kennwort, das zum Öffnen einer Schlüsselringdatei oder für den Zugriff auf die von MQIPTverwendete Verschlüsselungshardware verwendet wird, mit dem Befehl **mqiptPW** . Das verschlüsselte Kennwort kann von einer der folgenden Eigenschaften verwendet werden: **SSLClientKeyRingPW**, **SSLClientCAKeyRingPW**, **SSLServerKeyRingPW**, **SSLServerCAKeyRingPW** und **SSLCommandPortKeyRingPW**. In diesem Abschnitt wird die ordnungsgemäße Vorgehensweise zum Speichern eines Schlüsselringkennworts beschrieben, das von MQIPT verwendet werden kann.

Die Stashdatei **mqiptkeyman** (iKeyman) wird von MQIPT nicht unterstützt. Anstelle einer Stashdatei müssen Sie den Befehl **mqiptPW** verwenden, um das verschlüsselte Kennwort zu speichern.

In Versionen vor IBM MQ 9.1.5werden Schlüsselringkennwörter für die Verwendung durch MQIPT in Dateien gespeichert, auf die durch eine der **SSL*KeyRingPW** -Eigenschaften verwiesen wird.

Verschlüsseln Sie ab IBM MQ 9.1.5 Schlüsselringkennwörter für die Verwendung durch MQIPT mit dem Befehl **mqiptPW** und geben Sie das verschlüsselte Kennwort als Wert für die **SSL*KeyRingPW**-Eigenschaften an. MQIPT ist in der Lage, zwischen verschlüsselten Kennwörtern und Dateinamen in Eigenschaftswerten zu unterscheiden, um Kompatibilität mit Konfigurationen, die vor IBM MQ 9.1.5 erstellt wurden, herzustellen.

Deprecated Die Methode zum Verschlüsseln von Keystore-Kennwörtern, die in MQIPT-Versionen vor IBM MQ 9.1.5 verfügbar sind, ist veraltet, kann aber dennoch verwendet werden. Um den Schutz von Schlüsselringkennwörtern zu verbessern, sollten alle Schlüsselringkennwörter, die zuvor verschlüsselt wurden, mit der neuesten Zugriffsschutzmethode erneut verschlüsselt werden.

Zum Verschlüsseln eines Schlüsselringkennworts für die Verwendung durch MQIPTführen Sie die Schritte in „Gespeicherte Kennwörter in MQIPT verschlüsseln” auf Seite 1156aus.

Sie müssen das Kennwort `mqiptSample` zum Öffnen einer der Beispielschlüsselringdateien im Unterverzeichnis `samples/ssl` des Installationsverzeichnisses von MQIPT verwenden.

## Zertifikate aus einer Schlüsselringdatei in MQIPT auswählen

Es können mehrere persönliche Zertifikate in der gleichen Schlüsselringdatei oder im gleichen Verschlüsselungshardware-Token gespeichert werden. Daher können die **SSLClientSite***-Eigenschaften auf der Clientseite verwendet werden, um das Zertifikat auszuwählen, das zur Authentifizierung an den Server gesendet werden soll, und die **SSLServerSite***-Eigenschaften können auf der Serverseite verwendet werden, um das Zertifikat auszuwählen, das zur Authentifizierung an den Client gesendet werden soll.

Mithilfe dieser Eigenschaften kann ein Zertifikat auf der Basis seines definierten Namens (Distinguished Name, DN) ausgewählt werden. Alternativ dazu kann die Zertifikatsbezeichnung verwendet werden, um ein Zertifikat unter Verwendung der Eigenschaften **SSLServerSiteLabel** und **SSLClientSiteLabel** auszuwählen.

Für die Auswahl des vom TLS-Befehlsport verwendeten Serverzertifikats geben Sie mit der Eigenschaft **SSLCommandPortSiteLabel** die Bezeichnung des Zertifikats an.

## Vertrauensstellungen in MQIPT

Eine Schlüsselringdatei enthält ein persönliches Zertifikat, welches das Unterzeichnerzertifikat oder eine Kette von Unterzeichnerzertifikaten enthält.

Es gibt zwei Typen von Schlüsselringen, die von MQIPT verwendet werden:

**Schlüsselringdatei der Zertifizierungsstelle (CA)**

Dieser Schlüsselring enthält vertrauenswürdige CA-Zertifikate, die zur Validierung von Zertifikaten verwendet werden, die zu einem fernen Peer gehören. Mithilfe dieser CA-Zertifikate lässt sich ermitteln, ob der ferne Peer vertrauenswürdig ist. Für das Speichern von CA-Zertifikaten unterstützt MQIPT Schlüsselringdateien im PKCS #12-Format und Schlüsselspeicher für die Verschlüsselungshardware, die die PKCS #11-Schnittstelle unterstützen. Die CA-Schlüsseldateien von MQIPT werden über die Routeneigenschaften **SSLClientCAKeyRing** und **SSLServerCAKeyRing** ermittelt. Die Verwendung von Verschlüsselungshardware für den Zugriff auf CA-Zertifikate wird durch Festlegen der Eigenschaften **SSLClientCAKeyRingUseCryptoHardware** und **SSLServerCAKeyRingUseCryptoHardware** aktiviert.

Die Schlüsselringdatei auf der SSL/TLS-Clientseite sollte eine Liste vertrauenswürdiger CA-Zertifikate enthalten, die zur Authentifizierung des vom Server gesendeten Zertifikats verwendet werden. Wenn ein SSL-Server für die Clientauthentifizierung konfiguriert ist, sollte der CA-Schlüsselring auf der SSL/TLS-Serverseite eine Liste vertrauenswürdiger CA-Zertifikate enthalten, die zur Authentifizierung des vom Client gesendeten Zertifikats verwendet werden.

**Schlüsselringdatei für persönliche Zertifikate**

Dieser Schlüsselring enthält persönliche Zertifikate, mit denen sich MQIPT selbst bei einem fernen Peer identifiziert. Wenn Sie ein selbst signiertes Zertifikat generieren oder ein CA-signiertes Zertifikat anfordern, sollten Sie hierfür den Schlüsselring für persönliche Zertifikate verwenden. Für das Speichern von persönlichen Zertifikaten unterstützt MQIPT Schlüsselringdateien im PKCS #12-Format und Schlüsselspeicher für die Verschlüsselungshardware, die die PKCS #11-Schnittstelle unterstützen. In MQIPT werden Schlüsselringdateien für persönliche Zertifikate über die Routeneigenschaften **SSLClientKeyRing** und **SSLServerKeyRing** identifiziert. Die Verwendung von Verschlüsselungshardware für den Zugriff auf persönliche Zertifikate wird durch Festlegen der Eigenschaften **SSLClientKeyRingUseCryptoHardware** und **SSLServerKeyRingUseCryptoHardware** aktiviert.

Der Schlüsselring auf der SSL/TLS-Serverseite sollte das persönliches Zertifikat des MQIPT-Servers enthalten. Wenn die Clientauthentifizierung in einer SSL-Client-Route erforderlich ist, sollte der Schlüsselring auf der SSL/TLS-Clientseite das persönliche Zertifikat des Clients enthalten.

Wenn Sie die Clientauthentifizierung benötigen, müssen Sie die Eigenschaft **SSLServerAskClientAuth** auf der Serverseite aktivieren. Der Schlüsselring auf der Clientseite sollte das persönliche Zertifikat des Client enthalten. Der MQIPT-Schlüsselring auf der Serverseite, der durch die Eigenschaft **SSLServerCAKeyRing** angegeben wird, solle eine Liste vertrauenswürdiger CA-Zertifikate enthalten, die zum Authentifizieren des Client verwendet werden.

Wenn Sie keinen CA-Schlüsselring für eine Route konfigurieren, werden CA-Zertifikate von MQIPT stattdessen im Schlüsselring für persönliche Zertifikate gesucht, falls einer konfiguriert ist. Wenn beispielsweise für **SSLServerCAKeyRing** kein Wert festgelegt ist, sucht MQIPT in dem Schlüsselring, der durch **SSLServerKeyRing** angegeben wird, nach CA-Zertifikaten.

Als Alternative zur Verwendung von Zertifikaten, die von einer vertrauenswürdigen Zertifizierungsstelle signiert wurden, können Sie selbst signierte Zertifikate verwenden. Sie finden ein Beispiel für ein selbst signiertes Zertifikat in der `sslSample.pfx`-Beispielschlüsselringdatei, die mit MQIPT im Unterverzeichnis `samples/ssl` bereitgestellt wird. Zum Öffnen der PKCS#12-Beispielschlüsselringdateien müssen Sie das Kennwort `mqiptSample` verwenden.

Selbst signierte Zertifikate können in Testszenarios hilfreich sein, in denen Sie SSL/TLS-Konnektivität sicherstellen müssen, ohne eine Zertifizierungsstelle für ein Zertifikat zu bezahlen. In Produktionsumgebungen sollten selbst signierte Zertifikate jedoch nicht verwendet werden. Informationen zum Erstellen eines CA-signierten Zertifikats finden Sie im Artikel Schlüsselringdatei erstellen.

Sie können das Dienstprogramm **mqiptkeyman**verwenden, das mit MQIPTbereitgestellt wird, um digitale Zertifikate und Keystores zu verwalten. Installationsanweisungen und weitere Informationen finden Sie im Abschnitt „mqiptKeyman und mqiptKeycmd in MQIPT” auf Seite 1132.

Sie müssen alle Schlüsselringdateien und Kennwortdateien mit den Sicherheitsfunktionen des Betriebssystems schützen, um einen unbefugten Zugriff auf diese Dateien zu verhindern.

## SSL/TLS in MQIPT testen

Beispiele für das Testen einer SSL/TLS-Verbindung.

In Erste Schritte mit IBM MQ Internet Pass-Thru finden Sie eine Beschreibung verschiedener Szenarios. Sehen Sie sich insbesondere die folgenden Tasks an:

- SSL-/TLS-Server authentifizieren
- SSL-/TLS-Client authentifizieren
- MQIPT im SSL/TLS-Proxy-Modus ausführen
- MQIPT im SSL/TLS-Proxy-Modus mit einem Sicherheitsmanager ausführen

Um zu testen, ob Ihre SSL/TLS-Konfiguration ordnungsgemäß funktioniert, können Sie selbst signierte Zertifikate verwenden. Selbst signierte Zertifikate sind in Testszenarios nützlich, so dass Sie die SSL/TLS-Konnektivität sicherstellen können, ohne eine Zertifizierungsstelle (Certificate Authority, CA) für ein Zertifikat bezahlen zu müssen. Weitere Informationen finden Sie im Artikel Testzertifikate erstellen.

Sie finden ein Beispiel für ein selbst signiertes Zertifikat in der `sslSample.pfx`-Beispielschlüsselringdatei, die mit MQIPT im Unterverzeichnis `samples/ssl` bereitgestellt wird. Zum Öffnen der PKCS#12-Beispielschlüsselringdateien müssen Sie das Kennwort `mqiptSample` verwenden. Das Beispielzertifikat wird Ihnen zur Arbeitserleichterung während des Tests bereitgestellt. Die privaten Schlüssel des Beispielzertifikats sind allerdings allen MQIPT-Benutzern bekannt. Dies bedeutet, dass es unsicher ist und nur in einer Testumgebung verwendet werden sollte.

In Produktionsumgebungen sollten Sie keine selbst signierten Zertifikate verwenden, unabhängig davon, ob es sich um Beispielzertifikate handelt oder nicht. Fordern Sie stattdessen ein CA-signiertes Zertifikat von einer anerkannten Zertifizierungsstelle an. Informationen zum Erstellen eines CA-signierten Zertifikats finden Sie im Artikel Schlüsselringdatei erstellen.

Wenn Sie ein Zertifikat erstellen oder anfordern, sollten Sie berücksichtigen, welcher Schlüsseltyp, welche Schlüsselgröße und welcher digitale Signaturalgorithmus für Ihre Sicherheitsanforderungen geeignet sind. Weitere Informationen finden Sie in „Hinweise zu digitalen Zertifikaten für MQIPT" auf Seite 1134 .

Zertifikate und Zertifikatsmanagement-Technologien sind von einer Reihe von Drittanbietern verfügbar.

## SSL/TLS-Fehlernachrichten in MQIPT

Handshake-Fehler werden im MQIPT-Verbindungsprotokoll in Form von JSSE-Ausnahmebedingungen protokolliert.

Weitere Informationen finden Sie unter „Verbindungsprotokolle in MQIPT" auf Seite 1159. In der folgenden Tabelle werden die verschiedenen Ausnahmebedingungen, die wahrscheinliche Ursache und die entsprechende Aktion zur Fehlerbehebung beschrieben.

Zertifikatsausnahmen beziehen sich in der Regel auf die Zertifikate am fernen Ende der Verbindung.

Wenn sich der Fehler auf das Zertifikat eines IBM MQ-Clients oder eines Warteschlangenmanagers bezieht, enthält der Begriff *Schlüsselringdatei* das IBM MQ-Schlüsselrepository des fernen Partners.

In MQIPT werden Zertifikate einer Zertifizierungsstelle (CA-Zertifikate) in der Schlüsselringdatei der Zertifizierungsstelle gespeichert, die durch die Routeneigenschaften **SSLClientCAKeyRing** und **SSLServerCAKeyRing** angegeben wird. Wenn die Routeneigenschaften eines CA-Schlüsselrings nicht festgelegt sind, wird stattdessen die zugehörige persönliche Schlüsselringdatei (auf die aus der Eigenschaft **SSLClientKeyRing** oder **SSLServerKeyRing** verwiesen wird) nach CA-Zertifikaten durchsucht.

| Ausnahmebedingung | Ursache | Aktion |
| --- | --- | --- |
| CertificateException | Das Zertifikat ist nicht vertrauenswürdig, da es von einer Zertifizierungsstelle signiert wurde, die sich nicht im Schlüsselring der Zertifizierungsstelle befindet. | Stellen Sie sicher, dass sich alle erforderlichen CA-Zertifikate in der Schlüsselringdatei der Zertifizierungsstelle befinden. Verwenden Sie das IBM Key Management-Tool, das mit MQIPT bereitgestellt wird, um fehlende CA-Zertifikate hinzuzufügen. Beachten Sie dabei, eine Kopie jedes CA-Zertifikats aus einer vertrauenswürdigen Quelle abzurufen. |
| CertificateExpiredException | 1. Das Zertifikat ist abgelaufen: Das Datum mit dem Wert **notAfter** wurde überschritten.<br>2. Die Systemuhr ist falsch eingestellt. | 1. Fordern Sie ein neues Zertifikat an und fügen Sie es in die Schlüsselringdatei ein. Wenn das Zertifikat zu einer Zertifizierungsstelle gehört, speichern Sie das neue Zertifikat in der Schlüsselringdatei der Zertifizierungsstelle.<br>2. Stellen Sie sicher, dass die korrekte koordinierte Weltzeit für die Systemuhr festgelegt ist. |
| CertificateNotYetValidException | 1. Das Zertifikat wird vorzeitig verwenden: Das Datum mit dem Wert **notBefore** wurde noch nicht erreicht.<br>2. Die Systemuhr ist falsch eingestellt. | 1. Stellen Sie sicher, dass das Zertifikat generiert und korrekt signiert wurde. Wenn Ihr Unternehmen eine eigene Zertifizierungsstelle betreibt, ist die Systemuhr mit der koordinierten Weltzeit für die Zertifizierungsstelle möglicherweise falsch.<br>2. Stellen Sie sicher, dass die korrekte koordinierte Weltzeit für die Systemuhr festgelegt ist. |
| CertificateParsingException | 1. Das Zertifikat enthält ungültige DER-Daten.<br>2. Das Zertifikat verwendet nicht unterstützte DER-Funktionen. | Stellen Sie sicher, dass das Zertifikat korrekt generiert und im IBM Key Management-Tool angezeigt werden kann, das mit MQIPT bereitgestellt wird. Prüfen Sie, ob Sie ein neues Zertifikat mit weniger Zertifikatserweiterungen abrufen können. |
| CertificateRevokedException | Die Zertifikatswiderrufsprüfung ist aktiviert und es wurde ermittelt, dass das Zertifikat widerrufen wurde. | Das betroffene Zertifikat sollte als nicht vertrauenswürdig eingestuft werden. Rufen Sie ein Ersatzzertifikat ab und stellen Sie sicher, dass das neue Zertifikat und der zugehörige private Schlüssel in der Schlüsselringdatei enthalten sind. |

| Ausnahmebedingung | Ursache | Aktion |
| --- | --- | --- |
| CertPathBuilderException | Die Zertifikatskette wurde nicht von einer anerkannten Zertifizierungsstelle signiert. | 1. Wenn Sie Zertifikate verwenden, die von einer Zertifizierungsstelle signiert sind, stellen Sie sicher, dass alle Zertifikate der Stammzertifizierungsstelle und CA-Zwischenzertifikate in der Schlüsselringdatei der Zertifizierungsstelle vorhanden sind.<br>2. Wenn Sie selbst signierte Zertifikate verwenden, stellen Sie sicher, dass eine Kopie des öffentlichen Teils des fernen Zertifikats extrahiert und der Schlüsselringdatei der Zertifizierungsstelle hinzugefügt wurde. Vermeiden Sie die Verwendung von selbst signierten Zertifikaten in Produktionsumgebungen. |
| CertStoreException<br>KeyStoreException | Beim Lesen eines Zertifikats auf einem Schlüsselring ist aus einem der folgenden Gründe ein Fehler aufgetreten:<br>1. Die Schlüsselringdatei ist beschädigt.<br>2. Die Schlüsselringdatei fehlt.<br>3. Das gespeicherte Kennwort stimmt nicht mit dem Kennwort der Schlüsselringdatei überein.<br>4. Wenn die Route für die Verwendung von Verschlüsselungshardware konfiguriert ist, konnte MQIPT keine Verbindung zur Verschlüsselungshardware herstellen. | 1. Stellen Sie sicher, dass die Schlüsselringdatei gelesen werden kann und alle Zertifikate mit dem IBM Key Management-Tool angezeigt werden können.<br>2. Stellen Sie sicher, dass alle Routeneigenschaften für den Schlüsselring auf den richtigen Dateinamen verweisen.<br>3. Stellen Sie sicher, dass das gespeicherte Kennwort für die Schlüsselringdatei richtig ist. Verwenden Sie das Tool **mqiptPW**, um das korrekte Kennwort zu speichern.<br>4. Wenn die Route für die Verwendung von Verschlüsselungshardware konfiguriert ist, überprüfen Sie Folgendes:<br>• In der Java-Sicherheitseigenschaftendatei ist angegeben, dass der Sicherheitsprovider IBMPKCS11Impl installiert ist.<br>• Die Datei mit den Java-Sicherheitseigenschaften enthält den vollständig qualifizierten Namen der Konfigurationsdatei, mit der der Sicherheitsprovider IBMPKCS11Impl initialisiert wird.<br>• Die Konfigurationsdatei, mit der der Sicherheitsprovider IBMPKCS11Impl initialisiert wird, ist gültig. |

| Ausnahmebedingung | Ursache | Aktion |
|---|---|---|
| SSLException: No available certificate or key corresponds to the SSL cipher suites which are enabled. | Sie benötigen ein persönliches Zertifikat mit dem korrekten Schlüsseltyp für die CipherSuites, die Sie verwenden. Beispielsweise ist für CipherSuites, deren Namen mit SSL_ECDH_ECDSA_ beginnen, ein Zertifikat mit einem öffentlichen Elliptic Curve-Schlüssel erforderlich. Für die am häufigsten verwendeten CipherSuites ist ein Zertifikat mit einem öffentlichen RSA-Schlüssel erforderlich. | Öffnen Sie die Schlüsselringdatei mit dem IBM Key Management-Tool. Wählen Sie in der Ansicht 'Personal Certificates' (Persönliche Zertifikate) nacheinander jedes Zertifikat aus und zeigen Sie es an. Klicken Sie auf **View Details** (Details anzeigen) und navigieren Sie zum Abschnitt 'Subject Public Key', um die Typen für den öffentlichen Schlüssel anzuzeigen. Überprüfen Sie anschließend die MQIPT-Routeneigenschaften **SSLClientCipherSuites** und **SSLServerCipherSuites**, um sicherzustellen, dass die passenden CipherSuites aktiviert sind. |
| SSLException: Keine gemeinsamen Cipher-Suites SSLHandshakeException: Keine gemeinsamen Cipher-Suites | Beim Handshake konnte keine CipherSuite vereinbart werden, da es keine Überschneidung zwischen der Gruppe von aktivierten CipherSuites an beiden Enden der Verbindung gibt. Insbesondere aktiviert eine ausgehende IBM MQ-Verbindung nur eine einzelne Verschlüsselung, wodurch dieser Fehler auf SSLServer MQIPT-Routen besonders wahrscheinlich ist.<br>Dieser Fehler kann auch auftreten, wenn die drei folgenden Bedingungen alle erfüllt sind:<br>• keine CipherSuite auf der Route angegeben<br>• kein geeignetes Sitezertifikat im für die Route konfigurierten Schlüsselring enthalten<br>• anonyme CipherSuites inaktiviert | Überprüfen Sie die Liste der aktivierten CipherSuites in den MQIPT-Routeneigenschaften **SSLClientCipherSuites** und **SSLServerCipherSuites**. Sie können auch weitere CipherSuites aktivieren. In der bereitgestellten Tabelle finden Sie Informationen, wie Sie die richtigen CipherSuites ermitteln, um den CipherSpec-Wert für den jeweiligen IBM MQ-Kanal zu aktivieren.<br>Wenn auf der Route keine CipherSuite angegeben ist, überprüfen Sie, ob sich die Eigenschaften der Schlüsselringroute auf die richtige Schlüsselringdatei beziehen und dass der Schlüsselring ein persönliches Zertifikat enthält, das von MQIPT verwendet werden kann. Wenn die Route für die Verwendung von Verschlüsselungshardware konfiguriert ist, stellen Sie sicher, dass das Attribut **tokenlabel** in der Konfigurationsdatei, mit dem der Sicherheitsprovider IBMPKCS11Impl initialisiert wird, die korrekte Tokenbezeichnung für eine Verschlüsselungseinheit angibt. |

## mqiptKeyman und mqiptKeycmd in MQIPT

Bei **mqiptKeyman** (iKeyman) handelt es sich um eine Zertifikats- und Schlüsselmanagementanwendung, die IBM MQ-Benutzern bereits vertraut ist. Mit den Befehlen **mqiptKeyman** und **mqiptKeycmd** können symmetrische und asymmetrische Schlüssel, digitale Zertifikate und Zertifikatsanforderungen in den von IBM MQ Internet Pass-Thru verwendeten Schlüsselringdateien verwaltet werden. Diese Dateien können auch verwendet werden, um die Schlüsselringdateien selbst zu verwalten.

Die Befehle **mqiptKeyman** und **mqiptKeycmd** verwenden den Begriff *Schlüsseldatenbank* , um auf eine Schlüsselringdatei zu verweisen. Diese Begriffe sind synonym.

iKeyman kann in zwei Modi ausgeführt werden, als grafische Benutzerschnittstelle (GUI) und über die Befehlszeilenschnittstelle (Command-Line Interface, CLI). Geben Sie den Befehl **mqiptKeyman** ein, um die GUI zu starten, und den Befehl **mqiptKeycmd**, um die CLI auszuführen.

Die entsprechenden Befehle zum Verwalten von Zertifikaten in IBM MQ sind **strmqikm** zum Starten der GUI und **runmqckm** zum Ausführen der CLI. Die IBM MQ-Befehle werden im Abschnitt **runmqckm**, **runmqakm** und **strmqikm** zum Verwalten von digitalen Zertifikaten verwenden beschrieben.

**Anmerkung:** V 9.3.4 Deprecated Die Tools **mqiptKeycmd** und **mqiptKeyman** werden ab IBM MQ 9.3.4nicht mehr verwendet. Sie finden weitere Informationen unter **runmqckm**, **runmqakm** und **strmqikm** für die Verwaltung von digitalen Zertifikaten verwenden.

## Für MQIPT erforderliches Schlüsselringdateiformat

Wenn Schlüsselringdateien für die Verwendung in MQIPT erstellen, müssen Sie das PKCS#12-Dateiformat verwenden:

- Wählen Sie in der Benutzerschnittstelle beim Erstellen der Schlüsselringdatei im Feld **Key database type** (Schlüsseldatenbanktyp) den Eintrag PKCS#12 aus.
- Geben Sie in der Befehlszeilenschnittstelle den Parameter `-type pkcs12` im Befehl `mqiptKeycmd -keydb -create` an.

MQIPT kann auch auf Zertifikate zugreifen, die in Verschlüsselungshardware gespeichert sind, die die PKCS #11-Schnittstelle unterstützt. Über die Schnittstelle können auch Zertifikate in PKCS #11-Hardware verwaltet werden. Weitere Informationen finden Sie unter „PKCS #11 -Verschlüsselungshardware in MQIPT verwenden” auf Seite 1143.

## Schlüsselringkennwort für MQIPT verschlüsseln

Nachdem Sie die Schlüsselringdatei erstellt haben, müssen Sie das Schlüsselringkennwort in einem Format verschlüsseln, das MQIPT für den Zugriff auf die Datei verwenden kann. Weitere Informationen hierzu finden Sie im Artikel „Schlüsselringkennwort in MQIPT verschlüsseln” auf Seite 1127.

Beachten Sie, dass die Stashdatei-Funktion von MQIPT nicht unterstützt wird. Zum Verschlüsseln des Schlüsselringkennworts müssen Sie statt einer Stashdatei den Befehl **mqiptPW** verwenden.

## Befehlszeilenbeispiele

Die CLI verwendet die gleiche Syntax wie der IBM MQ-Befehl **runmqckm**. Hängen Sie an **mqiptKeycmd** die erforderlichen Parameter an, wie in den folgenden Beispielen dargestellt:

- So erstellen Sie eine PKCS#12-Datei:

```
mqiptKeycmd -keydb -create -db key.p12 -pw password -type pkcs12
```

- So erstellen Sie zu Testzwecken ein selbst signiertes persönliches Zertifikat:

```
mqiptKeycmd -cert -create -db key.p12 -pw password -type pkcs12
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"
-sig_alg SHA256WithRSA -size 2048
```

  Der Befehl erstellt ein digitales Zertifikat mit einem 2048-Bit langen öffentlichen RSA-Schlüssel und einer digitalen Signatur, die RSA mit dem SHA-256-Hashalgorithmus verwendet. Achten Sie beim Erstellen eines Zertifikats darauf, dass Sie einen Verschlüsselungsalgorithmus für öffentliche Schlüssel, eine Schlüsselgröße und einen digitalen Signaturalgorithmus auswählen, die für die Sicherheitsanforderungen Ihres Unternehmens geeignet sind. Weitere Informationen finden Sie unter „Hinweise zu digitalen Zertifikaten für MQIPT” auf Seite 1134.

In diesem Beispiel wird ein selbst signiertes Zertifikat verwendet, das für Testzwecke geeignet ist. In einer Produktionsumgebung sollte jedoch stattdessen ein signiertes Zertifikat einer Zertifizierungsstelle verwendet werden.

Beachten Sie, dass MQIPT v2.0 und ältere Versionen digitale SHA-2-Signaturen nicht unterstützen. Daher ist dieses Zertifikat nicht für die Herstellung sicherer SSL-Verbindungen zu früheren MQIPT-Releases geeignet; ein älterer Signaturalgorithmus, wie z. B. SHA1WithRSA, wäre erforderlich.

- So erstellen Sie eine Zertifikatsanforderung für ein CA-signiertes Zertifikat zu Produktionszwecken:

```
mqiptKeycmd -certreq -create -db key.p12 -pw password -type pkcs12 -file cert.req
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"
-sig_alg SHA256WithRSA -size 2048
```

Der Befehl erstellt eine digitale Zertifikatsanforderung mit einem 2048 Bit langen öffentlichen RSA-Schlüssel und einer digitalen Signatur, die RSA mit dem SHA-256-Hashalgorithmus verwendet. Achten Sie beim Erstellen eines Zertifikats darauf, dass Sie einen Verschlüsselungsalgorithmus für öffentliche Schlüssel, eine Schlüsselgröße und einen digitalen Signaturalgorithmus auswählen, die für die Sicherheitsanforderungen Ihres Unternehmens geeignet sind. Weitere Informationen finden Sie unter „Hinweise zu digitalen Zertifikaten für MQIPT" auf Seite 1134.

- Gehen Sie wie folgt vor, um die CA signierte persönliche Zertifikatsdatei `cert.crt` in die Schlüsselringdatei zu empfangen:

```
mqiptKeycmd -cert -receive -db key.p12 -pw password -type pkcs12 -file cert.crt
```

Sie müssen sicherstellen, dass das CA-Zertifikat der Zertifizierungsstelle, die das persönliche Zertifikat signiert hat, in der CA-Schlüsselringdatei vorhanden ist, z. B.:

```
mqiptKeycmd -cert -add -db key.p12 -pw password -type pkcs12 -file ca.crt -label rootCA
```

### *Hinweise zu digitalen Zertifikaten für MQIPT*

Zu den zu berücksichtigende Punkten gehören die Größe des Zertifikatsschlüssels, die Auswahl eines geeigneten digitalen Signaturalgorithmus für Zertifikate und das digitale Zertifikat sowie das CipherSuite compatibilityDigital und die CipherSuite -Kompatibilität.

## Hinweise zur Größe des Zertifikatsschlüssels für MQIPT

Die Größe des öffentlichen Schlüssels hängt von der Sicherheitsrichtlinie Ihres Unternehmens und dem verwendeten Verschlüsselungsalgorithmus ab. Im Allgemeinen sind größere Schlüsselgrößen sicherer. In der folgenden Tabelle sind die Mindestschlüsselgrößen aufgelistet, die Sie verwenden sollten:

| **Algorithmus** | **Mindestschlüsselgröße (Bit)** |
|---|---|
| Elliptische Kurve | 256 |
| RSA | 2048 |

Geben Sie die Schlüsselgröße Ihres Zertifikats an, wenn Sie eine Zertifikat- oder Zertifikatsanforderung erstellen.

- Wenn Sie den CLI-Befehl **mqiptKeycmd** verwenden, gibt der Parameter **-size** die Schlüsselgröße an.
- Wenn Sie die grafische Benutzerschnittstelle **mqiptKeyman** verwenden, gibt das Feld **Key Size** (Schlüsselgröße) im Fenster 'Certificate Creation' (Zertifikatserstellung) die Schlüsselgröße an.

## Einen geeigneten Algorithmus für die digitale Signatur auswählen

Um die Fälschung von digitalen Zertifikaten zu verhindern, ist es wichtig, einen starken digitalen Signaturalgorithmus zu verwenden. Wenn Sie ein Zertifikat erstellen oder anfordern, müssen Sie darauf achten, einen guten Algorithmus auszuwählen.

Sie sollten die Verwendung von alten, auf MD5 oder SHA-1 basierenden digitalen Signaturalgorithmen vermeiden, da diese Algorithmen für die moderne Verwendung nicht mehr ausreichend sicher sind. Verwenden Sie nach Möglichkeit einen der neueren, SHA-2-basierten digitalen Signaturalgorithmen, wie z. B. SHA-256 mit RSA (SHA256WithRSA).

Von MQIPT-Versionen vor Version 2.1 werden digitale SHA-2-Signaturen allerdings nicht unterstützt; verwenden Sie daher im Interesse der Interoperabilität mit früheren MQIPT-Releases den Algorithmus für digitale Signaturen 'SHA1WithRSA'. Sie sollten jedoch die Aktualisierung älterer Versionen von MQIPT einplanen und die Verwendung von digitalen MD5- und SHA-1-Signaturen auslaufen lassen.

- Wenn Sie den CLI-Befehl **mqiptKeycmd** verwenden, gibt der Parameter **-sig_alg** den digitalen Signaturalgorithmus an.
- Wenn Sie die grafische Benutzerschnittstelle **mqiptKeyman** verwenden, gibt das Feld **Signature Algorithm** (Signaturalgorithmus) im Fenster 'Certificate Creation' (Zertifikatserstellung) den Algorithmus für digitale Signaturen an.

### Digitale Zertifikate und Cipher-Suite-Kompatibilität in MQIPT

Nicht alle Cipher-Suites können mit allen digitalen Zertifikaten verwendet werden. Es gibt verschiedene Arten von Cipher-Suites, die nach ihrem CipherSuite-Namenspräfix gruppiert sind. Jeder Typ von Cipher-Suite bringt verschiedene Einschränkungen für den Typ des verwendbaren digitalen Zertifikats mit sich. Diese Einschränkungen gelten für alle SSL-/TLS-Verbindungen von MQIPT, sind jedoch besonders für Benutzer der Elliptic Curve-Verschlüsselung relevant. Beim sicheren Socket-Handshakes wählt MQIPT automatisch ein persönliches Zertifikat aus, um sich selbst zu identifizieren, das für die ausgehandelte Cipher-Suite geeignet ist. In den meisten Fällen interagiert MQIPT automatisch mit dem fernen Peer.
In bestimmten Szenarios müssen Sie jedoch möglicherweise eine bestimmte MQIPT-CipherSuite verwenden, um mit einem fernen IBM MQ-System zu interagieren. Die Anwendung **mqiptKeyman**, die mit MQIPT bereitgestellt wird, ist nur noch in der Lage, Zertifikate und Zertifikatsanforderungen mit öffentlichen DSA- und RSA-Schlüsseln zu erstellen. Darüber hinaus kann das Dienstprogramm IBM MQ **runmqakm** Zertifikate und Zertifikatsanforderungen mit öffentlichen Elliptic Curve-Schlüsseln erstellen. Wenden Sie sich bei der Erstellung anderer Arten von Zertifikaten an Ihre Zertifizierungsstelle.

Der Typ des zu verwendenden digitalen Zertifikats hängt von dem Typ der von Ihnen verwendeten Cipher-Suite ab:

- Cipher-Suites, deren Namen mit SSL_ECDH_ECDSA_ und SSL_ECDHE_ECDSA_ anfangen, benötigen ein digitales Zertifikat mit einem öffentlichen Elliptic Curve-Schlüssel.
- Cipher-Suites, deren Namen ein *anon* enthalten, sind anonym; sie benötigen kein digitales Zertifikat, um den fernen Peer zu identifizieren. Solche Cipher-Suites können den Systemaufwand für dieZertifikatslebenszyklus-Verwaltung in Netzen vermeiden, in denen eine alternative Möglichkeit zur Authentifizierung verwendet wird; im Allgemeinen sollte ihre Verwendung aufgrund des Fehlens der Authentifizierung aber vermieden wird.
- Andere Cipher-Suites benötigen ein digitales Zertifikat mit einem öffentlichen RSA-Schlüssel.

**Anmerkung:** Die Tools **mqiptKeyman** und **mqiptKeycmd** sind nicht in der Lage, Zertifikate oder Zertifikatsanforderungen mit einem öffentlichen Schlüssel mit Elliptic Curve zu erstellen. Zu diesem Zweck können Sie den Befehl **runmqakm** verwenden, der mit IBM MQ bereitgestellt wird. Der Befehl **runmqakm** wird im Abschnitt **runmqckm**, **runmqakm** und **strmqikm** zum Verwalten von digitalen Zertifikaten verwenden beschrieben.

## Zertifikatsexit in MQIPT

Mit einem Zertifikatsexit wird ein SSL/TLS-Peerzertifikat geprüft, das von MQIPT empfangen wird.

Sie können eine MQIPT-Route so konfigurieren, dass sie als SSL/TLS-Client fungiert, wenn eine neue Verbindung hergestellt wird, und als SSL/TLS-Server, wenn eine Verbindungsanforderung empfangen wird. Während des SSL/TLS-Handshakeprozesses empfängt ein SSL/TLS-Client ein Peerzertifikat vom Server, mit dem der Server authentifiziert werden kann. Ein SSL/TLS-Server kann auch ein Peerzertifikat vom Client empfangen, mit dem der Client authentifiziert werden kann.

Der Zertifikatsexit wird aufgerufen, wenn MQIPT ein Peerzertifikat empfängt, mit dem eine weitere Prüfung ausgeführt werden kann. Alle Ausnahmebedingungen, die vom Exit abgefangen werden, werden von MQIPT abgefangen und die Verbindungsanforderung wird beendet. Es ist daher ein bewährtes Verfahren, dass der Exit alle Ausnahmebedingungen abruft und einen entsprechenden Rückgabecode an MQIPT übergibt.

Es wird ein Beispiel bereitgestellt, mit dem ein Zertifikatsexit für weitere Informationen angezeigt werden kann, siehe Zertifikatsexit für die Authentifizierung eines SSL/TLS-Servers verwenden.

**Anmerkung:** MQIPT wird in einer einzelnen Java Virtual Machine ausgeführt und daher kann ein benutzerdefiniertes Zertifikat möglicherweise die normale Operation von MQIPT auf eine der folgenden Arten gefährden:

- Auswirkungen auf Systemressourcen
- Generieren von Engpässen
- Vermindern der Leistung

Sie sollten die Auswirkungen Ihres Zertifikatsexits ausgiebig testen, bevor Sie es in einer Produktionsumgebung implementieren.

### *Die Klasse com.ibm.mq.ipt.exit.CertificateExit in MQIPT*

Die Klasse com.ibm.mq.ipt.exit.CertificateExit ist eine abstrakte Klasse, die von der Klasse implementiert werden muss, die mit der Eigenschaft SSLExitName definiert ist.

Die Klasse enthält Standardimplementierungen für die Ausführung des Exits sowie einige öffentliche Methoden, die Sie Ihren Anforderungen entsprechend optional außer Kraft setzen können. Die vollständige Liste der unterstützten Methoden lautet wie folgt:

## Methoden

**public int init(IPTTrace)**

Die Initialisierungsmethode wird von MQIPT aufgerufen, wenn der Exit von MQIPT geladen wird und implementiert werden kann, um eine Initialisierung des Exits durchzuführen (Beispiel: Laden von Daten, die während des Validierungsprozesses verwendet werden). Die Standardimplementierung führt keine Aktion aus.

**public int refresh(IPTTrace)**

Die Methode "refresh" wird implementiert, um eine Aktualisierung von Daten durchzuführen, z. B. das erneute Laden von Daten für eine Platte, die während des Validierungsprozesses verwendet wird. Diese Methode wird aufgerufen, wenn der MQIPT-Administrator einen Aktualisierungsbefehl abgesetzt hat. Die Standardimplementierung führt keine Aktion aus.

**public void close(IPTTrace)**

Die Methode "close" wird implementiert, um die gesamte Verwaltung auszuführen, wenn ein Stopp der Route bevorsteht oder MQIPT geschlossen wird. Die Standardimplementierung führt keine Aktion aus.

**public CertificateExitResponse validate(IPTTrace)**

Die Methode "validate" wird aufgerufen, um eine Validierung des Peerzertifikats vorzunehmen. Das Rückgabeobjekt kann verwendet werden, um Informationen an MQIPT zurückzugeben, beispielsweise einen Rückkehrcode und Text, der dem Verbindungsprotokoll hinzugefügt werden kann. Die Standardimplementierung gibt eine CertificateExitResponse mit CertificateExitResponse.OK zurück.

Unterstützte Methoden für den Abruf von Eigenschaften:

**public int getListenerPort()**
Ruft den Routen-Listener-Port ab, wie über die Eigenschaft "ListenerPort" definiert

**public String getDestination()**
Ruft die Zieladresse ab, wie über die Eigenschaft "Destination" definiert.

**public int getDestinationPort()**
Ruft die Ziellistener-Portadresse ab, wie über die Eigenschaft "DestinationPort" definiert.

**public String getClientIPAddress()**
Ruft die IP-Adresse des Clients ab, von dem die Verbindungsanforderung gestellt wurde

**public int getClientPortAddress()**
Ruft die Portadresse ab, die vom Client, der die Verbindungsanforderung stellt, verwendet wird

**public boolean isSSLClient()**
Wird verwendet, um zu ermitteln, ob der Exit als SSL/TLS-Client oder SSL/TLS-Server aufgerufen wird. Wenn "true" zurückgegeben wird, befindet sich der Exit auf der Clientseite der Verbindung und prüft das vom Server erhaltene Zertifikat. Wenn "false" zurückgegeben wird, befindet sich der Exit auf der Serverseite der Verbindung und prüft das vom Client gesendete Zertifikat. Eine Route kann sowohl als SSL/TLS-Server als auch als SSL/TLS-Client fungieren und dabei Datenverkehr entschlüsseln und erneut verschlüsseln. In dieser Situation werden einige Instanzen der Klasse als Clients und einige als Server aufgerufen, obwohl es nur eine einzige Exitklasse gibt. Mithilfe von isSSLClient können Sie herausfinden, welche Situation für eine bestimmte Instanz zutrifft.

**public int getConnThreadID()**
Wird verwendet, um die ID des Worker-Threads abzurufen, der die Verbindungsanforderung handhabt; dies kann beim Debugging hilfreich sein.

**public String getChannelName()**
Ruft den IBM MQ-Kanalnamen ab, der in der Verbindungsanforderung verwendet wird. Nur verfügbar, wenn die eingehende Anforderung nicht SSL/TLS verwendet und MQIPT als SSL/TLS-Client fungiert.

**public String getQMName()**
Ruft den Namen des IBM MQ-Warteschlangenmanagers ab, der in der Verbindungsanforderung verwendet wird. Nur verfügbar, wenn die Clientanforderung nicht SSL/TLS verwendet und MQIPT als SSL/TLS-Client fungiert.

**public boolean getTimedout()**
Wird vom Exit verwendet, um herauszfinden, ob das Zeitlimit abgelaufen ist.

**public IPTCertificate getCertificate()**
Ruft das SSL/TLS-Zertifikat ab, das geprüft werden muss.

**public String getExitData()**
Ruft die Exitdaten ab, wie über die Eigenschaft "SSLExitData" definiert.

**public String getExitName()**
Ruft den Exitnamen ab, wie über die Eigenschaft "SSLExitName" definiert.

### *Die Klasse com.ibm.mq.ipt.exit.CertificateExitResponse in MQIPT*

Mit dieser Klasse werden nach der Prüfung eines Zertifikats die Informationen wieder an MQIPT übergeben.

## Konstruktoren

**public CertificateExitResponse(*int rc*, Zeichenfolgenachricht)**

Dieser Konstruktor kann verwendet werden, um einen Rückkehrcode und einen Nachrichtentext zurückzugeben. Folgende Ursachencodes sind möglich:

- ExitRc.OK
- ExitRc.VALIDATE_ERROR
- ExitRc.VALIDATE_REJECTED

**public CertificateExitResponse(*int rc*)**

Dieser Konstruktor kann verwendet werden, um einen Rückkehrcode ohne Nachrichtentext zurückzugeben. Folgende Ursachencodes sind möglich:

- ExitRc.OK
- ExitRc.VALIDATE_ERROR

- ExitRc.VALIDATE_REJECTED

**public CertificateExitResponse()**

Dieser Konstruktor kann verwendet werden, um den Rückgabecode 'ExitRc.OK' ohne Nachrichtentext zurückzugeben.

## Methoden

**public String getVersion()**

Diese Methode gibt die Version dieser Klasse zurück.

**public String toString**

Diese Methode gibt eine Zeichenfolgedarstellung der Antwort zurück, beispielsweise: Reason code: 4, Message: Failed CRL check.

### *Die Klasse com.ibm.mq.ipt.exit.IPTCertificate in MQIPT*

Diese Klasse enthält das SSL/TLS-Zertifikat, das überprüft werden soll.

## Methoden

**public int getVersion()**

Diese Methode gibt die Version dieser Klasse zurück.

**public byte [] getDerEncoding()**

Diese Methode gibt die Codierung ASN.1/DER des X.509-Zertifikats zurück bzw. NULL, wenn ein Fehler aufgetreten ist.

**public byte [] getPemEncoding()**

Diese Methode gibt die Codierung PEM (BASE64) des X.509-Zertifikats zurück bzw. NULL, wenn ein Fehler aufgetreten ist.

**public String getLabel()**

Diese Methode gibt die Zertifikatsbezeichnung zurück bzw. NULL, wenn ein Fehler aufgetreten ist.

**public String getName()**

Diese Methode gibt den definierten Namen des Zertifikats zurück bzw. NULL, wenn dieser nicht verfügbar ist. For example:

```
CN=Test Queue Manager,OU=Sales,O=Example,L=London,C=GB
```

**public String getIssuerName()**

Diese Methode gibt den definierten Namen für den Aussteller des Zertifikats zurück bzw. NULL, wenn dieser nicht verfügbar ist. For example:

```
CN=Certificate Authority,OU=Security,O=Example,L=New York,C=US
```

**public IPTCertificate getSigner()**

Diese Methode gibt das Unterzeichnerzertifikat zurück bzw. NULL, wenn dieses nicht verfügbar ist. Bei einem selbst signierten Zertifikat wird ein Verweis auf das Zertifikat selbst zurückgegeben.

**public String toString()**

Diese Methode gibt eine Zeichenfolgedarstellung des Zertifikats zurück.

### *Die Klasse com.ibm.mq.ipt.exit.IPTTrace in MQIPT*

Die MQIPT -Tracefunktionen stellen Eingangs-und Exitaufrufe bereit, die beim Einstieg in eine Methode und beim Verlassen einer Methode verwendet werden können. Es gibt auch verschiedene Datenaufrufe, um nützliche Informationen zu verfolgen.

## Methoden

**public void entry(String *fid*)**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erfolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitteln.

Diese Methode schreibt einen Eingang in die Traceausgabedatei mit der entsprechenden Einrückungsstufe, um den Punkt aufzuzeichnen, an dem der Steuerungsfluss in eine Methode eintritt. Dieser Aufruf ist optional, wenn er jedoch verwendet wird, muss innerhalb derselben Methode auch ein entsprechender Aufruf an "exit(String)" verwendet werden.

**public void exit(String *fid*)**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erfolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitteln.

Diese Methode schreibt einen Exit in die Traceausgabedatei mit der entsprechenden Einrückungsstufe, um den Punkt aufzuzeichnen, an dem der Steuerungsfluss eine Methode verlässt. Diese Methode wird nur verwendet, wenn innerhalb derselben Methode zuvor ein Aufruf an "entry(String)" verwendet wurde.

**public void exit(String *fid*, int *rc*)**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitten, und *rc* ist der numerische Rückkehrcode von der Methode. Diese Tracemethode sollte verwendet werden, um den Exit von Methoden aufzuzeichnen, die eine ganze Zahl zurückgeben.

Diese Methode schreibt einen Exit in die Traceausgabedatei mit der entsprechenden Einrückungsstufe, um den Punkt aufzuzeichnen, an dem der Steuerungsfluss eine Methode verlässt, sowie den numerischen Rückkehrcode von dieser Methode. Diese Methode wird nur verwendet, wenn innerhalb derselben Methode zuvor ein Aufruf an "entry(String)" verwendet wurde.

**public void exit(String *fid*, boolean *rc*)**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erfolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitten, und *rc* ist der boolesche Rückkehrcode von der Methode. Diese Tracemethode sollte verwendet werden, um den Exit von Methoden aufzuzeichnen, die einen booleschen Wert zurückgeben.

Diese Methode schreibt einen Exit in die Traceausgabedatei mit der entsprechenden Einrückungsstufe, um den Punkt aufzuzeichnen, an dem der Steuerungsfluss eine Methode verlässt, und den booleschen Rückkehrcode von dieser Methode. Diese Methode wird nur verwendet, wenn innerhalb derselben Methode zuvor ein Aufruf an "entry(String)" verwendet wurde.

**public void data(String *fid*, String data)**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erfolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitteln.

Diese Methode schreibt Zeichenfolgedaten in die Traceausgabedatei.

**public void data(String *fid*, int data)**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erfolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitteln.

Diese Methode schreibt ganzzahlige Daten in die Traceausgabedatei.

**public void data(String *fid*, byte[])**

Dabei wird *fid* verwendet, um herauszufinden, wo der Aufruf erfolgt ist, um also beispielsweise den Klassen- und Methodennamen zu ermitteln.

Diese Methode schreibt binäre Daten in die Traceausgabedatei.

## Beispieltrace

Zur leichteren Diagnose von Problemen in einem Exit können Sie dieselbe Tracefunktion wie MQIPT verwenden oder alternativ dazu eigene Tracefunktionen implementieren. Wenn Sie sich für die Verwendung der MQIPT-Tracefunktionen entscheiden, gibt es Eingangs- und Exitaufrufe, die beim Eingang in eine Methode und beim Verlassen einer Methode verwendet werden können. Es gibt auch verschiedene Datenaufrufe zum Verfolgen nützlicher Informationen, wie im folgenden Beispiel gezeigt.

```
/**
* This method is called to initialize the exit (for example, for
* loading validation information) and place itself in a ready
* state to validate connection requests.
*/
 public int init(IPTTrace t) {
 final String fid = "MyExit.init";

 // Trace entry into this method
 t.entry(fid);

 // Trace useful information
 t.data(fid, "Starting exit - MQIPT version " + getVersion());

 // Perform initialization and load any data
 t.data(fid, "Ready for work");

 // Trace exit from this method
 t.exit(fid);

 return ExitRc.OK;
}
```

Diese Methode erstellt einen Trace in dem im folgenden Beispiel dargestellten Format:

```
16:36:48.625    14   5000-1s    ------{ ConnectionThread.setCertificateExit()
16:36:48.625    14   5000-1s     Creating instance of certificate exit
16:36:48.625    14   5000-1s     Calling init() of certificate exit
16:36:48.625    14   5000-1s    -------{ MyExit.init()
16:36:48.625    14   5000-1s     Starting exit - MQIPT version 2.1.0.0
16:36:48.625    14   5000-1s    Ready for work
16:36:48.625    14   5000-1s    -------} MyExit.init() rc=0
16:36:48.625    14   5000-1s    ------} ConnectionThread.setCertificateExit() rc=0
```

### *Rückgabecodes für Zertifikatsexits in MQIPT*

Rückgabecodes, die MQIPT erkennt, wenn ein Zertifikatsexit in verschiedenen Situationen aufgerufen wird

Nachfolgende Rückgabecodes werden in folgenden Situationen von MQIPT beim Aufruf eines Zertifikatsexits erkannt:

| Rückgabecode | Beschreibung | init | Validieren | Aktualisierung |
|---|---|---|---|---|
| ExitRc.OK | Anforderung erfolgreich abgeschlossen. | ja | ja | ja |
| ExitRc.INIT_ERROR | Init-Anforderung fehlgeschlagen, Route wird inaktiviert. | ja | | |
| ExitRc.REFRESH_ERROR | Aktualisierungsanforderung fehlgeschlagen, Route wird inaktiviert. | | | ja |

| Rückgabecode | Beschreibung | init | Validieren | Aktualisierung |
|---|---|---|---|---|
| ExitRc.VALIDATE_ERROR | Validierungsprozess fehlgeschlagen, Verbindungsanforderung abgelehnt. | | ja | |
| ExitRc.VALIDATE_REJECTED | Validierungsanforderung abgelehnt, Verbindungsanforderung abgelehnt. | | ja | |

## LDAP und CRLs in MQIPT

MQIPT unterstützt die Verwendung eines LDAP-Servers (Lightweight Directory Access Protocol), um die Zertifikatswiderrufslisten-Authentifizierung (Certificate Revocation List, CRL) für ein digitales Zertifikat durchzuführen.

Die LDAP-Unterstützung wurde in ähnlicher Weise wie in IBM MQ implementiert, da derselbe LDAP-Server sowohl für IBM MQ als auch für MQIPT verwendet werden kann.

Während des SSL/TLS-Handshakes authentifizieren sich die kommunizierenden Partner gegenseitig mit digitalen Zertifikaten. Die Authentifizierung kann eine Überprüfung enthalten, dass das empfangene Zertifikat immer noch vertrauenswürdig ist. Zertifizierungsstellen (CAs) entziehen Zertifikate aus verschiedenen Gründen, einschließlich der folgenden:

- Der Eigner wurde in eine andere Organisation verschoben.
- Der private Schlüssel ist nicht mehr geheim.

CAs veröffentlichen widerrufliche persönliche Zertifikate in einer Zertifikatswiderrufsliste (Certificate Revocation List, CRL). CA-Zertifikate, die widerrufen wurden, werden in einer Berechtigungswiderrufsliste (ARL, Authority Revocation List, Berechtigungswiderrufsliste) veröffentlicht. Beachten Sie, dass nachfolgende Verweise auf CRLs auch für ARLs gelten.

Weitere Informationen zur Verwendung von LDAP-Servern mit IBM MQ und zur Verwaltung von CRLs und ARLs finden Sie im Abschnitt Mit Zertifikatswiderrufslisten und Berechtigungswiderrufslisten arbeiten.

MQIPT kann bis zu zwei LDAP-Server auf jeder Route unterstützen. Der erste LDAP-Server wird als Hauptserver behandelt, wobei der zweiter LDAP-Server als Backup dient. Der zweite Server wird nur verwendet, wenn der Hauptserver nicht erreicht werden kann. Der Sicherungsserver sollte ein Spiegelimage des Hauptservers sein.

Der Zugriff auf Informationen, die auf einem LDAP-Server gespeichert sind, kann mit einer Benutzer-ID und einem Kennwort unter Verwendung von LDAP-Benutzer-ID und -Kennwort geschützt werden. Kennwörter für LDAP-Server können in der MQIPT -Konfiguration von IBM MQ 9.1.5verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern, die von MQIPTverwendet werden sollen, finden Sie unter „Gespeicherte Kennwörter in MQIPT verschlüsseln” auf Seite 1156.

Wenn MQIPT ein PKCS#12-Token aus einer Schlüsselringdatei lädt, werden alle CA-Zertifikate auf die CRL-Gültigkeit überprüft. Wenn das CA-Zertifikat über eine angehängte CRL verfügt, wird überprüft, ob es abgelaufen ist, und wenn ja, wird eine neuere CRL vom LDAP-Server abgerufen. Die abgerufenen CRLs werden ggf. in das aktuelle Token geladen und an das zugehörige CA-Zertifikat angehängt.

Wenn beim Senden einer Abfrage an den LDAP-Server keine Einträge vorhanden sind, die mit der angegebenen Zertifizierungsstelle übereinstimmen, wird davon ausgegangen, dass es keine CRLs für diese Zertifizierungsstelle gibt, und der Sicherungsserver wird nicht verwendet. Wenn der Haupt-LDAP-Server jedoch nicht innerhalb eines bestimmten Zeitrahmens erreicht werden kann oder keine Antwort zurückgibt, wird der Sicherungsserver verwendet. Eventuelle vom Sicherungsserver ausgehende Fehlermeldungen führen dazu, dass die Clientverbindung beendet wird. Diese Aktion kann überschrieben werden, indem die Eigenschaft **LDAPIgnoreErrors** auf `true` gesetzt wird.

Alle durch MQIPT abgerufenen CRLs werden in einem Cache gespeichert und von allen Verbindungen auf dieser Route gemeinsam genutzt. Wenn eine zwischengespeicherte CRL abgelaufen ist, wird die CRL aus dem Cache entfernt, und eine neue CRL wird aus dem LDAP-Server abgerufen. Ist keine neue CRL verfügbar, wird der Verbindungsaufbau verweigert.

Eine vom LDAP-Server abgerufene CRL wird ebenfalls auf Ablauf überprüft und eine Warnung wird angezeigt (MQCPW001). Die abgelaufene CRL wird dennoch in das System geladen und Verbindungsanforderung, die auf diese CRL verweisen, werden abgelehnt. Sie sollten die abgelaufene CRL im LDAP-Server durch eine aktuelle ersetzen.

Die Eigenschaft **LDAPCacheTimeout** kann verwendet werden, um zu steuern, wie oft der CRL-Cache gelöscht wird. Der Standardwert ist 1 Tag. Wird dieser Wert auf `0` gesetzt, werden die Cacheeinträge erst gelöscht, wenn die Route neu gestartet wird.

Eine abgelaufene CRL kann in einer Schlüsselringdatei oder auf einem LDAP-Server gespeichert werden. Wurde keine neue CRL ausgegeben, werden alle weiteren Verbindungsanforderung abgelehnt. Sie können abgelaufene CRLs ignorieren, indem Sie die Eigenschaft **IgnoreExpiredCRLs** aktivieren.

**Anmerkung:** Wenn Sie entweder die Eigenschaft **LDAPIgnoreErrors** oder die Eigenschaft **IgnoreExpiredCRLs** aktivieren, kann ein widerrufenes Zertifikat dazu verwendet werden, eine SSL/TLS-Verbindung herzustellen.

## Mehrwertige Eigenschaften der Zertifikats-DN OU in MQIPT

Sie können mehrere Werte von Organisationseinheiten (Organizational Units, OU) in definierten Namen von Zertifikaten abgleichen.

Die folgenden Routeneigenschaften unterstützen jetzt den Abgleich mehrerer OU-Werte:

- **SSLClientDN_OU**
- **SSLClientSiteDN_OU**
- **SSLServerDN_OU**
- **SSLServerSiteDN_OU**

Für den Abgleich mehrerer OU-Werte verwenden Sie ein Komma als Trennzeichen im Wert für die Routeneigenschaft. For example:

```
SSLClientDN_OU=Sales, Europe
```

Hierdurch werden Zertifikate abgeglichen, die `OU=Sales` und `OU=Europe` enthalten. Die OU-Werte werden in der gleichen Instanz wie mehrere OU-Werte in IBM MQ SSLPEER-Filtern abgeglichen.

Geben Sie eine Routeneigenschaft im Abschnitt `[route]` nur einmal an. Die korrekte Vorgehensweise beim Abgleich mehrerer OU-Werte ist die einmalige Angebe der Eigenschaft, wie im vorherigen Beispiel gezeigt. Wenn Sie dasselbe Attribut mehr als einmal in demselben `mqipt.conf`-Abschnitt eingeben, wird der letzte Wert wirksam. Bei den folgenden Einträge wird beispielsweise nur der Wert `Europe` abgeglichen, da die erste Zeile von der zweiten Zeile überschrieben wird:

```
SSLClientDN_OU=Sales
SSLClientDN_OU=Europe
```

Wenn Sie ein Literalkomma in einem OU-Wert abgleichen müssen, fügen Sie direkt vor dem Komma einen Backslash (\) als Escapezeichen ein. For example:

```
SSLClientDN_OU=Sales\, Europe
```

Dies entspricht einem einzigen Wert: `OU=Sales, Europe`. Ein Backslash, auf den nicht direkt ein Komma folgt, entspricht einem Backslash als Literal.

Wenn Sie ein Upgrade von einem früheren Release von MQIPT ausführen und weiterhin Kommas in OU-Werten abgleichen müssen, müssen Sie Backslashes als Escapezeichen in die OU-Routeneigenschaften einfügen, damit das bisher genutzte Verhalten beibehalten wird.

## Deprecated Veraltete Protokolle und Cipher Suites in MQIPT aktivieren

Standardmäßig sind Secure Sockets-Protokolle und Cipher-Suites, die als unsicher betrachtet werden, in der Java runtime environment (JRE) inaktiviert, die mit MQIPTbereitgestellt wird. Diese veralteten Protokolle und CipherSuites müssen erst aktiviert werden, damit sie verwendet werden können.

### Informationen zu diesem Vorgang

Wenn Sie sich der potenziellen Gefahren bewusst sind, aber immer noch eines dieser Protokolle oder eine der CipherSuites verwenden müssen, die in MQIPT als unsicher betrachtet werden, folgen Sie dieser Prozedur, um das erforderliche Protokoll oder die CipherSuite zu aktivieren.

**Anmerkung:** Veraltete Protokolle und CipherSuites können nicht mit dem TLS-Befehlsport verwendet werden.

### Vorgehensweise

1. Bearbeiten Sie die Datei `java.security`, die sich im Verzeichnis *mqipt_path*`/java/jre/lib/security` befindet. Dabei ist *mqipt_path* die Position, an der MQIPT installiert ist.
2. Fügen Sie der JRE Unterstützung für ein Protokoll oder einen Algorithmus hinzu, indem Sie den entsprechenden Eintrag aus der Liste der inaktivierten Algorithmen in der Eigenschaft `jdk.tls.disabledAlgorithms` entfernen.
   - Um die Unterstützung für ein Protokoll hinzuzufügen, entfernen Sie das Protokoll aus der Liste der inaktivierten Algorithmen. Wenn Sie beispielsweise Unterstützung für TLS 1.0 hinzufügen möchten, entfernen Sie TLSv1 aus der Liste.
   - Um die Unterstützung für eine CipherSuite hinzuzufügen, entfernen Sie die zugehörigen Algorithmen aus der Liste der inaktivierten Algorithmen. Um beispielsweise Unterstützung für die SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA -Cipher-Suite hinzuzufügen, entfernen Sie 3DES_EDE_CBC und DESede aus der Liste.
3. Um SSL 3.0 in der JRE zu aktivieren, muss auch die Systemeigenschaft `com.ibm.jsse2.disableSSLv3=false` festgelegt werden.

   Wenn Sie MQIPT über die Befehlszeile mit dem Befehl **mqipt** starten, können Sie die Eigenschaft mithilfe der Umgebungsvariablen **MQIPT_JVM_OPTIONS** festlegen. For example:

   ```
   set MQIPT_JVM_OPTIONS=-Dcom.ibm.jsse2.disableSSLv3=false
   ```

   Windows Wenn MQIPT als Windows -Dienst installiert ist, können Sie die Eigenschaft festlegen, indem Sie einen Zeichenfolgewert in der Windows -Registrierung unter dem Schlüssel `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MQInternetPassThru` definieren. Der Wert sollte die folgenden Attribute haben:

   **Name**
   MqiptJvmOptions

   **Wert**
   -Dcom.ibm.jsse2.disableSSLv3=false
4. Um SSL 3.0, TLS 1.0 oder TLS 1.1 in einer MQIPT -Route zu aktivieren, fügen Sie das entsprechende Protokoll zur Routeneigenschaft **SSLServerProtocols** bzw. **SSLClientProtocols** hinzu.
5. Starten Sie MQIPT erneut, damit die Änderungen an den JRE-Eigenschaften wirksam werden.

## PKCS #11 -Verschlüsselungshardware in MQIPT verwenden

MQIPT kann auf digitale Zertifikate zugreifen, die in Verschlüsselungshardware gespeichert sind, die die PKCS #11-Schnittstelle unterstützt.

## Vorbereitende Schritte

Bevor Sie mit der Konfiguration von MQIPT für die Verwendung von Verschlüsselungshardware beginnen, stellen Sie sicher, dass die Verschlüsselungskarte, der Kartentreiber und die gesamte zugehörige Unterstützungssoftware installiert sind und ordnungsgemäß funktionieren.

Die Unterstützung der PKCS #11-Verschlüsselungshardware in MQIPT wird durch den IBM Java PKCS11-Verschlüsselungsprovider (Provider IBMPKCS11Impl) bereitgestellt. Weitere Informationen zum Provider IBMPKCS11Impl und die Liste der Verschlüsselungskarten, die von Java 8 unterstützt werden, finden Sie unter IBM PKCS11-Verschlüsselungsprovider.

## Informationen zu diesem Vorgang

Sie können die persönlichen Zertifikate und die CA-Zertifikate, auf die MQIPT zugreift, in einem Schlüsselspeicher für die Verschlüsselungshardware speichern. Da eine PKCS #11-Einheit normalerweise jedoch nicht ausreichend Speicherplatz zum Speichern einer großen Menge von Unterzeichnerzertifikaten bietet, können Sie auch einen separaten dateibasierten Schlüsselspeicher für CA-Zertifikate verwenden.

Führen Sie diese Prozedur aus, um MQIPT für die Verwendung von Zertifikaten in einem Schlüsselspeicher für die Verschlüsselungshardware zu konfigurieren.

**Anmerkung:** Bei der Verwendung von Verschlüsselungshardware mit MQIPT handelt es sich um eine IBM MQ Advanced-Funktion. Um diese Funktion zu verwenden, muss der lokale Warteschlangenmanager, der über die MQIPT -Route verbunden ist, ebenfalls über die Berechtigung IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS VUEoder IBM MQ Advanced for z/OS verfügen.

## Vorgehensweise

1. Erstellen Sie die Konfigurationsdatei, die bei der Initialisierung des IBMPKCS11Impl-Providers verwendet wird.

   Laden Sie Beispielkonfigurationsdateien für jede der Hardwareverschlüsselungskarten herunter, die vom IBMPKCS11Impl-Provider unterstützt werden, und konfigurieren Sie ein Beispiel für Ihr System. Die Beispiele können aus dem folgenden Abschnitt in der IBM Documentation für Java heruntergeladen werden: Konfigurationsdatei.

   Bei der Konfigurationsdatei handelt es sich um eine Textdatei, die mindestens die folgenden Attribute enthalten sollte:

   **Name**
   Das Namenssuffix der Providerinstanz.

   **Bibliothek**
   Der vollständig qualifizierte Name der PKCS #11-Bibliothek, die mit der Verschlüsselungshardware bereitgestellt wird.

   **tokenlabel**
   Die Tokenbezeichnung der PKCS #11-Verschlüsselungseinheit.

   Die Konfigurationsdatei kann beispielsweise die folgenden Einträge enthalten:

   ```
   name = IPTPKCS11Provider
   library = /usr/lib64/pkcs11/PKCS11_API.so
   tokenlabel = icatoken
   ```

2. Bearbeiten Sie die Sicherheitseigenschaftendatei von Java, `java.security`, die sich im Unterverzeichnis `java/jre/lib/security` des MQIPT-Installationsverzeichnisses befindet.
   a) Fügen Sie den Sicherheitsprovider IBMPKCS11Impl hinzu, falls er noch nicht in der Datei vorhanden ist.

      Sie können dazu beispielsweise die folgende Zeile hinzufügen:

      ```
      security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
      ```

   b) Fügen Sie den vollständig qualifizierten Namen der Konfigurationsdatei nach dem Providernamen hinzu.

Wenn beispielsweise die Konfigurationsdatei, die Sie in Schritt „1” auf Seite 1144 erstellt haben, `/opt/mqipt/pkcs11.cfg` heißt, sollten Sie diesen Pfad zur gleichen Zeile wie der Sicherheitsprovider hinzufügen:

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/mqipt/
pkcs11.cfg
```

3. Wenn Sie CA-Zertifikate nicht in der Verschlüsselungshardware speichern, sondern eine Schlüsselringdatei für CA-Zertifikate verwenden, erstellen Sie eine CA-Schlüsselringdatei im PKCS #12-Format.

   Sie können eine CA-Schlüsselringdatei entweder über die grafische Benutzerschnittstelle von **mqiptKeyman** oder über die Befehlszeilenschnittstelle von **mqiptKeycmd** erstellen.

   - Für die Verwendung der CLI geben Sie den folgenden Befehl ein:

     ```
     mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
     ```

     Dabei ist *Dateiname* der Name der Schlüsselringdatei, die erstellt werden soll, und *Kennwort* ist das Schlüsselringkennwort.
   - Führen Sie für die Verwendung der GUI diese Schritte aus:
     a. Starten Sie die GUI mit dem Befehl **mqiptKeyman**.
     b. Klicken Sie auf **Schlüsseldatenbankdatei** > **Öffnen**.
     c. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **PKCS11Config** aus.
     d. Klicken Sie auf **OK**. Das Fenster 'Open Cryptographic Token' wird geöffnet.
     e. Wählen Sie die Tokenbezeichnung für die Verschlüsselungseinheit aus, unter der Sie die Zertifikate speichern möchten.
     f. Geben Sie im Feld **Cryptographic Token Password** (Kennwort für Verschlüsselungstoken) das Kennwort ein, das für den Zugriff auf die Verschlüsselungshardware erforderlich ist.
     g. Zum Erstellen einer neuen CA-Schlüsselringdatei wählen Sie **Create new secondary key database file** (Neue sekundäre Schlüsseldatenbankdatei erstellen) aus.
     h. Klicken Sie auf **Key database type** (Schlüsseldatenbanktyp) und wählen Sie **PKCS12** aus.
     i. Geben Sie im Feld **File Name** (Dateiname) den Dateinamen des CA-Schlüsselrings ein.
     j. Geben Sie im Feld **Location** (Position) den vollständigen Pfad zur CA-Schlüsselringdatei ein.
     k. Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird geöffnet.
     l. Geben Sie im Feld **Password** (Kennwort) ein Kennwort für den CA-Schlüsselring ein und wiederholen Sie die Eingabe im Feld **Confirm Password** (Kennwort bestätigen).
     m. Klicken Sie auf **OK**.
4. Fordern Sie mit **mqiptKeycmd** oder **mqiptKeyman** ein persönliches Zertifikat für die Verschlüsselungshardware an.

   - Für die Verwendung der CLI geben Sie den folgenden Befehl ein:

     ```
     mqiptKeycmd -certreq -create -crypto module_name -tokenlabel hardware_token
                 -pw password -label label -size key_size
                 -sig_alg algorithm -dn distinguished_name -file filename
     ```

     Dabei gilt:

     **-crypto *Modulname***
     Gibt den vollständig qualifizierten Namen der PKCS #11-Bibliothek an, die mit der Verschlüsselungshardware geliefert wird.

     **-tokenlabel *Tokenbezeichnung***
     Gibt die Tokenbezeichnung für die PKCS #11-Verschlüsselungseinheit an.

     **-pw *password***
     Gibt das Kennwort für den Zugriff auf die Verschlüsselungshardware an.

**-label *Bezeichnung***
Gibt die Zertifikatsbezeichnung an.

**-size *Schlüsselgröße***
Gibt die Schlüsselgröße an. Die Werte 512, 1024, 2048 oder 4096 sind zulässig.

**-sig_alg *Algorithmus***
Gibt den asymmetrischen Signaturalgorithmus an, der zum Erstellen des Schlüsselpaars für den Eintrag verwendet wird. Folgende Werte sind möglich: MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA oder SHAWithDSA. Der Standardwert ist SHA256WithRSA.

**-dn *definierter_Name***
Gibt den definierten X.500-Namen in doppelte Anführungszeichen an.

**-file *Dateiname***
Gibt den Dateinamen für die Zertifikatsanforderung an.

- Führen Sie für die Verwendung der GUI diese Schritte aus:
  a. Klicken Sie im Menü **Erstellen** auf **Neue Zertifikatsanforderung** .
  b. Geben Sie in das Feld **Schlüsselkennsatz** die Zertifikatsbezeichnung ein.
  c. Wählen Sie die erforderliche **Schlüsselgröße** und den **Signaturalgorithmus** aus.
  d. Geben Sie Werte für **Common Name** und **Organization** ein, und wählen Sie ein **Land** aus. Geben Sie für die verbleibenden optionalen Felder entweder die Standardwerte an, oder geben Sie neue Werte ein oder wählen Sie neue Werte aus.
  e. Geben Sie im Feld **Geben Sie den Namen einer Datei ein, in der die Zertifikatsanforderung gespeichert werden soll** entweder den Standardwert certreq.armein, oder geben Sie einen neuen Wert mit einem vollständigen Pfad ein.
  f. Klicken Sie auf **OK**.
  g. In der Liste **Persönliche Zertifikatsanforderungen** wird die Bezeichnung der neuen persönlichen Zertifikatsanforderung angezeigt, die Sie erstellt haben. Die Zertifikatsanforderung wird in der von Ihnen ausgewählten Datei gespeichert.

5. Nachdem die Zertifizierungsstelle Ihnen das persönliche Zertifikat gesendet hat, fügen Sie das CA-Zertifikat dem Speicher für den Verschlüsselungsschlüssel oder der CA-Schlüsselringdatei hinzu, wenn es nicht bereits vorhanden ist.
   - Um die CLI zum Hinzufügen des CA-Zertifikats zur CA-Schlüsselringdatei zu verwenden, geben Sie den folgenden Befehl ein:

     ```
     mqiptKeycmd -cert -add -db filename -pw password -type pkcs12
                 -label label -file cert_filename
     ```

     Dabei ist *Dateiname* der Name der CA-Schlüsselringdatei, *Kennwort* ist das Kennwort für den CA-Schlüsselring, *Bezeichnung* ist die an das Zertifikat angeschlossene Bezeichnung und *Zertifikatsdateiname* ist der Name der Datei mit dem CA-Zertifikat.
   - Um die CLI zum Hinzufügen des CA-Zertifikats zur Verschlüsselungshardware zu verwenden, geben Sie den folgenden Befehl ein:

     ```
     mqiptKeycmd -cert -add -crypto module_name -tokenlabel hardware_token
                 -pw password -label label -file cert_filename
     ```

     Dabei ist *Modulname* der vollständig qualifizierte Name der PKCS #11-Bibliothek, die mit der Verschlüsselungshardware bereitgestellt wird, *Hardwaretoken* ist die Tokenbezeichnung der PKCS #11-Verschlüsselungseinheit, *Kennwort* ist das Kennwort für den Zugriff auf die Verschlüsselungshardware, *Bezeichnung* ist die an das Zertifikat angeschlossene Bezeichnung und *Zertifikatsdateiname* ist der Name der Datei mit dem CA-Zertifikat.

- Führen Sie für die Verwendung der GUI diese Schritte aus:
  a. Wählen Sie im Feld **Key database content** die Option **Signer Certificates** aus.
  b. Klicken Sie auf **Hinzufügen**. Das Fenster CA-Zertifikat aus einem Datei hinzufügen wird geöffnet.
  c. Geben Sie den Namen und die Position der Zertifikatsdatei ein, in der das Zertifikat gespeichert ist, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
  d. Klicken Sie auf **OK**. Das Fenster "Enter a Label" wird geöffnet.
  e. Geben Sie im Fenster "Enter a Label" den Namen des Zertifikats ein.
  f. Klicken Sie auf **OK**. Das Zertifikat wird der Schlüsseldatenbank hinzugefügt.

6. Empfangen Sie das von der Zertifizierungsstelle bereitgestellte persönliche Zertifikat im Schlüsselspeicher für die Verschlüsselungshardware.
   - Für die Verwendung der CLI geben Sie den folgenden Befehl ein:

     ```
     mqiptKeycmd -cert -receive -file filename -crypto module_name
                 -tokenlabel hardware_token -pw password
     ```

     Dabei ist *Dateiname* der Name der Datei mit dem zu empfangenden Zertifikat, *Modulname* ist der vollständig qualifizierte Name der PKCS #11-Bibliothek, die mit der Verschlüsselungshardware bereitgestellt wird, *Hardwaretoken* ist die Tokenbezeichnung der PKCS #11-Verschlüsselungseinheit und *Kennwort* ist das Kennwort für den Zugriff auf die Verschlüsselungshardware.

     Wenn das CA-Zertifikat nicht in der Verschlüsselungshardware, sondern in einem CA-Schlüsselring gespeichert wird, erhalten Sie eine Warnung, dass die Zertifikatskette nicht geprüft werden kann, da der Befehl **mqiptKeycmd** beim Empfang des persönlichen Zertifikats im Speicher des Verschlüsselungsschlüssels nicht auf den CA-Schlüsselring zugreifen kann.
   - Führen Sie für die Verwendung der GUI diese Schritte aus:
     a. Klicken Sie auf **Empfangen** . Das Fenster 'Receive Certificate from a File' (Zertifikat aus einer Datei empfangen) wird angezeigt.
     b. Geben Sie den Namen und die Position der Zertifikatsdatei für das neue persönliche Zertifikat ein, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
     c. Klicken Sie auf **OK**. Im Feld **Persönliche Zertifikate** wird die Bezeichnung des neuen persönlichen Zertifikats angezeigt, das Sie hinzugefügt haben.
7. Verschlüsseln Sie das Kennwort für den Zugriff auf die Verschlüsselungshardware mit dem Befehl **mqiptPW**.

   Geben Sie den folgenden Befehl ein:

   ```
   mqiptPW -sf encryption_key_file
   ```

   Dabei steht *Verschlüsselungsschlüsseldatei* für den Namen einer Datei, die den Kennwortverschlüsselungsschlüssel für Ihre MQIPT-Installation enthält. Sie müssen den Parameter **-sf** nicht angeben, wenn Ihre MQIPT-Installation den Standardkennwortverschlüsselungsschlüssel verwendet. Geben Sie bei einer entsprechenden Aufforderung das Kennwort für den Zugriff auf die Verschlüsselungshardware ein, das verschlüsselt werden soll.

   Weitere Informationen zur Verschlüsselung von Schlüsselspeicherkennwörtern finden Sie unter „Schlüsselringkennwort in MQIPT verschlüsseln” auf Seite 1127.
8. Wenn Sie in Schritt „3” auf Seite 1145 eine CA-Schlüsselringdatei erstellt haben, verschlüsseln Sie das Kennwort für die CA-Schlüsselringdatei, indem Sie die Anweisungen in Schritt „7” auf Seite 1147 befolgen.
9. Bearbeiten Sie die `mqipt.conf`-Konfigurationsdatei.
   a) Vergewissern Sie sich, dass Sie über die entsprechende Berechtigung zur Verwendung dieser IBM MQ Advanced-Funktion verfügen, indem Sie die globale Eigenschaft **EnableAdvancedCapabilities** auf `Wahr` setzen.

b) Aktivieren Sie die Verwendung des Schlüsselspeichers für Verschlüsselungshardware in der Route, indem Sie eine oder mehrere der Eigenschaften **SSLServerKeyRingUseCryptoHardware**, **SSLServerCAKeyRingUseCryptoHardware**, **SSLServerKeyRingUseCryptoHardware** oder **SSLServerKeyRingUseCryptoHardware** auf `Wahr` setzen.

   Weitere Informationen zu den Eigenschaften, mit denen die Verwendung der Verschlüsselungshardware in einer Route aktiviert wird, finden Sie unter MQIPT-Routeneigenschaften.

   Ab IBM MQ 9.2.0 können Sie auch Verschlüsselungshardware mit dem TLS-Befehlsport verwenden, indem Sie die Eigenschaft **SSLCommandPortKeyRingUseCryptoHardware** auf `Wahr` setzen.

c) Wenn Sie eine Schlüsselringdatei für CA-Zertifikate verwenden, geben Sie die Position des CA-Schlüsselrings an, indem Sie eine oder mehrere der Eigenschaften **SSLServerCAKeyRing** oder **SSLServerCAKeyRing** festlegen.

   Wenn Sie eine Route zur Verwendung von Verschlüsselungshardware für das Sitezertifikat konfiguriert haben und keine CA-Schlüsselringdatei angeben, wird der Schlüsselspeicher der Verschlüsselungshardware als CA-Schlüsselspeicher verwendet.

d) Geben Sie mit der Eigenschaft **SSLServerKeyRingPW**, **SSLServerCAKeyRingPW**, **SSLClientKeyRingPW**, **SSLClientCAKeyRingPW** oder **SSLCommandPortKeyRingPW** das verschlüsselte Kennwort für den Zugriff auf die Verschlüsselungshardware und den CA-Schlüsselring an.

   Legen Sie als Wert der **SSL*KeyRingPW**-Eigenschaften die Ausgabe des verschlüsselten Kennworts durch den Befehl **mqiptPW** fest.

e) Wenn die Verschlüsselungshardware mehrere persönliche Zertifikate enthält, geben Sie an, welches Zertifikat von MQIPT ausgewählt werden soll, um zur Authentifizierung an den SSL/TLS-Server oder -Client gesendet zu werden.

   Sie können angeben, welches Zertifikat ausgewählt werden soll, indem Sie eine oder mehrere der **SSLClientSite***-Eigenschaften für eine SSL-/TLS-Clientroute oder eine oder mehrere der **SSLServerSite***-Eigenschaften für eine SSL-/TLS-Serverroute festlegen.

   Sie können angeben, welches Zertifikat vom TLS-Befehlsport verwendet werden soll, indem Sie mit der Eigenschaft **SSLCommandPortSiteLabel** den Namen der Zertifikatsbezeichnung angeben.

   Weitere Informationen zur Auswahl von Zertifikaten aus einem Schlüsselring finden Sie unter „Zertifikate aus einer Schlüsselringdatei in MQIPT auswählen” auf Seite 1127. Die Eigenschaften für die Auswahl eines Zertifikats aus einem Schlüsselring sind im Abschnitt MQIPT-Routeneigenschaften beschrieben.

Wenn Sie beispielsweise einen Schlüsselspeicher für die Verschlüsselungshardware für das Sitezertifikat in einer TLS-Server-Route verwenden und eine Schlüsselringdatei zum Speichern der CA-Zertifikate für die gleiche Route, fügen Sie der Routendefinition die folgenden Eigenschaften hinzu:

```
SSLServerKeyRingUseCryptoHardware=true
SSLServerKeyRingPW=<mqiptPW>1!gORdM4wft5d1rCgNMDEGag==!dZxhgQD2A8Ea0yeqawQvPg==
SSLServerCAKeyRing=/opt/mqipt/ssl/ca.pfx
SSLServerCAKeyRingPW=<mqiptPW>1!3Vdrpiu6kMwn0sWRCVgT5g==!LHltGLEg3OFvN8+02Re0YA==
SSLServerSiteLabel=mqiptsite
```

10. Starten Sie MQIPT erneut.

## Java security managerinMQIPT

Der Java security manager kann mit jeder MQIPT-Funktion verwendet werden, um ein höheres Sicherheitsniveau zu bieten.

**Anmerkung:** Deprecated Die Verwendung von Java security manager mit MQIPT wird nicht weiter unterstützt, da Java security manager in einem zukünftigen Release von Javanicht mehr verwendet wird.

MQIPT verwendet den standardmäßigen Java security manager, wie in der Klasse `java.lang.SecurityManager` definiert. Die Java security manager-Funktion in MQIPT kann mit der globalen Eigenschaft **SecurityManager** aktiviert oder inaktiviert werden. Weitere Informationen finden Sie im Abschnitt Globale Eigenschaften von MQIPT.

Der Java security manager verwendet zwei Standardrichtliniendateien:

- Eine globale Systemrichtliniendatei mit dem Namen *$MQIPT_PATH*/java/jre/lib/security/java.policy (wobei *$MQIPT_PATH* das Verzeichnis ist, in dem MQIPT installiert ist) wird von allen Instanzen einer virtuellen Maschine auf einem Host verwendet.
- Eine benutzerspezifische Richtliniendatei mit dem Namen .java.policy, die im Ausgangsverzeichnis des Benutzers vorhanden sein kann.

Es kann auch eine zusätzliche MQIPT-Richtliniendatei verwendet werden. Die MQIPT-Richtliniendatei sollte anstelle der zuvor beschriebenen Standardrichtliniendateien verwendet werden. Weitere Informationen finden Sie unter **SecurityManagerPolicy** im Abschnitt Globale Eigenschaften vonMQIPT .

Die Syntax der Richtliniendatei ist recht komplex. Sie kann zwar in einem Texteditor geändert werden, in der Regel ist es jedoch einfacher, Änderungen über das mit Java bereitgestellte Policy-Tool vorzunehmen. Das Dienstprogramm "Richtlinientool" befindet sich im Verzeichnis *$MQIPT_PATH*/java/jre/bin und ist in der Java-Dokumentation vollständig dokumentiert.

Eine Beispielrichtliniendatei (mqiptSample.policy) wurde mit MQIPT bereitgestellt, um Ihnen anzuzeigen, welche Berechtigungen für die Ausführung von MQIPT festgelegt werden müssen.

Sie müssen die Beispielrichtliniendatei Ihrer Konfiguration entsprechend bearbeiten. Insbesondere kann es sein, dass das Ausgangsverzeichnis von MQIPT mit der Konfigurationsdatei mqipt.conf möglicherweise nicht mit dem MQIPT-Installationsverzeichnis identisch ist. Achten Sie daher darauf, dass Sie die korrekten Verzeichnisse angeben, wenn Sie **FilePermission**-Einträge in der Sicherheitsrichtlinie konfigurieren.

Folgende Einträge müssen geändert werden:

- Der Eintrag **java.io.FilePermission** , der Lese -und Schreibzugriff auf das Verzeichnis errors erteilt. Der Dateipfad in diesem Eintrag muss auf das MQIPT -Ausgangsverzeichnis verweisen, da sich hier das Verzeichnis errors befindet. MQIPT erstellt FFST -Fehlerdatenerfassungsdateien (AMQ*.FDC) und Tracedateien (AMQ*.TRC*) im Verzeichnis errors . Sie müssen sicherstellen, dass MQIPT über die Berechtigung zum Erstellen von Trace- und FFST-Dateien im Verzeichnis errors verfügt, so dass eine Fehlerbehebung möglich ist.
- Der Eintrag **java.io.FilePermission** , der Lese -und Schreibzugriff auf das Verzeichnis logs erteilt. Der Dateipfad in diesem Eintrag muss auf das MQIPT -Ausgangsverzeichnis verweisen, da sich hier das Verzeichnis logs befindet. MQIPT erstellt Verbindungsprotokolldateien (mqipt*.log) im Verzeichnis logs , wenn die globale Eigenschaft **ConnectionLog** aktiviert ist.
- Die **java.io.FilePermission**-Eintrage, die Lese- und Ausführungszugriff (execute) auf alle Verzeichnisse im MQIPT-Installationsverzeichnis erteilen, also beispielsweise auf die Verzeichnisse bin, exits, lib und ssl. Die Dateipfade in diesen Einträgen müssen so geändert werden, dass sie auf das MQIPT-Installationsverzeichnis verweisen. Einige dieser Einträge können weggelassen werden, wenn sie nicht benötigt werden.
- Die **java.net.SocketPermission**-Einträge müssen geändert werden, um die eingehenden Verbindungen in die einzelnen empfangsbereiten MQIPT-Routen zu steuern. Es werden die Berechtigungen für die Empfangsbereitschaft und zum Akzeptieren für den Listener-Port und die Listener-Adresse für jede MQIPT-Route benötigt.
- Die **java.net.SocketPermission**-Einträge müssen geändert werden, um die ausgehenden Verbindungen von den einzelnen MQIPT-Routen zu steuern. Die Berechtigung connect ist für alle Routenziele, Proxy-Server oder LDAP-Server erforderlich, zu denen die MQIPT -Route eine Verbindung herstellt. Bei Angabe von Adressen unter Verwendung eines Hostnamens ist die Auflöseberechtigung resolve anstelle einer IP-Adresse erforderlich.

Je nach Konfiguration müssen Sie möglicherweise auch die folgenden Einträge hinzufügen:

- Einen **java.io.FilePermission**-Eintrag, um Lesezugriff auf die Konfigurationsdatei mqipt.conf oder das MQIPT-Ausgangsverzeichnis mit der Datei mqipt.conf zu erteilen.
- Einen **java.io.FilePermission**-Eintrag, um Lesezugriff auf die Sicherheitsrichtliniendatei selbst zu erteilen. Dies ist hilfreich, wenn die Sicherheitsrichtliniendatei aufgrund einer MQIPT-Aktualisierung erneut gelesen wird.

- Einige **java.io.FilePermission**-Einträge, um Lesezugriff auf alle SSL/TLS-Schlüsselringdateien und Schlüsselringkennwortdateien zu erteilen. Dies ist nur bei Verwendung einer Route erforderlich, für welche die Eigenschaften **SSLClient** oder **SSLServer** aktiviert sind, oder wenn der TLS-Befehlsport konfiguriert ist.
- Einige **java.io.FilePermission**-Einträge, um Lese- oder Ausführungszugriff auf alle MQIPT-Exitklassen zu erteilen. Dies ist nur erforderlich, wenn ein MQIPT-Exit aktiviert ist. Möglicherweise müssen Sie zusätzliche Berechtigungen erteilen, wenn der Exit dies erfordert.

**Anmerkung:** Windows **java.io.FilePermission** -Einträge müssen für jeden Backslash im Pfad zwei Backslash-Zeichen (\\) verwenden. Dies liegt daran, dass ein einzelner Backslash als Escapezeichen verwendet wird.

In der Beispieldatei wird davon ausgegangen, dass MQIPT auf einem Windows-System in C:\Program Files\IBM\MQ Internet Pass-Thru installiert wurde. Außerdem wird davon ausgegangen, dass das Ausgangsverzeichnis von MQIPT (die Position der mqipt.conf-Datei) mit dem MQIPT-Installationsverzeichnis identisch ist.

Wenn Sie MQIPT in einer anderen Position installiert haben, müssen Sie das Verzeichnis in der Definition **codeBase** ändern, sodass es auf Ihr MQIPT-Installationsverzeichnis verweist. Achten Sie darauf, das richtige Präfix (file:/) und das richtige Dateisuffix (/lib/com.ibm.mq.ipt.jar) einzuschließen. Auf AIX and Linux -Systemen kann eine typische **codeBase** URL file:/opt/mqipt/lib/com.ibm.mq.ipt.jarsein, vorausgesetzt, MQIPT ist in /opt/mqiptinstalliert.

Berechtigungen werden in der Regel mit drei Attributen definiert. Für die Steuerung von Socketverbindungen lauten ihre Werte wie folgt:

**class permission (Klassenberechtigung)**
java.net.SocketPermission

**name to control (Zu steuernder Name)**
Das Format hierfür lautet hostname:port, jede Komponente des Namens kann dabei durch ein Platzhalterzeichen angegeben werden. Hostname kann ein Domänenname oder eine IP-Adresse sein. Die Position ganz links im Hostnamen kann durch einen Stern (*) angegeben werden. harry.company1.com würde beispielsweise mit jeder dieser Zeichenfolgen übereinstimmen:

- harry
- harry.company1.com
- *.company1.com
- *
- 198.51.100.123 (sofern dies die IP-Adresse von harry.company1.com) ist.

Die Portkomponente des Namens kann als einzelne Portadresse oder als Bereich von Portadressen angegeben werden, z. B.:

**1414**
nur Port 1414

**1414-**
alle Portadressen größer-gleich 1414

**-1414**
alle Portadressen kleiner-gleich 1414

**1-1414**
alle Portadressen von 1 bis einschließlich 1414

**allowed action (Zulässige Aktion)**
Von java.net.SocketPermission werden folgende Aktionen verwendet:

**accept**
Verbindungen können vom angegebenen Ziel akzeptiert werden

**Verbinden**
Verbindungen zum angegebenen Ziel sind zulässig

**listen (überwachen)**
Die Anwendung ist an dem angegebenen Port oder den Ports für Verbindungsanforderungen empfangsbereit

**Auflösen**
DNS kann zum Auflösen von Domänennamen für IP-Adressen verwendet werden

Die Steuerung der Java security manager kann auch über die Systemeigenschaften `java.security.manager` und `java.security.policy` Java erfolgen. Es wird jedoch empfohlen, die Eigenschaften **SecurityManager** und **SecurityManagerPolicy** zur Steuerung von MQIPTzu verwenden.

Um Diagnoseinformationen in Trace- und FFST-Datensätze einzuschließen, muss MQIPT auf bestimmte MQIPT-Systemeigenschaften und Umgebungsvariablen zugreifen. Sie müssen immer die folgenden Eigenschaften in die Java-Sicherheitsrichtlinie einbeziehen:

```
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.runtime.version", "read";
permission java.util.PropertyPermission "java.vm.info", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.lang.RuntimePermission "getenv.MQIPT_PATH";
permission java.lang.RuntimePermission "getStackTrace";
permission javax.management.MBeanServerPermission "createMBeanServer";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#-[com.ibm.mq.ipt:ty
pe=IPTManager]", "registerMBean";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#-[com.ibm.mq.ipt:ty
pe=IPTManager]", "unregisterMBean";
permission javax.management.MBeanTrustPermission "register";
```

Wenn Sie nicht alle diese Eigenschaften einbeziehen, funktioniert MQIPT nicht ordnungsgemäß und die Problemdiagnose ist beeinträchtigt.

## Sicherheitsexits in MQIPT

Verwenden Sie einen Sicherheitsexit, um den Zugriff auf ein Ziel wie in der Routeneigenschaft **Destination** definiert zu steuern. Der Sicherheitsexit wird an dem Punkt aufgerufen, an dem MQIPT eine Verbindungsanforderung von einem Client empfängt, aber bevor er die Verbindung zum Ziel herstellt.

Basierend auf den einleitenden Verbindungseigenschaften entscheidet der Sicherheitsexit, ob der Verbindungsaufbau abgeschlossen werden kann.

Beim Starten einer Route wird der Sicherheitsexit aufgerufen, um die Verbindung zu initialisieren und bereit zu sein, eine Verbindungsanforderung zu verarbeiten. Der Initialisierungsprozess sollte dazu genutzt werden, eventuell vorhandene Benutzerdaten zu laden und diese Daten für den schnellen und problemlosen Zugriff vorzubereiten, so dass der Zeitaufwand beim Verarbeiten einer Verbindungsanforderung minimiert wird.

Jede Route kann einen eigenen Sicherheitsexit haben.

- Die Eigenschaft **SecurityExit** dient dazu, den benutzerdefinierten Sicherheitsexit zu aktivieren bzw. inaktivieren.
- Die Eigenschaft **SecurityExitName** dient dazu, den Klassennamen des benutzerdefinierten Sicherheitsexits zu definieren.
- Die Eigenschaft **SecurityExitPath** dient dazu, den Namen des Verzeichnisses zu definieren, in dem die Klassendatei enthalten ist. Wenn diese Eigenschaft nicht gesetzt ist, wird davon ausgegangen, dass sich die Klassendatei im Unterverzeichnis 'exits' befindet. Mit **SecurityExitPath** kann auch der Name einer JAR-Datei definiert werden, die den benutzerdefinierten Sicherheitsexit enthält.
- Die Eigenschaft **SecurityExitTimeout** wird von MQIPT verwendet, um festzustellen, wie lange es beim Validieren einer Verbindungsanforderung auf eine Antwort vom Sicherheitsexit warten soll.

Im Abschnitt Routeneigenschaften für MQIPT finden Sie Informationen zu den Eigenschaften des Sicherheitsexits.

MQIPT verwendet die Klasse `SecurityExit`, um einen benutzerdefinierten Sicherheitsexit aufzurufen. Diese Klasse muss um den benutzerdefinierten Sicherheitsexit erweitert und die meisten ihrer Methoden außer Kraft gesetzt werden, um die erforderliche Funktionalität bereitzustellen. Ein Objekt `SecurityExitResponse` wird verwendet, um Daten an MQIPT zurückzugeben, und diese Daten werden von MQIPT verwendet, um zu entscheiden, ob die Verbindungsanforderung akzeptiert oder zurückgewiesen werden soll. Das Objekt `SecurityExitResponse` kann auch eine neue Ziel- und Zielportadresse enthalten, die verwendet wird, um die Route zu überschreiben, die durch die Eigenschaften des Sicherheitsexits definiert ist.

Es werden drei Beispielsicherheitsexits bereitgestellt, um zu zeigen, wie ein Sicherheitsexit implementiert werden kann.

- `SampleSecurityExit` zeigt, wie der Zugriff auf einen IBM MQ-Warteschlangenmanager anhand des Namens des IBM MQ-Kanals gesteuert wird. Er ermöglicht nur eine Verbindung mit einem Kanalnamen, der mit der Zeichenfolge "MQIPT" beginnt. Weitere Informationen finden Sie unter Sicherheitsexit verwenden.
- `SampleRoutingExit` ermöglicht das dynamische Routing von Clientverbindungsanforderungen an einen Pool von definierten IBM MQ-Servern, wobei jeder Server einen Warteschlangenmanager mit demselben Namen und gleichen Attributen hostet. Das Beispiel beinhaltet eine Konfigurationsdatei, die eine Liste mit Servernamen enthält. Weitere Informationen finden Sie unter Clientverbindungsanforderungen an IBM MQ-Warteschlangenmanager-Server mithilfe von Sicherheitsexits weiterleiten.
- `SampleOneRouteExit` ermöglicht das dynamische Routing an einen IBM MQ-Warteschlangenmanager, der von dem in der Verbindungsanforderungverwendeten IBM MQ-Kanalnamen abgeleitet wird. Das Beispiel beinhaltet eine Konfigurationsdatei, die eine Zuordnung der Warteschlangenmanagernamen zu den Servernamen enthält. Weitere Informationen finden Sie unter Clientverbindungsanforderungen dynamisch weiterleiten.

**Anmerkung:** MQIPT wird in einer einzelnen JVM ausgeführt, so dass ein benutzerdefinierter Sicherheitsexit den normalen Betrieb von MQIPT auf eine der folgenden Arten gefährden kann:

- Auswirkungen auf Systemressourcen
- Generieren von Engpässen
- Vermindern der Leistung

Sie sollten die Auswirkungen Ihres Sicherheitsexits ausgiebig testen, bevor Sie sie in einer Produktionsumgebung implementieren.

## Klasse com.ibm.mq.ipt.exit.SecurityExit in MQIPT

Diese Klasse und ihre öffentlichen Methoden müssen um den benutzerdefinierten Sicherheitsexit erweitert werden, um Zugriff auf einige allgemeine Daten zu erhalten und einige MQIPT-Initialisierungsschritte zu ermöglichen.

Vor dem Aufruf der einzelnen Methoden durch MQIPT werden einige Eigenschaften für die zu verwendende Methode zur Verfügung gestellt. Ihre Werte können unter Verwendung der entsprechenden get-Methoden abgerufen werden, die in dieser Klasse definiert sind.

### Methoden

**public int init(IPTTrace)**

Folgende Eigenschaften stehen zur Verfügung:

- Listener-Port
- destination
- Zielport
- Version

Die Methode `init` wird von MQIPT aufgerufen, wenn eine Route gestartet wird. Bei der Rückgabe von dieser Methode muss der Sicherheitsexit bereit sein, um eine Verbindungsanforderung zu validieren. Gültige Rückkehrcodes sind `ExitRc.OK` oder `ExitRc.INIT_ERROR`.

**public int refresh(IPTTrace)**
Folgende Eigenschaften stehen zur Verfügung:

- Listener-Port
- destination
- Zielport

Die Methode `refresh` wird von MQIPT aufgerufen, wenn die MQIPT-Konfiguration aktualisiert wird. Diese Aktion wird in der Regel ausgeführt, wenn eine Eigenschaft in der Konfigurationsdatei geändert wurde. MQIPT lädt erneut alle Eigenschaften aus der Konfigurationsdatei, um festzustellen, welche Eigenschaften geändert wurden und ob eine Route erneut gestartet werden muss.

Diese Methode sollte alle externen Daten neu laden, die von ihr verwendet werden; d. h. Daten, die von der Methode **`init`** geladen werden. Gültige Rückkehrcodes sind `ExitRc.OK` oder `ExitRc.REFRESH_ERROR`.

**public void close(IPTTrace)**
Folgende Eigenschaften stehen zur Verfügung:

- Listener-Port
- destination
- Zielport

Die Methode `close` wird von MQIPT aufgerufen, wenn sie gestoppt wird. Mit dieser Methode sollten alle Systemressourcen freigegeben werden, die der Exit während des Betriebs angefordert hat. MQIPT wartet mit dem Herunterfahren, bis diese Methode abgeschlossen ist.

Diese Methode wird auch aufgerufen, wenn zuvor ein Sicherheitsexit aktiviert wurde, dieser aber jetzt in der Konfigurationsdatei inaktiviert wurde.

**public SecurityExitResponse validate(IPTTrace)**
Folgende Eigenschaften stehen zur Verfügung:

- Listener-Port
- destination
- Zielport
- Zeitlimit
- Client-IP-Adresse
- Client-Port-Adresse
- Kanalname
- Name des Warteschlangenmanagers

Die Methode `validate` wird von MQIPT aufgerufen, wenn eine Verbindungsanforderung zur Validierung empfangen wird. Der Kanalname und der Name des Warteschlangenmanagers sind nicht verfügbar, wenn die Eigenschaft **`SSLProxyMode`** aktiviert wurde, da diese Funktion nur für die Tunnelung von TLS-Daten verwendet wird und daher die Daten, die in der Regel aus dem Anfangsdatenfluss abgerufen werden, nicht lesbar sind.

Der Sicherheitsexit muss ein `SecurityExitResponse`-Objekt mit den folgenden Informationen zurückgeben:

- Ursachencode (muss gesetzt sein)
- Neue Zieladresse (optional)
- Adresse des neuen Ziel-Listener-Ports (optional)
- Nachricht (optional)

Der Ursachencode gibt an, ob die Verbindung von MQIPT akzeptiert oder abgelehnt wird. Die Felder `newDestination` und `newDestinationPort` können optional so festgelegt werden, dass ein neuer Zielwarteschlangenmanager definiert wird. Wenn Sie diese Eigenschaften nicht festlegen, werden die in

der Konfigurationsdatei definierten Routeneigenschaften **Destination** und **DestinationPort** verwendet. Alle Nachrichten werden an den Verbindungsprotokolldateieintrag angehängt.

Die folgenden Methoden werden für den Abruf der Werte von MQIPT-Konfigurationseigenschaften unterstützt:

**public int getListenerPort()**
Ruft den von der Eigenschaft **ListenerPort** definierten Route-Listener-Port ab

**public String getDestination()**
Ruft die von der Eigenschaft **Destination** definierte Zieladresse ab

**public int getDestinationPort()**
Ruft die von der Eigenschaft **DestinationPort** definierte Ziellistener-Portadresse ab

**public String getClientIPAddress()**
Ruft die IP-Adresse des Clients ab, von dem die Verbindungsanforderung gestellt wurde

**public int getClientPortAddress()**
Ruft die Portadresse ab, die vom Client, der die Verbindungsanforderung stellt, verwendet wird

**public int getTimeout()**
Ruft den Zeitlimitwert ab. MQIPT wartet darauf, dass der in der Eigenschaft **SecurityExitTimeout** definierte Sicherheitsexit eine Anforderung validiert

**public int getConnThreadID()**
Ruft die Verbindungsthread-ID ab, die die Verbindungsanforderung verarbeitet (nützlich für Debugzwecke).

**public String getChannelName()**
Ruft den IBM MQ-Kanalnamen ab, der in der Verbindungsanforderung verwendet wird

**public String getQMName()**
Ruft den Namen des IBM MQ-Warteschlangenmanagers ab, der in der Verbindungsanforderung verwendet wird

**public boolean getTimedout()**
Kann vom Sicherheitsexit verwendet werden, um festzustellen, ob das Zeitlimit abgelaufen ist.

## Die Klasse com.ibm.mq.ipt.exit.SecurityExitResponse in MQIPT

Diese Klasse wird verwendet, um von einem benutzerdefinierten Sicherheitsexit eine Antwort an MQIPT zurückzugeben, und dient dazu, zu ermitteln, ob die Verbindungsanforderung akzeptiert oder zurückgewiesen werden soll.

Objekte dieses Typs werden nur in der Validierungsmethode erstellt (siehe „Klasse com.ibm.mq.ipt.exit.SecurityExit in MQIPT" auf Seite 1152). Für die Erstellung dieser Objekte sind Komfortkonstruktoren vorhanden, und für jede Eigenschaft gibt es Methoden. Weitere Informationen finden Sie in den Beispiel-Sicherheitsexits.

Durch das Erstellen eines standardmäßigen SecurityExitResponse-Objekts wird die Verbindungsanforderung zurückgewiesen.

### Konstruktoren

- **public SecurityExitResponse (String dest, int destPort, int rc, String msg)**

  Dabei gilt:

  - `dest` ist das neue Ziel
  - `destPort` ist die neue Zielportadresse
  - `rc` ist der Ursachencode
  - `msg` ist eine Nachricht, die dem Verbindungsprotokolleintrag hinzugefügt wird
- **public SecurityExitResponse (String dest, int destPort, int rc)**
- **public SecurityExitResponse (int rc, String msg)**

- **public SecurityExitResponse (int rc)**

### Methoden

**public void setDestination(String dest)**
Setzt eine neue Zieladresse für die Verbindungsanforderung

**public void setDestinationPort(int port) throws IPTException**
Setzt eine neue Ziellistener-Portadresse für die Verbindungsanforderung - für eine ungültige Portadresse eine IPTException auslösen

**public void setMessage(String msg)**
Fügt dem Verbindungsprotokolldatensatz eine Nachricht hinzu

**public void setReasonCode(int rc)**
Legt den Ursachencode für die Verbindungsanforderung fest.

## Rückgabecodes für Sicherheitsexits in MQIPT

Die Rückgabecodes, die MQIPT erkennt, wenn ein Sicherheitsexit in verschiedenen Situationen aufgerufen wird.

Nachfolgende Rückgabecodes werden in folgenden Situationen von MQIPT beim Aufruf eines Sicherheitsexits erkannt:

| Rückgabecode | Beschreibung | init | Validieren | Aktualisierung |
|---|---|---|---|---|
| ExitRc.OK | Anforderung erfolgreich abgeschlossen. | ja | ja | ja |
| ExitRc.INIT_ERROR | Init-Anforderung fehlgeschlagen, Route wird inaktiviert. | ja | | |
| ExitRc.REFRESH_ERROR | Aktualisierungsanforderung fehlgeschlagen. | | | ja |
| ExitRc.NOT_AUTHORIZED | Validierungsprozess fehlgeschlagen, Verbindungsanforderung abgelehnt. | | ja | |
| ExitRc.DISABLE_SSL | Validierungsanforderung erfolgreich, bei Verbindung zum Ziel wird SSL oder TLS nicht verwendet. | | ja | |

# Steuerung der Portnummer in MQIPT

Bei Verwendung von MQIPT ist es möglich, den Bereich der lokalen Portnummern zu beschränken, die bei der Erstellung einer abgehenden Verbindung verwendet wird.

Legen Sie die Eigenschaft **OutgoingPort** für die Route fest, um die ursprüngliche lokale Portnummer anzugeben, und legen Sie **MaxConnectionThreads** fest, um die Anzahl der zu verwendenden Ports anzugeben. Wenn Sie beispielsweise **OutgoingPort** auf 1600 setzen und **MaxConnectionThreads** auf 20, liegt der Bereich der lokalen Portnummern für diese Route bei 1600 - 1619.

Es liegt in der Verantwortung des MQIPT-Administrators, sicherzustellen, dass es keine Konflikte bei den Portnummern zwischen den Routen gibt.

Wenn **OutgoingPort** nicht definiert ist, bedeutet ein Standardwert von 0, dass für jede Verbindung eine vom System zugeordnete Portnummer verwendet wird.

Bei Verwendung von HTTP liegt die Anzahl der abgehenden Ports doppelt so hoch wie ohne Verwendung von HTTP. Im vorherigen Beispiel wäre der Nummernbereich 1600-1639, wenn die Route HTTP verwendet.

Weitere Informationen finden Sie unter Portnummern zuordnen .

### Mehreren Netzen zugehörige Systeme

Wenn Sie ein mehreren Netzen zugehöriges System verwenden, können Sie mit der Eigenschaft **LocalAddress** angeben, an welche IP-Adresse eine abgehende Verbindung gebunden wird. Die Angabe von Hostnamen wird von dieser Eigenschaft nicht unterstützt.

## Gespeicherte Kennwörter in MQIPT verschlüsseln

Die MQIPT-Konfiguration kann Kennwörter für den Zugriff auf verschiedene Ressourcen sowie das Kennwort für den Zugriff auf MQIPT über den Befehlsport einschließen. Ab IBM MQ 9.2.0 sollten alle diese Kennwörter durch eine Verschlüsselung geschützt werden.

### Informationen zu diesem Vorgang

In Versionen vor IBM MQ 9.2.0 können nur Kennwörter verschlüsselt werden, die von MQIPT für den Zugriff auf Schlüsselringe oder Verschlüsselungshardwareschlüsselspeicher verwendet werden. Die verschlüsselten Kennwörter werden in Dateien gespeichert, die von einer der **SSL*KeyRingPW**-Eigenschaften referenziert werden. Andere Kennwörter für LDAP-Server und das MQIPT-Zugriffskennwort werden als unverschlüsselter Text in der Konfigurationsdatei `mqipt.conf` gespeichert.

Ab IBM MQ 9.2.0 sollten alle für die Verwendung durch MQIPT gespeicherten Kennwörter geschützt werden, indem sie mit dem Befehl **mqiptPW** verschlüsselt werden. Die verschlüsselten Kennwörter werden als Eigenschaftswerte in der Konfigurationsdatei `mqipt.conf` gespeichert. MQIPT ist in der Lage, zwischen verschlüsselten Kennwörtern, Klartextkennwörtern und Dateinamen in Eigenschaftswerten zu unterscheiden. Sie sollten alle Kennwörter, die für die Verwendung durch MQIPT gespeichert werden, auf diese Weise verschlüsseln, da es sich um die sicherste Zugriffsschutzmethode handelt.

Deprecated Die Methode zum Verschlüsseln von Schlüsselspeicherkennwörtern, die in MQIPT vor IBM MQ 9.2.0 verwendet wurden, ist veraltet, kann aber für Konfigurationseigenschaften verwendet werden, die vor IBM MQ 9.2.0verfügbar waren. Um den Schutz von Schlüsselringkennwörtern zu verbessern, sollten alle Schlüsselringkennwörter, die zuvor verschlüsselt wurden, mit der neuesten Zugriffsschutzmethode erneut verschlüsselt werden.

**Anmerkung:** Die Eigenschaft **SSLCommandPortKeyRingPW** in der Konfigurationsdatei `mqipt.conf` und die Eigenschaft **SSLClientCAKeyRingPW** in der Eigenschaftendatei **mqiptAdmin** können sich nicht auf Kennwortdateien beziehen. Als Werte für diese Eigenschaften muss die Ausgabe der Zeichenfolge für das verschlüsselte Kennwort durch den Befehl **mqiptPW** festgelegt werden.

Wenn in der MQIPT-Konfiguration ein unverschlüsseltes oder schwach geschütztes Kennwort vorhanden ist, wird entweder beim Start von MQIPT oder beim Start einer Route eine Warnung ausgegeben.

Verwenden Sie dieses Verfahren, um ein Kennwort, das für die Verwendung durch MQIPT gespeichert werden soll, mithilfe der neuesten Zugriffsschutzmethode zu verschlüsseln. Zum Verschlüsseln eines Schlüsselringkennworts in MQIPT vor IBM MQ 9.2.0führen Sie die Schritte in „Schlüsselringkennwort vor MQIPT in IBM MQ 9.2.0 verschlüsseln” auf Seite 1157aus.

### Vorgehensweise

1. Optional: Erstellen Sie eine Datei, die den Kennwortverschlüsselungsschlüssel enthält, sofern noch keine vorhanden ist.

   MQIPT verwendet einen Verschlüsselungsschlüssel zum Verschlüsseln von Kennwörtern. Sie können in einer Datei einen eigenen Verschlüsselungsschlüssel angeben. Die Datei muss mindestens ein Zeichen und darf nur eine Textzeile enthalten.

   Alle gespeicherten Kennwörter für eine Instanz von MQIPT werden mit dem gleichen Kennwortverschlüsselungsschlüssel verschlüsselt und entschlüsselt. Daher benötigen Sie für jede MQIPT-Installation nur eine einzelne Schlüsseldatei für die Kennwortverschlüsselung.

Sie können einen anderen Kennwortverschlüsselungsschlüssel verwenden, um die in der Eigenschaftendatei **mqiptAdmin** gespeicherten Kennwörter zu verschlüsseln, als den Verschlüsselungsschlüssel, mit dem Kennwörter in der MQIPT-Konfiguration verschlüsselt werden.

Wenn Sie MQIPT als Service ausführen möchten, der automatisch gestartet wird, müssen Sie die Kennwortverschlüsselungsschlüsseldatei mit dem Standardnamen `mqipt_cred.key` erstellen und sie in das Ausgangsverzeichnis von MQIPT stellen.

Sie müssen keinen Kennwortverschlüsselungsschlüssel angeben, aus Sicherheitsgründen wird dies allerdings empfohlen. Wenn Sie keinen eigenen Verschlüsselungsschlüssel angeben, wird der Standardverschlüsselungsschlüssel verwendet.

**Anmerkung:** Sie müssen sicherstellen, dass in der Schlüsseldatei für die Kennwortverschlüsselung die entsprechenden Dateiberechtigungen festgelegt sind, damit nur berechtige Benutzer den Verschlüsselungsschlüssel lesen können. Nur der Benutzer, der den Befehl **mqiptPW** ausführt, und der Benutzer, unter dem MQIPT ausgeführt wird, benötigen die Berechtigung zum Lesen des Kennwortverschlüsselungsschlüssels.

2. Verschlüsseln Sie das Kennwort mit dem Befehl **mqiptPW**.

   Die Syntax des Befehls **mqiptPW** wird im Abschnitt mqiptPW (gespeichertes Kennwort verschlüsseln) beschrieben.

   Wenn Sie in Schritt „1" auf Seite 1156eine Kennwortverschlüsselungsschlüsseldatei erstellt haben, geben Sie den Dateinamen mit dem Parameter **-sf** für **mqiptPW**an. Der folgende Befehl kann beispielsweise ausgegeben werden, um ein Kennwort mit dem Verschlüsselungsschlüssel in der Datei zu verschlüsseln, die im Parameter **-sf** angegeben ist:

   ```
   mqiptPW -sf /opt/mqipt/mqipt_password.key
   ```

3. Geben Sie bei einer entsprechenden Aufforderung das zu verschlüsselnde Kennwort ein.

   Das verschlüsselte Kennwort wird von **mqiptPW** ausgegeben.

4. Kopieren Sie das verschlüsseltes Kennwort in die entsprechende Eigenschaft in der Konfigurationsdatei `mqipt.conf` oder die Eigenschaftendatei **mqiptAdmin**.

   In der folgenden Zeile wird beispielsweise ein verschlüsseltes Kennwort für das MQIPT-Zugriffskennwort angegeben:

   ```
   AccessPW=<mqiptPW>1!QL+2Jvj/tigKKlD7Nz8Oqw==!AMDBefOUrmPf5i1OuqV5MA==
   ```

5. Starten Sie MQIPT. Wenn Sie in Schritt „1" auf Seite 1156 eine Kennwortverschlüsselungsschlüsseldatei mit einem anderen als den Standardnamen erstellt haben, geben Sie den Namen der Verschlüsselungsschlüsseldatei beim Start von MQIPT an.

   Sie können den Namen der Kennwortverschlüsselungsschlüsseldatei mit dem Parameter **-sf** angeben, wenn Sie MQIPT starten. Geben Sie beispielsweise den folgenden Befehl aus, um MQIPT mit dem Verschlüsselungsschlüssel in der Datei zu starten, die mit dem Parameter **-sf** angegeben wird:

   ```
   mqipt /opt/mqipt -sf /opt/mqipt/mqipt_password.key
   ```

   Informationen zu anderen Methoden zur Angabe des Kennwortverschlüsselungsschlüsseldateinamens beim Starten von MQIPT finden Sie im Abschnitt Kennwortverschlüsselungsschlüssel angeben.

   Sie können den Namen der Kennwortverschlüsselungsschlüsseldatei für den Befehl **mqiptAdmin** mit der Eigenschaft **PasswordProtectionKeyFile** in der Eigenschaftendatei **mqiptAdmin** angeben.

## Schlüsselringkennwort vor MQIPT in IBM MQ 9.2.0 verschlüsseln

Vor IBM MQ 9.2.0werden verschlüsselte Kennwörter, die für den Zugriff auf die von MQIPT verwendeten Schlüsselringe verwendet werden, in Dateien gespeichert.

### Informationen zu diesem Vorgang

Befolgen Sie die Prozedur in dieser Task, um ein Schlüsselringkennwort für die Verwendung durch MQIPT vor IBM MQ 9.2.0zu verschlüsseln. Verwenden Sie ab MQIPT in IBM MQ 9.2.0 for Long Term Supportdie in

„Gespeicherte Kennwörter in MQIPT verschlüsseln” auf Seite 1156beschriebene sicherere Schutzmethode.

### Vorgehensweise

1. Verschlüsseln Sie das Schlüsselringkennwort mit dem Befehl **mqiptPW**.
   Geben Sie folgenden Befehl ein, um das Kennwort zu verschlüsseln:

   ```
   mqiptPW password filename
   ```

   Dabei gilt Folgendes:

   ***password***
   das Klartextkennwort ist, das für den Zugriff auf den Schlüsselring benötigt wird.

   ***Dateiname***
   der Name der Kennwortdatei ist, die erstellt werden soll

   Die Syntax des Befehls **mqiptPW** wird im Abschnitt mqiptPW (gespeichertes Kennwort verschlüsseln) beschrieben.
2. Setzen Sie die entsprechende Routeneigenschaft auf den Namen der Datei, die das verschlüsselte Kennwort enthält, das in Schritt „1” auf Seite 1158 erstellt wurde.
   Wenn Sie beispielsweise die Kennwortdatei für den Schlüsselring angeben möchten, der das TLS-Serverzertifikat von MQIPT enthält, fügen Sie die folgende Zeile zur `mqipt.conf`-Konfigurationsdatei hinzu:

   ```
   SSLServerKeyRingPW=filename
   ```

## Weitere Sicherheitsaspekte für MQIPT

MQIPT verfügt über mehrere zusätzliche Funktionen, die einen Entwickler beim Erstellen einer sicheren Lösung unterstützen.

- Enthält ein internes Netz sehr viele Clients, die ausgehende Verbindungen anfordern, können diese alle über einen MQIPT innerhalb der Firewall geführt werden. Der Firewall-Administrator muss dann nur der MQIPT-Maschine eine externe Zugriffsberechtigung erteilen.
- MQIPT kann nur zu solchen Warteschlangenmanagern eine Verbindung herstellen, die explizit in der zugehörigen Konfigurationsdatei angegeben sind, außer MQIPT wird als SOCKS-Proxy eingesetzt oder verwenden einen Sicherheitsexit.
- MQIPT überprüft, ob die von ihm empfangenen und gesendeten Nachrichten gültig sind und dem IBM MQ-Protokoll entsprechen. Dadurch wird verhindert, dass MQIPT für Hackerangriffe außerhalb des IBM MQ-Protokolls benutzt wird. Wenn MQIPT als SSL/TLS-Proxy agiert, kann MQIPT, wenn alle IBM MQ-Daten und -Protokolle verschlüsselt wurden, nur den ersten SSL/TLS-Handshake garantieren. In diesem Fall sollten Sie den Java security manager verwenden.
- MQIPT lässt zu, dass Kanalexits ihre eigenen durchgehenden Sicherheitsprotokolle ausführen.
- Sie können die die maximale Anzahl ankommender Verbindungen über die Eigenschaft `MaxConnectionThreads` festlegen. Dadurch können anfällige interne Warteschlangenmanagerdaten vor Denial-of-Service-Attacken geschützt werden

### Konfigurationsdatei

Sie müssen verhindern, dass die MQIPT-Konfigurationsdatei `mqipt.conf` von nicht berechtigten Benutzern gelesen wird, da sie sensible Informationen wie beispielsweise das **AccessPW**-Kennwort enthalten kann, mit dem der ferne Verwaltungszugriff auf MQIPT gesteuert wird. Schützen Sie alle Kennwörter, die in der Konfigurationsdatei angegeben sind, indem Sie der Prozedur in „Gespeicherte Kennwörter in MQIPT verschlüsseln” auf Seite 1156 folgen. Stellen Sie außerdem sicher, dass `mqipt.conf` vor einer unbefugten Änderung geschützt ist. Legen Sie die Berechtigungen der Betriebssystemdatei für `mqipt.conf` so

fest, dass die Datei nur aus dem Benutzerkonto gelesen und aktualisiert werden kann, mit dem MQIPT ausgeführt wird.

### Befehlsport

Die MQIPT -Befehlsports akzeptieren Verwaltungsbefehle, die über das Netz an eine ferne Instanz von MQIPT mit dem Befehl **mqiptAdmin** ausgegeben wurden.

Ab IBM MQ 9.2.0 kann MQIPT mit einem nicht gesicherten Befehlsport und einem mit TLS gesicherten Befehlsport konfiguriert werden. Verbindungen zum nicht gesicherten Befehlsport sind nicht verschlüsselt.

**Anmerkung:** Daten, die über das Netz an den nicht gesicherten Befehlsport gesendet werden (einschließlich dem MQIPT-Zugriffskennwort), können für andere Benutzer im Netz sichtbar sein.

Sie müssen entscheiden, ob Sie einen Befehlsport aktivieren müssen, und die Risiken der Fernverwaltung von MQIPT abschätzen, bevor Sie den nicht gesicherten Befehlsport oder den TLS-Befehlsport aktivieren. Ab IBM MQ 9.2.0 können mit dem Befehl **mqiptAdmin** lokale Instanzen von MQIPT verwaltet werden, die mit dem gleichen Benutzer wie der Befehl **mqiptAdmin** ausgeführt werden, ohne dass ein Befehlsport erforderlich ist. Deshalb müssen Sie möglicherweise keinen Befehlsport aktivieren, um lokale Instanzen von MQIPT zu verwalten.

Wenn der nicht gesicherte Befehlsport oder der TLS-Befehlsport aktiviert ist, müssen Sie einen unbefugten Zugriff auf den Befehlsport verhindern. Beim Sichern des Zugriffs auf den Befehlsport sollten Sie die folgenden Punkte berücksichtigen:

- Beschränken Sie die Gruppe der Computer, die eine Verbindung zum Befehlsport von MQIPT herstellen können, mithilfe einer Firewall.
- Aktivieren Sie die Authentifizierung auf den Befehlsports mithilfe der Eigenschaften **AccessPW** und **RemoteCommandAuthentication**. Weitere Informationen zur Aktivierung der Authentifizierung von Befehlsports finden Sie im Abschnitt Authentifizierung von Befehlsports.
- Inaktivieren Sie eventuell den fernen Systemabschluss mit der Eigenschaft **RemoteShutdown**.
- Sie können auch die Eigenschaften **CommandPortListenerAddress** und **SSLCommandPortListenerAddress** verwenden, um die Befehlsports so zu konfigurieren, dass sie in einer bestimmten Netzschnittstelle empfangsbereit sind.

Weitere Informationen zur Verwendung des Befehls **mqiptAdmin** zur Verwaltung von MQIPTfinden Sie unter MQIPT über die Befehlszeile verwalten.

## Verbindungsprotokolle in MQIPT

MQIPT stellt eine Funktion für das Verbindungsprotokoll bereit, das Listen aller erfolgreichen und nicht erfolgreichen Verbindungsversuche enthält.

In das Verbindungsprotokoll wird ein Eintrag für jede Verbindung geschrieben, die von einer MQIPT-Route empfangen oder hergestellt wird, und für jeden Verwaltungsbefehl, der von MQIPT empfangen wird. Das Verbindungsprotokoll wird mithilfe der Eigenschaften **ConnectionLog** und **MaxLogFileSize** gesteuert. Weitere Informationen finden Sie im Abschnitt Globale Eigenschaften von MQIPT.

Bei jedem Start von MQIPT wird ein neues Verbindungsprotokoll erstellt. Zur Kennzeichnung enthält der Dateiname die aktuelle Zeitmarke, z. B.:

```
mqiptYYYYMMDDHHmmSS.log
```

Dabei gilt Folgendes:

YYYY für das Jahr
MM für den Monat
DD für den Tag
HH für die Stunden
mm für die Minuten

SS für die Sekunden

Wenn ein Verbindungsprotokoll die maximale Größe erreicht, die durch die Eigenschaft **MaxLogFile-Size** festgelegt wird, wird eine Sicherungsdatei (`mqipt001.log`) erstellt. Es werden maximal zwei Sicherungsdateien verwaltet (`mqipt001.log` und `mqipt002.log`).

Ein Eintrag im Verbindungsprotokoll stellt den jeweiligen Teil einer Verbindungsanforderung dar. Eine Verbindungsanforderung, die von MQIPT empfangen wird, und die daraus resultierende neue Verbindung, die MQIPT zur Zieladresse herstellt, werden als zwei Protokolleinträge und nach Beendigung der jeweiligen Verbindung als zwei weitere Einträge angezeigt.

Hier ist das Verbindungsprotokoll für eine erfolgreiche Verbindungsanforderung:

```
Wed May 15 13:13:51 BST 2013 conn accept 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:51 BST 2013 conn conn   127.0.0.1(3843) localhost(3500) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close  127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close  127.0.0.1(3843) localhost(3500) OK 5000-0
```

Hier ist ein Verbindungsprotokoll für eine fehlgeschlagene Verbindungsanforderung:

```
Wed May 15 14:56:40 BST 2013 conn accept 127.0.0.1(4138) 127.0.0.1(7000) OK    7000-0
Wed May 15 14:56:40 BST 2013 conn close  127.0.0.1(4138) 127.0.0.1(7000) ERROR 7000-0
 Unrecognized SSL handshake request '54'
```

## Einträge im Verbindungsprotokoll

Jeder Eintrag im Verbindungsprotokoll enthält die folgenden Informationen:

- Zeitpunkt, an den der Eintrag erstellt wurde.
- Der Typ des Eintrags. Dieser kann einen der folgenden Werte annehmen:

  **admin**
  Verwaltungsbefehl

  **conn**
  Routenverbindung
- Das aufgetretene Ereignis. Dieser kann einen der folgenden Werte annehmen:

  **accept**
  Die Verbindungsanforderung wurde empfangen

  **schließen**
  Die Verbindung wurde geschlossen

  **conn**
  Verbindungsanforderung an das Routenziel

  **dspipt**
  Empfang des MQIPT-Befehls wird angezeigt

  **nodata**
  Es wurden keine Daten vom Anrufer empfangen

  **ping**
  Eine Pinganforderung wurde empfangen

  **Status**
  Zeigt den Status des empangenen Befehls an

  **refr**
  Ein Befehl zur Aktualisierung wurde empfangen

  **Stopp**
  Ein Befehl zum Stoppen wurde empfangen
- Quellennetzadresse und Portnummer. Der Wert LOCAL wird für Verwaltungsbefehle angezeigt, die lokal ohne Verwendung des Befehlsports ausgegeben wurden.

- Zielnetzadresse und Portnummer. Dies wird nicht für Verwaltungsbefehle angezeigt, die lokal ohne Verwendung des Befehlsports ausgegeben wurden.
- Beendigungscode. Der Wert kann 0K oder ERR0R sein.
- Die MQIPT-Thread-ID.
- Eine optionale Fehlernachricht.

## IBM MQ Internet Pass-Thru mithilfe von Containern konfigurieren

Sie können IBM MQ Internet Pass-Thru (MQIPT) in einem Container ausführen. Das vom Container verwendete Basisimage muss ein unterstütztes Linux-Betriebssystem verwenden.

### Prozedur

- Ein Beispiel-Image für den MQIPT-Docker ist im GitHub-Repository 'mq-container' verfügbar. Zum Erstellen und Ausführen des Containers folgen Sie den Anweisungen in IBM MQ Internet Pass-Thru auf Docker.

### Nächste Schritte

Sie können aktive Container mit dem Befehl **`docker ps`** anzeigen. Verwenden Sie den Befehl **`docker logs ${CONTAINER_ID}`** , um die Konsolenausgabe von MQIPT anzuzeigen, die in einem Docker -Container ausgeführt wird.

## V 9.3.0 Streaming-Warteschlangen konfigurieren

Das Feature "Streaming-Warteschlangen" ermöglicht es Ihnen, eine Duplikatkopie jeder Nachricht, die an eine Warteschlange eingereiht wird, an eine zweite Warteschlange zu kopieren. Die Konfiguration von Streaming-Warteschlangen wird in einer Warteschlange nach Warteschlangenbasis ausgeführt.

Lokale und Modellwarteschlangen haben zwei neue Attribute, die sich auf Streaming-Warteschlangen beziehen:

**STREAMQ**
Dies ist der Name der Warteschlange, an die gestreamte Nachrichten zugestellt werden sollen. Sie sollten das Attribut **STREAMQ** auf den Namen einer anderen Warteschlange setzen.

Es gibt Einschränkungen, bei denen Warteschlangen konfiguriert werden können, um Nachrichten in andere Warteschlangen zu streamen, und es gibt Einschränkungen, für die Warteschlangen als Ziel für gestreamte Nachrichten festgelegt werden können. Informationen zu Nachrichtenstreaming-Einschränkungen finden Sie unter Einschränkungen für Datenstromwarteschlange.

**STRMQOS**
Dies ist die Servicequalität, die bei der Zustellung von gestreamten Nachrichten verwendet wird.

Sie können das Attribut **STRMQOS** auf einen von zwei Werten setzen:

**BESTEF**
Der beste Aufwand, bei dem es sich um den Standardwert handelt.

Der Warteschlangenmanager versucht, eine Kopie jeder Nachricht an die Warteschlange zu übergeben, die im Attribut **STREAMQ** angegeben ist. Wenn es ein Problem bei der Zustellung der gestreamten Nachricht gibt, hat dies keine Auswirkungen auf die Zustellung der ursprünglichen Nachricht.

**MUSTDUP**
Der Warteschlangenmanager versucht, eine Kopie jeder Nachricht an die Streaming-Warteschlange zu übergeben.

Wenn ein Problem bei der Zustellung der gestreamten Nachricht auftritt, wird die ursprüngliche Nachricht nicht an ihre Warteschlange zugestellt und die Anwendung empfängt MQCC_FAILED mit einem entsprechenden Ursachencode.

Weitere Informationen finden Sie in den MQSC-Befehlen ALTER queues, DEFINE queues und DISPLAY QUEUEUE sowie in den PCF-Befehlen Change, Copy und Create Queue, Inquire Queue und Inquire Queue (Response).

Wenn mehr als eine Kopie jeder Nachricht erforderlich ist, können Sie das Attribut **STREAMQ** so konfigurieren, dass es auf den Namen einer IBM MQ-Aliaswarteschlange verweist, deren Ziel sich auf ein IBM MQ-Topic bezieht. Wenn eine Nachricht an die ursprüngliche Warteschlange eingereiht wird, wird eine Kopie der Nachricht in dem genannten Topic veröffentlicht.

Sie müssen sicherstellen, dass Sie über API- oder verwaltete Subskriptionen für das Topic-Objekt verfügen, da jede Subskription eine Kopie der Nachricht empfängt. Die Nachricht, die den Subskribenten zugestellt wird, folgt den gleichen Regeln wie andere Publish/Subscribe-Nachrichten. Beispiel: Jede Nachricht hat eine neue Nachrichten-ID und die Kontextfelder des MQMD unterscheiden sich von denen in der ursprünglichen Nachricht. Weitere Informationen zu den Ähnlichkeiten und Unterschieden zwischen Original- und gestreamten Nachrichten finden Sie unter Gestreamte Nachrichten.

## Beispiele

### Beispiel für den besten Aufwand

Im folgenden Beispiel ist dies eine lokale Warteschlange ORDERS.QUEUE wird geändert, um gestreamte Nachrichten in eine zweite Warteschlange ANALYTICS.QUEUE. Die BESTEF-Servicequalität wird verwendet, um sicherzustellen, dass beim Einreihen der gestreamten Nachricht in ANALYTICS.QUEUE, z. B. wenn ANALYTICS.QUEUE ist voll, die ursprüngliche Nachricht kann weiterhin in die ORDERS.QUEUE.

Diese Art der Konfiguration kann verwendet werden, um Analysen für die empfangenen Reihenfolgen durchzuführen, indem die gestreamten Nachrichten analysiert werden, während die ursprünglichen Nachrichten in die Auftragswarteschlange eingereiht und verarbeitet werden. Ein Vorteil des Features der Streaming-Warteschlange besteht darin, dass Sie die gestreamten Nachrichten in der ANALYTICS.QUEUE auf ihre Verarbeitung warten lassen können, ohne dass dies Auswirkungen auf die tatsächlichen Aufträge hat, die vom Unternehmen erfüllt werden.

```
DEFINE QLOCAL(ANALYTICS.QUEUE)

ALTER QLOCAL(ORDERS.QUEUE) STRMQOS(BESTEF) STREAMQ(ANALYTICS.QUEUE)
```

**Anmerkung:** Im Beispiel wurde **STRMQOS** auf BESTEFgesetzt, obwohl Sie dieses Attribut aus dem Befehl **ALTER** auslassen können, da BESTEF die Standardservicequalität ist.

### Beispiel für Duplikat

In diesem Beispiel wird eine lokale Warteschlange PAYMENTS.QUEUE so geändert, dass gestreamte Kopien jeder Nachricht in eine andere lokale Warteschlange AUDIT.QUEUE. eingereiht werden. Es ist wichtig, dass jede Nachricht, die in die Zahlungswarteschlange eingereiht wird, in die Prüfungswarteschlange weitergeleitet wird, sodass die Servicequalität MUSTDUP verwendet wird.

Wenn es ein Problem bei der Zustellung der gestreamten Nachricht an die Warteschlange gibt, wird auch die ursprüngliche Nachricht nicht zugestellt, und die Anwendung erhält einen entsprechenden Abschluss- und Ursachencode. Die Anwendung muss die Einreihung in derselben Weise wiederholen, wie sie es tun würde, wenn nur eine einzige Warteschlange betroffen wäre.

```
DEFINE QLOCAL(AUDIT.QUEUE)

ALTER QLOCAL(PAYMENTS.QUEUE) STRMQOS(MUSTDUP) STREAMQ(AUDIT.QUEUE)
```

**Anmerkungen:**

1. Es ist nicht erforderlich, dass die Streaming-Warteschlange beim Ändern der ursprünglichen Warteschlange vorhanden ist. Es ist jedoch zu beachten, dass aufgrund der verwendeten Servicequalität MUSTDUP Versuche, Nachrichten in die ursprüngliche Warteschlange einzureihen, fehlschlagen, bis Sie die Streaming-Warteschlange definiert haben.

2. Bei der Verwendung eines Aliasnamens für eine Warteschlange mit dem Ziel eines Topic-Objekts wird die Zustellung der gestreamten Nachricht auch dann als erfolgreich angesehen, wenn es keine Subskribenten gibt, und die ursprüngliche Nachricht wird an ihre Warteschlange zugestellt.
3. Wenn eine gestreamte Nachricht nicht an die eigene Warteschlange zugestellt werden kann, versucht der Warteschlangenmanager nicht, sie an die Warteschlange für nicht zustellbare Nachrichten zuzustellen. Wenn jedoch eine gestreamte Nachricht an eine entfernte Warteschlange gesendet wird, kann sie, wenn sie über einen Kanal an einen anderen Warteschlangenmanager weitergeleitet wird, gemäß den bestehenden Regeln für nicht zustellbare Nachrichten an eine Warteschlange für nicht zustellbare Nachrichten geliefert werden.

## Streaming-Warteschlange konfigurieren

In der Streaming-Warteschlange muss keine zusätzliche Konfiguration ausgeführt werden. Sie empfängt Nachrichten aus allen Warteschlangen, die sie als Streaming-Warteschlange benennen. Es kann jedoch sinnvoll sein, die in der Streaming-Warteschlange konfigurierten Attributwerte zu berücksichtigen.

Wenn die ursprüngliche Warteschlange beispielsweise eine maximale Tiefe von 100.000 hat und die Streaming-Warteschlange nur eine maximale Tiefe von 5000 hat, gehen möglicherweise gestreamte Nachrichten verloren, wenn STRMQOS auf BESTEF gesetzt ist oder wenn STRMQOS bei der Festlegung auf MUSTDUP mit dem Fehler MQRC_Q_FULL fehlschlägt, obwohl die ursprüngliche Warteschlange viel Platz auf dem Server hat.

Berücksichtigen Sie, welche Attribute in der Streaming-Warteschlange möglicherweise geändert werden müssen, um auf der Basis der Konfiguration der ursprünglichen Warteschlange über entsprechende Werte zu verfügen.

**Zugehörige Konzepte**
Streaming-Warteschlangen

# Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in diesem Dokument beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf Produkte, Programme oder Services von IBM bedeuten nicht, dass nur Produkte, Programme oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East & Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellensprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos ohne Zahlung an IBM in jeder Form kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben sind. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

## Informationen zu Programmierschnittstellen

Die bereitgestellten Informationen zur Programmierschnittstelle sollen Sie bei der Erstellung von Anwendungssoftware für dieses Programm unterstützen.

Dieses Handbuch enthält Informationen über vorgesehene Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zu schreiben, um die Services von WebSphere MQzu erhalten.

Diese Informationen können jedoch auch Angaben über Diagnose, Bearbeitung und Optimierung enthalten. Die Informationen zu Diagnose, Bearbeitung und Optimierung sollten Ihnen bei der Fehlerbehebung für die Anwendungssoftware helfen.

**Wichtig:** Verwenden Sie diese Diagnose-, Änderungs-und Optimierungsinformationen nicht als Programmierschnittstelle, da sie Änderungen unterliegen.

# Marken

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (https://www.eclipse.org/) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Teilenummer:

(1P) P/N: