

9.0

*IBM MQ konfigurieren*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 849 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 0 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

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

© **Copyright International Business Machines Corporation 2007, 2023.**

---

# Inhaltsverzeichnis

<b>Konfiguration.....</b>	<b>5</b>
Warteschlangenmanager auf Multiplatforms erstellen und verwalten.....	5
Standardwarteschlangenmanager erstellen.....	9
Vorhandenen WS-Manager als Standardwert erstellen.....	10
Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern.....	11
WS-Manager starten.....	12
Stoppen eines Warteschlangenmanagers.....	13
WS-Manager erneut starten.....	15
Löschen eines Warteschlangenmanagers.....	16
Verbindungen zwischen dem Server und dem Client konfigurieren.....	17
Zu verwendende Übertragungsart.....	18
Einen erweiterten transaktionsorientierten Client konfigurieren.....	20
Definieren von MQI-Kanälen.....	32
AMQP-Kanäle erstellen und verwenden.....	33
Serververbindungs- und Clientverbindungsdefinitionen auf verschiedenen Plattformen erstellen..	38
Serververbindungs- und Clientverbindungsdefinitionen auf dem Server erstellen.....	42
Kanalexitprogramme für MQI-Kanäle.....	51
Client mit einer Gruppe mit gemeinsamer Warteschlange verbinden.....	55
Client mit einer Konfigurationsdatei konfigurieren.....	56
IBM MQ-Umgebungsvariablen verwenden.....	81
Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern.....	93
Konfigurationsdaten unter UNIX, Linux, and Windows ändern.....	93
Konfigurationsdaten unter IBM i ändern.....	102
Attribute für die Änderung von IBM MQ-Konfigurationsinformationen.....	115
Konfigurationsdaten des Warteschlangenmanagers ändern.....	122
Verteilte Warteschlangensteuerung konfigurieren.....	151
Verteilte Warteschlangenverfahren in IBM MQ.....	152
Einführung in die verteilte Warteschlangenverwaltung.....	174
Kanäle unter UNIX, Linux, and Windows überwachen und steuern.....	208
Kanäle in IBM i überwachen und steuern.....	233
WS-Manager-Cluster konfigurieren.....	255
Publish/Subscribe-Messaging konfigurieren.....	382
Publish/Subscribe-Nachrichtenattribute in der Warteschlange festlegen.....	382
In Warteschlange eingereichtes Publish/Subscribe starten.....	384
In Warteschlange eingereichtes Publish/Subscribe stoppen.....	384
Datenstrom hinzufügen.....	385
Datenstrom löschen.....	386
Einen Subskriptionspunkt hinzufügen.....	386
Verteilte Publish/Subscribe-Netze konfigurieren.....	388
Mehrere Installationen konfigurieren.....	408
Anwendungen in einer Umgebung mit mehreren Installationen verbinden.....	409
Primäre Installation ändern.....	419
WS-Manager einer Installation zuordnen.....	421
Installationen von IBM MQ auf einem System finden.....	422
Hochverfügbarkeit, Wiederherstellung und Neustart konfigurieren.....	424
Automatische Clientverbindungswiederholung.....	425
Überwachung der Konsolnachricht.....	431
Hochverfügbarkeitskonfigurationen.....	435
Protokollierung: Stellen Sie sicher, dass die Nachrichten nicht verloren gehen.....	561
IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen.....	590
Änderungen an der Cluster-Fehlerbehebung (auf anderen Servern als z/OS).....	599
JMS-Ressourcen konfigurieren.....	601

Verbindungsfactorys und Ziele in einem JNDI-Namensbereich konfigurieren.....	602
JMS-Objekte mit IBM MQ Explorer konfigurieren.....	605
JMS-Objekte mit dem Verwaltungstool konfigurieren.....	606
JMS-Ressourcen in WebSphere Application Server konfigurieren.....	616
Anwendungsserver für die Verwendung der neuesten Wartungsstufe des Ressourcenadapters konfigurieren.....	628
Eigenschaft JMS <b>PROVIDERVERSION</b> konfigurieren.....	631
Permanente WebSphere Application Server-Subskriptionen entfernen.....	640
IBM MQ Console und REST API konfigurieren.....	643
Sicherheit konfigurieren.....	643
CSRF-Schutz konfigurieren.....	644
Konfigurieren des HTTP-Host-Namens.....	645
HTTP-und HTTPS-Ports konfigurieren.....	646
Konfigurieren des Antwortzeitlimits.....	647
Autostart konfigurieren.....	649
Protokollierung konfigurieren.....	650
Ablaufintervall für LTPA-Token.....	653
administrative REST API-Gateway konfigurieren.....	654
messaging REST API konfigurieren.....	655
REST API für MFT konfigurieren.....	656
Die JVM des mqweb-Servers optimieren.....	658
Dateistruktur der Installationskomponente IBM MQ Console und REST API.....	658
IBM MQ in Docker konfigurieren.....	660
Docker-Unterstützung auf Linux-Systemen.....	661
Eigenes Image des IBM MQ-Warteschlangenmanagers mit Docker planen.....	661
Beispiel für ein IBM MQ-Warteschlangenmanagerimage mit Docker erstellen.....	662
Lokale Bindungsanwendungen in separaten Containern ausführen.....	666
IBM MQ für die Verwendung mit Push-Themen und Plattformereignissen für Salesforce konfigurieren.....	669
IBM MQ Bridge to Salesforce konfigurieren.....	670
Ereignisnachrichten für Salesforce-Plattformereignisse erstellen.....	675
IBM MQ Bridge to Salesforce ausführen.....	682
IBM MQ für die Verwendung mit Blockchain konfigurieren.....	683
Konfigurationsdatei für den IBM MQ Bridge to blockchain erstellen.....	686
IBM MQ Bridge to blockchain ausführen.....	691
IBM MQ Bridge to blockchain-Clientbeispiel ausführen.....	695
Warteschlangenmanager unter z/OS erstellen.....	697
Vorbereiten der Anpassung von Warteschlangenmanagern unter z/OS.....	697
IBM MQ for z/OS konfigurieren.....	702
Warteschlangenmanager auf z/OS testen.....	764
Kommunikation mit anderen Warteschlangenmanagern konfigurieren.....	773
IBM MQ mit IMS verwenden.....	805
IBM MQ mit CICS verwenden.....	814
Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen.....	814
OTMA-Exits in IMS verwenden.....	816
Verwendung von IBM z/OSMF zur Automatisierung von IBM MQ.....	821
IBM MQ Advanced for z/OS VUE konfigurieren.....	834
MFT-Agentenkonnektivität zu fernen z/OS-Warteschlangenmanagern.....	835
IBM MQ Advanced for z/OS VUE für die Verwendung mit Blockchain konfigurieren.....	835
<b>Bemerkungen.....</b>	<b>849</b>
Informationen zu Programmierschnittstellen.....	850
Marken.....	851

# 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.

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 und verwalten“](#) auf Seite 5.

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 151.

Anweisungen zum Verbinden Ihrer Clients mit einem IBM MQ-Server unter Verwendung unterschiedlicher Methoden finden Sie in [„Verbindungen zwischen dem Server und dem Client konfigurieren“](#) auf Seite 17.

Anweisungen zum Konfigurieren eines WS-Manager-Clusters finden Sie im Abschnitt [„WS-Manager-Cluster konfigurieren“](#) auf Seite 255.

Sie können das Verhalten von IBM MQ oder eines Warteschlangenmanagers ändern, indem Sie Konfigurationsinformationen ändern. Weitere Informationen finden Sie unter [„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“](#) auf Seite 93. Im Allgemeinen müssen Sie einen Warteschlangenmanager nicht erneut starten, damit die Konfigurationsänderungen wirksam werden, mit Ausnahme der in dieser Produktdokumentation genannten Änderungen.

 Anweisungen zur Konfiguration von IBM MQ for z/OS finden Sie im Abschnitt [„Warteschlangenmanager unter z/OS erstellen“](#) auf Seite 697.

## Zugehörige Tasks

[„Warteschlangenmanager unter z/OS erstellen“](#) auf Seite 697

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

## Zugehörige Informationen

[IBM MQ - Technische Übersicht](#)

[Lokale IBM MQ-Objekte verwalten](#)

[Ferne IBM MQ-Objekte verwalten](#)

 [IBM i verwalten](#)

 [IBM MQ for z/OS verwalten](#)

[Planung](#)

 [IBM MQ-Umgebung unter z/OS planen](#)

---

## **Warteschlangenmanager auf Multiplatforms erstellen und verwalten**

---

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 UNIX, 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 genannt wird, können Sie Ihren Warteschlangenmanager ACCOUNTS . SATURN . QUEUE . MANAGER benennen, wobei SATURN einen bestimmten Warteschlangenmanager angibt und QUEUE . MANAGER eine Erweiterung ist, die Sie allen Warteschlangenmanagern erteilen können. Alternativ können Sie dies weglassen. Beachten Sie jedoch, dass ACCOUNTS . SATURN und ACCOUNTS . SATURN . QUEUE . MANAGER unterschiedliche WS-Manager-Namen 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 WS-Manager 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 9.

### **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.

Wenn Sie einen Warteschlangenmanager erstellen, verwenden Sie 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 [Mit Warteschlangenmanagern arbeiten](#).

### **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.

Wenn Sie einen WS-Manager erstellen, verwenden Sie 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 424.

### UNIX

#### Nur für IBM MQ for UNIX-Systeme

Sie können das Warteschlangenmanagerverzeichnis `/var/mqm/qmgrs/qmgrauch` 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 9
  - [„Vorhandenen WS-Manager als Standardwert erstellen“](#) auf Seite 10
  - [„Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern“](#) auf Seite 11
  - [„WS-Manager starten“](#) auf Seite 12
  - [„Stoppen eines Warteschlangenmanagers“](#) auf Seite 13
  - [„WS-Manager erneut starten“](#) auf Seite 15
  - [„Löschen eines Warteschlangenmanagers“](#) auf Seite 16

## Zugehörige Tasks

[„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“](#) auf Seite 93

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

„Warteschlangenmanager unter z/OS erstellen“ auf Seite 697

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

### Zugehörige Informationen

[Erstellen eines Warteschlangenmanagers mit dem Namen QM1](#)

[System- und Standardobjekte](#)

[crtmqm](#)

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 und verwalten“ auf Seite 5 beschriebenen Hinweise.

**UNIX** Wenn Sie **crtmqm** verwenden, um einen Warteschlangenmanager unter UNIX 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 Erwägung 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 WS-Manager 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.MANAGER` erstellt:

```
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 Tasks

„Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern“ auf Seite 11

Die Konfigurationsinformationen für IBM MQ werden unter UNIX, 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.

### Zugehörige Informationen

Mit Warteschlangenmanagern arbeiten

Mit lokalen Warteschlangen arbeiten

System-und Standardobjekte

Multi

## Vorhandenen WS-Manager als Standardwert erstellen

Sie können einen vorhandenen Warteschlangenmanager als Standardwarteschlangenmanager definieren, entweder manuell mit Hilfe eines Texteditors oder (unter Windows und Linux) über den IBM MQ Explorer.

### 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 11.

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

„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

## **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 10.

### 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**.

## **Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern**

Die Konfigurationsinformationen für IBM MQ werden unter UNIX, 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:

- Bei der Installation des Produkts 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 ist eine *mqs.ini*-Datei vorhanden.

- Wenn Sie einen neuen Warteschlangenmanager erstellen, wird automatisch eine neue Warteschlangenmanagerkonfigurationsdatei (`qm.ini`) erstellt. Dieser Parameter enthält Konfigurationsparameter für den Warteschlangenmanager.

**V 9.0.0** Wenn Sie den AMQP-Service installiert haben, gibt es eine zusätzliche Konfigurationsdatei, die Sie sichern müssen:

- **Windows** Auf Windows-Systemen: `amqp_win.properties`
- **Linux** **UNIX** Auf UNIX- und Linux-Systemen: `amqp_unix.properties`

### Zugehörige Tasks

„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93  
 Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

„IBM MQ-Warteschlangenmanagerdaten sichern und wiederherstellen“ auf Seite 590

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.

## Multi **WS-Manager starten**

Wenn Sie einen WS-Manager erstellen, müssen Sie ihn starten, damit er Befehle oder MQI-Aufrufe verarbeiten kann.

### Informationen zu diesem Vorgang

Sie können einen WS-Manager mit dem Befehl `strmqm` starten. Eine Beschreibung des Befehls `strmqm` und seiner Optionen finden Sie unter [strmqm](#).

**Windows** **Linux** Alternativ können Sie auf Windows- und Linux-Systemen (x86- und x86-64-Plattformen) einen Warteschlangenmanager mit dem IBM MQ Explorer starten.

**Windows** Unter Windows können Sie einen Warteschlangenmanager über den IBM MQ Explorer automatisch starten, wenn das System gestartet wird. Weitere Informationen hierzu finden Sie im Abschnitt [Verwaltung mit dem IBM MQ Explorer](#).

### Prozedur

- Um einen Warteschlangenmanager mit dem Befehl `strmqm` zu starten, geben Sie den Befehl gefolgt vom Namen des Warteschlangenmanagers ein, den Sie starten wollen.

Geben Sie beispielsweise den folgenden Befehl ein, um einen WS-Manager mit dem Namen QMB zu starten:

```
strmqm QMB
```

**Anmerkung:** Sie müssen den Befehl `strmqm` aus der Installation verwenden, die dem Warteschlangenmanager zugeordnet ist, mit dem Sie arbeiten. Mit dem Befehl `dspmqr -o installation` können Sie herausfinden, welcher Installation ein Warteschlangenmanager zugeordnet ist.

Der Befehl `strmqm` gibt die Steuerung erst zurück, wenn der WS-Manager gestartet wurde und bereit ist, Verbindungsanforderungen zu akzeptieren.

- **Windows** **Linux**  
 Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager mit dem IBM MQ Explorer zu starten:
  - a) Öffnen Sie den IBM MQ Explorer.
  - b) Wählen Sie in der Navigatoransicht den WS-Manager aus.

c) Klicken Sie auf **Start** .

## Ergebnisse

Der WS-Manager wird gestartet.

Wenn der Start des Warteschlangenmanagers länger als ein paar Sekunden dauert, gibt IBM MQ periodisch Informationsnachrichten mit Einzelheiten zum Startfortschritt aus.

Multi

## Stoppen eines Warteschlangenmanagers

Sie können den Befehl **endmqm** verwenden, um einen Warteschlangenmanager zu stoppen. Dieser Befehl bietet drei Möglichkeiten zum Stoppen eines Warteschlangenmanagers: ein gesteuerter oder stiller Systemabschluss, ein sofortiges Herunterfahren und ein präventives Herunterfahren. Alternativ dazu können Sie unter Windows und Linux einen Warteschlangenmanager mit IBM MQ Explorer stoppen.

### Informationen zu diesem Vorgang

Es gibt drei Möglichkeiten, einen Einzel-Instanz-Warteschlangenmanager mit dem Befehl **endmqm** zu stoppen:

#### Gesteuerter Abschluss (Quiesce)

Standardmäßig führt der Befehl **endmqm** einen gesteuerten Abschluss des angegebenen Warteschlangenmanagers durch. Ein Quiesce-Shutdown wartet, bis alle verbundenen Anwendungen die Verbindung getrennt haben. Daher kann es eine Weile dauern, bis die Verbindung abgeschlossen ist.

#### Sofortige Beendigung

Für einen sofortigen Systemabschluss können alle aktuellen MQI-Aufrufe ausgeführt werden, aber alle neuen Aufrufe schlagen fehl. Dieser Typ des Herunterfahrens wartet nicht darauf, dass Anwendungen vom WS-Manager getrennt werden.

#### Präventiver Systemabschluss

Der WS-Manager wird sofort gestoppt. Use this type of shutdown only in exceptional circumstances, for example, when a queue manager does not stop as a result of a normal **endmqm** command.

Mit dem Befehl **endmqm** werden alle Instanzen eines Multi-Instanz-Warteschlangenmanagers auf dieselbe Weise wie ein Einzel-Instanz-Warteschlangenmanager beendet. Dabei können Sie **endmqm** auf der aktiven oder auf einer der Standby-Instanzen eines Multi-Instanz-Warteschlangenmanagers absetzen. Den Befehl **endmqm** zum Beenden des Warteschlangenmanagers müssen Sie allerdings auf der aktiven Instanz ausgeben.

Eine ausführliche Beschreibung des Befehls **endmqm** und seiner Optionen finden Sie im Abschnitt [endmqm](#).

**Tipp:** Probleme beim Herunterfahren eines Warteschlangenmanagers werden häufig durch Anwendungen verursacht. Zum Beispiel, wenn Anwendungen:

- MQI-Rückkehrcodes nicht korrekt überprüfen
- Keine Benachrichtigung über Quiesce anfordern
- Beenden ohne Verbindung zum WS-Manager (durch Absetzen eines MQDISC -Aufrufs)

Wenn bei dem Versuch, den Warteschlangenmanager zu stoppen, ein Problem auftritt, können Sie den Befehl **endmqm** mithilfe von Strg-C unterbrechen. Sie können dann einen weiteren **endmqm** -Befehl ausgeben, aber dieses Mal mit einem Parameter, der den Typ der erforderlichen Beendigung angibt.

Windows

Linux

Als Alternative zur Verwendung des Befehls **endmqm** können Sie unter Windows und Linux einen Warteschlangenmanager stoppen, indem Sie die IBM MQ Explorer verwenden, um entweder eine gesteuerte oder eine sofortige Beendigung durchzuführen.

## Prozedur

- Um den WS-Manager mit dem Befehl **endmqm** zu stoppen, geben Sie den Befehl gefolgt von dem Parameter, falls erforderlich, und den Namen des Warteschlangenmanagers ein, den Sie stoppen wollen.

**Anmerkung:** Sie müssen den Befehl **endmqm** aus der Installation verwenden, die dem Warteschlangenmanager zugeordnet ist, mit dem Sie arbeiten. Verwenden Sie den folgenden Befehl, um herauszufinden, welcher Installation ein Warteschlangenmanager zugeordnet ist: `dspmqr -o installation`.

- Um einen gesteuerten (stillgelegten) Systemabschluss durchzuführen, geben Sie den Befehl **endmqm** wie im folgenden Beispiel gezeigt ein, der einen Warteschlangenmanager mit dem Namen QMB stoppt:

```
endmqm QMB
```

Alternativ entspricht die Eingabe des Befehls **endmqm** mit dem Parameter **-c**, wie im folgenden Beispiel gezeigt, einem `endmqm QMB`-Befehl.

```
endmqm -c QMB
```

In beiden Fällen wird die Steuerung sofort an Sie zurückgegeben und Sie werden nicht benachrichtigt, wenn der WS-Manager gestoppt wurde. Wenn der Befehl warten soll, bis alle Anwendungen gestoppt sind und der WS-Manager beendet ist, bevor die Steuerung an Sie zurückgegeben wird, verwenden Sie stattdessen den Parameter **-w** wie im folgenden Beispiel gezeigt.

```
endmqm -w QMB
```

- Um einen sofortigen Systemabschluss durchzuführen, geben Sie den Befehl **endmqm** mit dem Parameter **-i** wie im folgenden Beispiel gezeigt ein:

```
endmqm -i QMB
```

- Wenn Sie einen präventiven Systemabschluss durchführen möchten, geben Sie den Befehl **endmqm** mit dem Parameter **-p** wie im folgenden Beispiel gezeigt ein:

```
endmqm -p QMB
```



**Achtung:** Ein präventiver Abschluss kann für verbundene Anwendungen unvorhersehbare Auswirkungen haben. Verwenden Sie diese Option nur, wenn alle anderen Versuche, den Warteschlangenmanager mit einem normalen **endmqm**-Befehl zu stoppen, fehlgeschlagen sind.  Wenn der vorbeugende Systemabschluss nicht funktioniert, versuchen Sie stattdessen, Warteschlangenmanager manuell zu stoppen.

- Zur Anforderung einer automatischen Clientverbindungswiederholung geben Sie den Befehl **endmqm** mit dem Parameter **-r** ein. Bei Angabe dieses Parameters werden die Verbindungen von Clients zu anderen Warteschlangenmanagern in ihrer Warteschlangenmanagergruppe wiederhergestellt.

**Anmerkung:** Beim Beenden eines Warteschlangenmanagers mit dem Standardbefehl **endmqm** wird die automatische Clientverbindungswiederholung nicht ausgelöst.

- Zur Übertragung einer Standby-Instanz eines Multi-Instanz-Warteschlangenmanagers nach dem Beenden der aktiven Instanz geben Sie den Befehl **endmqm** mit dem Parameter **-s** in der aktiven Instanz des Multi-Instanz-Warteschlangenmanagers ein.
- Um die Standby-Instanz eines Multi-Instanz-Warteschlangenmanagers zu beenden, während die aktive Instanz weiterhin ausgeführt wird, geben Sie den Befehl **endmqm** mit dem Parameter **-x** in der Standby-Instanz des Multi-Instanz-Warteschlangenmanagers ein.

- **Windows** **Linux**

Führen Sie unter Windows und Linux die folgenden Schritte aus, um den Warteschlangenmanager mit dem IBM MQ Explorer zu stoppen:

- a) Öffnen Sie den IBM MQ Explorer.
- b) Wählen Sie den Warteschlangenmanager in der Navigatoransicht aus.
- c) Klicken Sie auf **Stoppen** .  
Die Anzeige **End Queue Manager** wird angezeigt.
- d) Wählen Sie **Gesteuert** oder **Sofort** aus.
- e) Klicken Sie auf **OK** .  
Der Warteschlangenmanager wird gestoppt.

### Zugehörige Informationen

[Aktualisierungen der Wartungsstufe für Multi-Instanz-Warteschlangenmanager unter Windows anwenden](#)  
[Aktualisierungen der Wartungsstufe bei Multi-Instanz-Warteschlangenmanagern unter UNIX und Linux ausführen](#)

## Multi **WS-Manager erneut starten**

Sie können den Befehl **strmqm** verwenden, um einen Warteschlangenmanager erneut zu starten. Auf Windows- und Linux-x86-64-Systemen können Sie einen Warteschlangenmanager auch über den IBM MQ Explorer erneut starten.

### Informationen zu diesem Vorgang

Sie können einen WS-Manager mit dem Befehl **strmqm** erneut starten. Eine Beschreibung des Befehls **strmqm** und der zugehörigen Optionen finden Sie im Abschnitt [strmqm](#) .

- **Windows** **Linux**

Auf Windows- und Linux-x86-64-Systemen können Sie einen Warteschlangenmanager erneut starten, indem Sie den IBM MQ Explorer auf die gleiche Weise wie zum Starten eines Warteschlangenmanagers verwenden.

### Prozedur

- Um einen Warteschlangenmanager mit dem Befehl **strmqm** erneut zu starten, geben Sie den Befehl gefolgt vom Namen des WS-Managers ein, den Sie erneut starten wollen.  
Geben Sie beispielsweise den folgenden Befehl ein, um einen WS-Manager mit dem Namen `saturn.queue.manager` zu starten:

```
strmqm saturn.queue.manager
```

- **Windows** **Linux**

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager mit dem IBM MQ Explorer zu starten:

- a) Öffnen Sie den IBM MQ Explorer.
- b) Wählen Sie in der Navigatoransicht den WS-Manager aus.
- c) Klicken Sie auf **Start** .

### Ergebnisse

Der WS-Manager wird erneut gestartet.

Falls der Neustart des Warteschlangenmanagers länger als ein paar Sekunden dauert, gibt IBM MQ periodisch Informationsnachrichten mit Einzelheiten zum Startfortschritt aus.

## Löschen eines Warteschlangenmanagers

Sie können einen Warteschlangenmanager mit dem Befehl **dltmqm** löschen. Alternativ dazu können Sie auf Windows- und Linux-Systemen den IBM MQ Explorer verwenden, um einen Warteschlangenmanager zu löschen.

### Vorbereitende Schritte



#### Achtung:

- Das Löschen eines Warteschlangenmanagers ist ein drastischer Schritt, da Sie auch alle Ressourcen löschen, die dem Warteschlangenmanager zugeordnet sind, einschließlich aller Warteschlangen und deren Nachrichten und aller Objektdefinitionen. Wenn Sie den Befehl **dltmqm** verwenden, wird keine Eingabeaufforderung angezeigt, mit der Sie Ihre Meinung ändern können. Wenn Sie die Eingabetaste drücken, gehen alle zugeordneten Ressourcen verloren.
- **Windows** Unter Windows wird beim Löschen eines Warteschlangenmanagers auch der Warteschlangenmanager aus der automatischen Startliste gelöscht (siehe „WS-Manager starten“ auf Seite 12). Nach Abschluss des Befehls wird die Nachricht IBM MQ queue manager ending angezeigt. Sie werden nicht darüber informiert, dass der Warteschlangenmanager gelöscht wurde.
- Wenn Sie einen Cluster-WS-Manager löschen, wird er nicht aus dem Cluster entfernt. Weitere Informationen finden Sie in den Hinweisen zur Verwendung in [dltmqm](#).

### Informationen zu diesem Vorgang

Sie können einen WS-Manager mit dem Befehl **dltmqm** löschen. Eine Beschreibung des Befehls **dltmqm** und seiner Optionen finden Sie unter [dltmqm](#). Stellen Sie sicher, dass nur vertrauenswürdige Administratoren die Berechtigung haben, diesen Befehl zu verwenden. (Informationen zur Sicherheit finden Sie im Abschnitt [Sicherheit unter UNIX, Linux, and Windows einrichten](#).)

**Windows** **Linux** Alternativ können Sie auf Windows- und Linux-Systemen (x86- und x86-64-Plattformen) einen Warteschlangenmanager mit dem IBM MQ Explorer löschen.

### Prozedur

- Führen Sie die folgenden Schritte aus, wenn Sie einen WS-Manager mit dem Befehl **dltmqm** löschen möchten:
  - a) Stoppen Sie den Warteschlangenmanager.
  - b) Geben Sie den folgenden Befehl ein:

```
dltmqm QMB
```

**Anmerkung:** Sie müssen den Befehl **dltmqm** aus der Installation verwenden, die dem Warteschlangenmanager zugeordnet ist, mit dem Sie arbeiten. Um herauszufinden, welcher Installation ein Warteschlangenmanager zugeordnet ist, verwenden Sie den Befehl `dspmqr -o installation`.

- **Windows** **Linux**
  - a) Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager mit dem IBM MQ Explorer zu löschen:
    - a) Öffnen Sie den IBM MQ Explorer.
    - b) Wählen Sie in der Navigatoransicht den WS-Manager aus.
    - c) Wenn der WS-Manager nicht gestoppt wurde, stoppen Sie ihn.  
Um den Warteschlangenmanager zu stoppen, klicken Sie ihn mit der rechten Maustaste an, und klicken Sie anschließend auf **Stoppen**.
    - d) Löschen Sie den Warteschlangenmanager.

Um den Warteschlangenmanager zu löschen, klicken Sie ihn mit der rechten Maustaste an, und klicken Sie anschließend auf **Löschen**.

## Ergebnisse

Der WS-Manager wird gelöscht.

## Verbindungen zwischen dem Server und dem Client 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.

### Vorgehensweise

1. Entscheiden Sie sich für die Form der Kommunikation, die Sie verwenden werden.  
Weitere Informationen finden Sie in [„Zu verwendende Übertragungsart“](#) auf Seite 18.
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

[„Einen erweiterten transaktionsorientierten Client konfigurieren“](#) auf Seite 20

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

[„Definieren von MQI-Kanälen“](#) auf Seite 32

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*.

[„Kanalexitprogramme für MQI-Kanäle“](#) auf Seite 51

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

### Zugehörige Tasks

[„Serververbindungs- und Clientverbindungsdefinitionen auf verschiedenen Plattformen erstellen“](#) auf Seite 38

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.

[„Serververbindungs- und Clientverbindungsdefinitionen auf dem Server erstellen“](#) auf Seite 42

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

[„Client mit einer Gruppe mit gemeinsamer Warteschlange verbinden“](#) auf Seite 55

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.

„Client mit einer Konfigurationsdatei konfigurieren“ auf Seite 56

Sie konfigurieren Ihre Clients mit Hilfe von Attributen in einer Textdatei. Diese Attribute können durch Umgebungsvariablen oder auf andere plattformspezifische Methoden überschrieben werden.

### Zugehörige Informationen

[IBM MQ-MQI-Clientanwendungen mit Warteschlangenmanagern verbinden](#)

[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	 IBM i  UNIX  Windows	 IBM i  UNIX  Windows  z/OS
LU 6.2	 UNIX <sup>1</sup>  Windows	 IBM i  UNIX <sup>1</sup>  Windows  z/OS

Tabelle 1. Übertragungsprotokolle - Kombination von IBM MQ MQI client- und Serverplattformen (Forts.)

Übertragungsprotokoll	IBM MQ MQI client	IBM MQ-Server
NetBIOS	 Windows	 Windows
SPX	 Windows	 Windows

**Anmerkung:**

1. Mit Ausnahme von Linux (POWER-Plattform)

**Zugehörige Konzepte**

„TCP-Verbindung unter Windows definieren“ auf Seite 219

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 UNIX and Linux definieren“ auf Seite 227

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 247

Sie können eine TCP-Verbindung in der Kanaldefinition mit Hilfe des Felds Verbindungsname definieren.

„TCP-Verbindung unter z/OS definieren“ auf Seite 795

Um eine TCP-Verbindung zu definieren, gibt es eine Reihe von Einstellungen für die Konfiguration.

„LU 6.2-Verbindung unter Windows definieren“ auf Seite 221

SNA muss so konfiguriert werden, dass ein LU 6.2-Dialog zwischen den beiden Maschinen aufgebaut werden kann.

„LU 6.2-Verbindung unter UNIX and Linux definieren“ auf Seite 231

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 249

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 223

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 Verweise**

„TCP/IP-Verbindungslimits“ auf Seite 19

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 798

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	100
 HP-UX	20
 Linux	100
 IBM i	255
 Solaris	100
 Windows-Server	100
 Windows-Workstation	100
 z/OS	255

Wenn das Verbindungslimit erreicht ist, empfängt der Client vom Aufruf MQCONN den Rückkehrcode MQRC\_HOST\_NOT\_AVAILABLE und einen Fehler AMQ9202 im Clientfehlerprotokoll ( /var/mqm/errors/AMQERR0n.LOG auf UNIX and Linux-Systemen bzw. amqerr0n.log im Fehlerunterverzeichnis der Clientinstallation von IBM MQ 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.

## 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.

### 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

Frühere Versionen von IBM WebSphere MQ haben WebSphere Application Server 4 oder 5 unterstützt. Um den erweiterten transaktionsorientierten Client verwenden zu können, mussten bestimmte Konfigurationstasks ausgeführt werden. WebSphere Application Server 6 und höher enthält einen IBM WebSphere MQ -oder IBM MQ -Messaging-Provider, sodass Sie den erweiterten transaktionsorientierten Client nicht verwenden müssen.

## Zugehörige Konzepte

„XA-kompatible Transaktionsmanager konfigurieren“ auf Seite 21

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

„Microsoft Transaction Server“ auf Seite 31

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

## 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](#)
- ▶ **HP-UX** [IBM MQ-Client unter HP-UX installieren](#)
- ▶ **Linux** [IBM MQ-Client unter Linux installieren](#)
- ▶ **Solaris** [IBM MQ-Client unter Solaris 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

ten 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 28.

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 29
- [„Konfigurieren eines erweiterten transaktionsorientierten Clients für Tuxedo“](#) auf Seite 31

### **Zugehörige Konzepte**

[„Die Parameter CHANNEL, TRPTYPE, CONNAME und QMNAME der Zeichenfolge 'xa\\_open'“](#) auf Seite 26  
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 27

Der Aufruf 'xa\_open' schlägt unter bestimmten Umständen fehl.

### **Zugehörige Tasks**

[„Extended Transactional Client mit TLS-Kanälen verwenden“](#) auf Seite 29

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 27

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 27

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 Client-Verbindungen**

Einige XA-Transaktionsmanager verwenden Sequenzen von Transaktionskoordinierauffufen, 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 `5C6-00D40079` empfängt.

Für Transaktionsmanager, die auf diese Situation stoßen, führen Sie die folgenden Aktionen aus, um dem Transaktionsmanager die Interaktion mit IBM MQ for z/OS zu ermöglichen:

- Wenden Sie den Fix für APAR `PI73140` an.
- Aktivieren Sie die von `PI73140` bereitgestellte Änderung für den Serververbindungskanal, der vom Transaktionsmanager verwendet wird.

Sie können die Änderung aktivieren, indem Sie das Schlüsselwort `CSQSERVICE1` (in Großbuchstaben) an einer beliebigen Stelle im Beschreibungsfeld des `SVRCONN`-Kanals angeben.

Beachten Sie, dass Kanäle mit dem Schlüsselwort `CSQSERVICE1` die folgenden Einschränkungen aufweisen:

- 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.

### **Zugehörige Konzepte**

„XA-kompatible Transaktionsmanager konfigurieren“ auf Seite 21

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 steht `parm_name` für den Namen eines Parameters und `parm_value` für den 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. Beispiel:

```
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.

 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 Serververbindungen 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, trtype=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 an den Warteschlangenmanager des Servers 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.

Wenn die Anwendung eine Verbindung zu einem Warteschlangenmanager einer früheren Version als 7.0.1 herstellt, ist der Aufruf 'xa\_open' erfolgreich, aber die Transaktion verfügt über die Disposition QMGR der Arbeitseinheit mit Wiederherstellung.  Stellen Sie sicher, dass Anwendungen, die die Disposition GROUP der Arbeitseinheit mit Wiederherstellung erfordern, nur mit Warteschlangenmanagern unter 7.0.1 oder höher verbunden sind.

 Wenn die Anwendung einen QSG-Namen im Parameterfeld QMNAME verwendet und die Eigenschaft GROUPUR auf dem WS-Manager, mit dem sie verbunden ist, inaktiviert ist, schlägt der Aufruf von xa\_open fehl.

 Wenn der Anwendungsclient eine Verbindung zu einem z/OS -Warteschlangenmanager mit 7.0.1 oder höher herstellt, kann er für den Parameter QMNAME den Namen einer Gruppe mit gemeinsamer Warteschlange (QSG) 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 27.

<i>Tabelle 3. Angenommen, Werte des Parameters AXLIB</i>		
<b>Wert von TPM</b>	<b>Plattform</b>	<b>Assumierter Wert von AXLIB</b>
CICS	AIX	/usr/lpp/encina/lib/libEncServer.a (EncServer_shr.o)
CICS	HP-UX	/opt/encina/lib/libEncServer.sl
CICS	Solaris	/opt/encina/lib/libEncServer.so
CICS	Systeme mit Windows	libEncServer
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	HP-UX	/opt/tuxedo/lib/libtux.sl
Tuxedo	Solaris	/opt/tuxedo/lib/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:

#### **MQRMIASwitch**

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

#### **MQRMIASwitchDynamic**

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 28](#) dargestellten Bibliotheken.

<i>Tabelle 4. IBM MQ-Bibliotheken, die die XA-Switchstrukturen enthalten</i>	
<b>Plattform</b>	<b>Bibliothek mit den XA-Switchstrukturen</b>
AIX HP-UX Linux Solaris	<code>MQ_INSTALLATION_PATH/lib/libmqcxa</code>
Systeme mit Windows	<code>MQ_INSTALLATION_PATH\bin\mqcxa.dll</code> <sup>1</sup>

`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 28

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) **cic-sadd** hinzu. Die XAD-Ressourcendefinition gibt die folgenden Informationen an:

- Eine xa\_open-Zeichenfolge
- Der vollständig qualifizierte Pfadname einer Switchloaddatei.

Eine Switchloaddatei wird für die Verwendung durch CICS auf jeder der folgenden Plattformen bereitgestellt: AIX-, HP-UX-, Solaris- und Windows-Systeme. 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 30](#).

Tabelle 5. Die Switchloaddateien

Plattform	Switchloaddatei
AIX HP-UX Linux Solaris	<code>MQ_INSTALLATION_PATH/lib/amqczsc</code>
Systeme mit Windows	<code>MQ_INSTALLATION_PATH\bin\mqcc4swi.dll</code> <sup>1</sup>

`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, connname=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 31](#) 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 HP-UX Linux Solaris	amqzscgx.c	amqczscg
Systeme mit 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-UBBCONFIG-Datei 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. Unter AIX, HP-UX und Solaris lautet der Name der Datei ubbstxcx.cfg, und auf Windows-Systemen lautet der Name der Datei ubbstxcn.cfg.

- Im Eintrag für einen WS-Manager in der Tuxedo-Ressourcenmanagertabelle:
  - udataobj/RM ( AIX, HP-UX und Solaris)
  - udataobj\rm ( Windows-Systeme)

Geben Sie den Namen einer XA-Switchstruktur und den vollständig qualifizierten Pfadnamen der Bibliothek an, die die Struktur enthält. 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.

### **Microsoft Transaction Server**

Es ist keine zusätzliche Konfiguration erforderlich, bevor Sie 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 38.

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 42.

### 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 183

Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

## ULW AMQP-Kanäle erstellen und verwenden

Wenn Sie die IBM MQ -Unterstützung für MQ Light -APIs 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](#)
- ▶ **HP-UX** [IBM MQ-Komponenten für HP-UX-Systeme](#)
- ▶ **Linux** [IBM MQ-RPM-Komponenten für Linux-Systeme](#)
- ▶ **Linux** [IBM MQ-Debian-Komponenten für Linux-Ubuntu-Systeme](#)
- ▶ **Solaris** [IBM MQ-Komponenten für Solaris-Systeme](#)
- ▶ **Windows** [IBM MQ-Funktionen für Windows-Systeme](#)

Um eine Testverbindung zum Warteschlangenmanager herstellen zu können, benötigen Sie einen MQ Light-Client. MQ Light-Clients sind für Node.js, Ruby, Java und Python verfügbar. Weitere Informationen zu verfügbaren Clients finden Sie auf der Website der [IBM MQ Light Community](#).

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.

### Informationen zu diesem Vorgang

Bei dem folgenden Verfahren wird davon ausgegangen, dass über einen vorhandenen Warteschlangenmanager verfügen.

Wenn Sie einen neuen Warteschlangenmanager benötigen, wird ein Beispielscript einbezogen, das sich im Verzeichnis `mqinstall/amqp/samples` befindet. Das Script erstellt einen neuen Warteschlangenmanager, startet den AMQP-Service, erstellt einen neuen Kanal namens `SAMPLE.AMQP.CHANNEL` und startet den Kanal.

**Anmerkung:** AMQP-Kanäle unterstützen keine benutzerdefinierten AMQP-Services. AMQP-Kanäle unterstützen nur den Systemstandardservice `SYSTEM.AMQP.SERVICE`.

**Windows** ▶ **Linux** Wenn Sie das Beispielscript `SampleMQM.sh` unter Linux oder `SampleMQM.bat` unter Windows ausführen, können Sie die folgende Prozedur unter „6“ auf Seite 35 starten.

Sie können den Standardkanal `SYSTEM.DEF.AMQP` zum Testen von MQ Light-Verbindungen zum Warteschlangenmanager verwenden oder einen neuen Kanal erstellen.

Bei dem folgenden Verfahren wird der Standardkanal verwendet.

## Vorgehensweise

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

```
runmqsc QMNAME
```

2. **V 9.0.5**  
(Nur erforderlich, wenn Ihr Warteschlangenmanager aus IBM MQ 9.0.4 oder einer früheren Version stammt). 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 9.0.5 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 *MANUAL* setzen, können Sie dadurch verhindern, dass der Service beim Starten 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 MCAUSER-Benutzer-ID-Einstellung nur für Benutzer-IDs mit einer Länge von bis zu 12 Zeichen unterstützt.

- a) Verwenden Sie den Befehl **ALTER CHANNEL**, um die MCAUSER-Benutzer-ID 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 Fehlermeldung angezeigt.

4. Verwenden Sie den Befehl **START CHANNEL**, um den Standardkanal SYSTEM.DEF.AMQP zu starten:

```
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 in der Befehlsausgabe STATUS (RUNNING) 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. Beispiel:

```
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 wird das Zertifikat in dem Schlüsselrepository des Warteschlangenmanagers, dessen Bezeichnung dem Warteschlangenmanager-Attribut CERTLABL entspricht, von der SSL/TLS-Verschlüsselung als Name für den Kanal verwendet. Sie können ein anderes Zertifikat auswählen, indem Sie CERTLABL einstellen. 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. Ob ein Zertifikat von einer SSL/TLS-Clientverbindung erfordert wird, können Sie über SSLCAUTH festlegen. Verwenden Sie den Befehl **ALTER CHANNEL**, um festzulegen, ob von einer SSL/TLS-Clientverbindung ein Zertifikat benötigt wird. Beispiel:

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

- **V 9.0.0.10** Wenn Sie das Attribut SSLCAUTH auf REQUIRED setzen, kann der definierte Name (DN) des Zertifikats vom Client überprüft werden. Um den definierten Namen des Zertifikats vom

Client zu überprüfen, legen Sie das Attribut `SSLPEER` fest. Verwenden Sie den Befehl **ALTER CHANNEL**, um den definierten Namen des Zertifikats vom Client zu überprüfen. Beispiel:

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

Alternativ können Sie auch Kanalauthentifizierungsdatensätze verwenden, um Verbindungen zuzulassen oder zu blockieren, da diese Methode eine größere Granularität bietet als die Verwendung des Attributs `SSLPEER`. Weitere Informationen zum Festlegen von `SSLPEER` und zur Verwendung von Kanalauthentifizierungsdatensätzen als Alternative finden Sie im Abschnitt [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 Topic `public` subskribiert.

**Anmerkung:** Die `client-id` wird automatisch generiert, es sei denn, Sie geben über den Parameter `-i` eine ID an.

11. Navigieren Sie in einem neuen Befehlsfenster zum Verzeichnis `node_modules/mqlight/samples` und führen Sie die Beispielsenderanwendung mit dem folgenden Befehl aus:

```
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 befindet sich das Beispiel in folgenden Positionen:

- **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 Beispielpfängeranwendung 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 Informationen

[strmqm](#)

[AMQP-Clientanwendungen entwickeln](#)

[AMQP-Clients schützen](#)

**ULW**

## 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 MQ Light-APIs:

- **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 Informationen

[AMQP-Clientanwendungen entwickeln](#)

[AMQP-Clients schützen](#)

## **ULW** Protokolldateien für AMQP-Kanäle

Die Protokolldateien für AMQP-Kanäle werden in demselben IBM MQ-Datenverzeichnis wie 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 Dateien AMQP\*.FDC 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 Informationen

[Fehlerprotokolle unter UNIX, Linux, and Windows](#)

[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.

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

 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 39.
- Informationen zum Erstellen eines Clientverbindungskanals auf einem IBM MQ MQI client finden Sie unter [„Clientverbindungskanal auf dem IBM MQ MQI client erstellen“](#) auf Seite 40.

## 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 auf dem IBM MQ MQI client erstellen

Sie können einen Clientverbindungskanal auf der Client-Workstation mit MQSERVER oder mit der MQCNO-Struktur in einem MQCONN-Aufruf definieren.

### Verwenden von MQSERVER

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.

- Geben Sie wie folgt unter Windows eine einfache Kanaldefinition an:

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

- Geben Sie wie folgt auf UNIX and Linux-Systemen eine einfache Kanaldefinition an:

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

- Geben Sie wie folgt auf IBM i-Systemen eine einfache Kanaldefinition an:

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

Dabei gilt:

- ChannelName muss mit dem Namen identisch sein, der auf dem Server definiert ist. Er darf keinen Schrägstrich enthalten.
- TransportType kann je nach Ihrer IBM MQ MQI client-Plattform einen der folgenden Werte haben:
  - LU62
  - TCP
  - NETBIOS
  - SPX

**Anmerkung:** Auf UNIX and Linux-Systemen muss die Groß-/Kleinschreibung beachtet werden und der Wert für 'TransportType' muss in Großbuchstaben geschrieben werden. Ein MQCONN -oder MQCONNX -Aufruf gibt 2058 zurück, wenn der TransportType nicht erkannt wird.

- ConnectionName ist der Name des Servers, wie er für das Übertragungsprotokoll definiert ist (TransportType).

Beispiel unter Windows:

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

oder, auf UNIX and Linux-Systemen:

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

**Anmerkung:** Informationen zum Ändern der TCP/IP-Anschlussnummer finden Sie in „MQSERVER“ auf Seite 88.

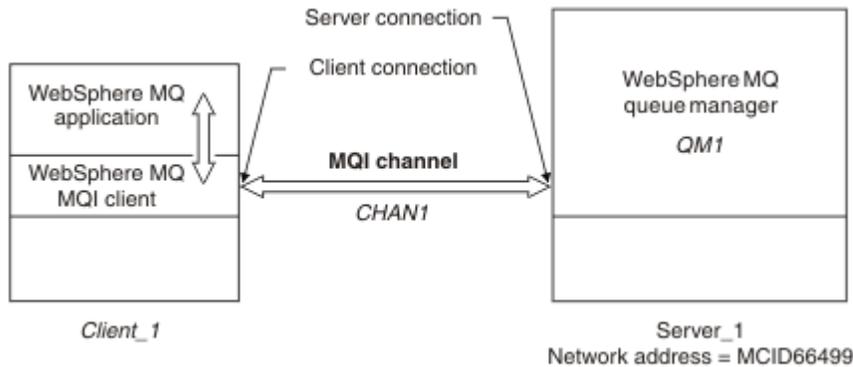


Abbildung 1. Einfache Kanaldefinition

Im Folgenden sind einige weitere Beispiele für einfache Kanaldefinitionen enthalten:

- Unter Windows:

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

- Auf UNIX and Linux-Systemen:

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

- **IBM i** Unter IBM i:

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

Hierbei steht BOX99 für den LU 6.2-Verbindungsnamen.

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.

**Anmerkung:** Weitere Informationen zur Umgebungsvariablen *MQSERVER* finden Sie in „MQSERVER“ auf Seite 88.

### MQCNO-Struktur in einem MQCONNX-Aufruf verwenden

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. Weitere Informationen in Bezug auf TLS können in der Struktur der TLS-Konfigurationsoptionen (MQSCO) bereitgestellt werden, auf die auch von der MQCNO-Struktur in einem **MQCONN** -Aufruf verwiesen wird.

Weitere Informationen zu den MQCNO-, MQCD- und MQSCO-Strukturen finden Sie in [MQCNO](#), [MQCD](#) und [MQSCO](#).

**Anmerkung:** Das Beispielprogramm für MQCONN heißt **amqscnxc**. Ein weiteres Beispielprogramm namens **amqssslc** veranschaulicht die Verwendung der MQSCO-Struktur.

## 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.
-   Unter Linux und Windows können Sie auch IBM MQ Explorer verwenden.
-  Unter z/OS können Sie auch die Fenster "Operation" und "Control" verwenden.
-  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 47.
2. Informationen zum Definieren eines Clientverbindungskanals finden Sie in [„Definieren des Clientverbindungskanals auf dem Server“](#) auf Seite 48.

### Zugehörige Konzepte

[„Definitionstabelle für den Clientkanal“](#) auf Seite 43

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-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

### Zugehörige Tasks

[„Serververbindungskanal auf dem Server definieren“](#) auf Seite 47

Erstellen Sie eine Serververbindungskanaldefinition für den Warteschlangenmanager.

[„Definieren des Clientverbindungskanals auf dem Server“](#) auf Seite 48

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

[„Zugreifen auf Clientverbindungskanaldefinitionen“](#) auf Seite 49

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. **V 9.0.0** Ab IBM MQ 9.0 bietet das Produkt auch die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren.

## Definitionstabelle für den Clientkanal

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-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

Der Zweck der Clientkanaldefinitionstabelle (CCDT) besteht darin, die Kanaldefinitionen zu ermitteln, die von Clientanwendungen für die Verbindung zum Warteschlangenmanager verwendet werden. Die Kanaldefinition gibt auch die Authentifizierungsinformationen an, die für die Verbindungen gelten.

Die CCDT ist eine Binärdatei. Sie wird von einem WS-Manager generiert. Der WS-Manager liest die CCDT-Datei nicht.

**Multi** Unter Multiplatforms wird die CCDT beim Erstellen des Warteschlangenmanagers erstellt. Die CCDT, die einem Warteschlangenmanager zugeordnet ist, wird mit den Objektdefinitionen synchronisiert. Daher werden beim Definieren, Ändern oder Löschen eines Clientkanalobjekts sowohl die Warteschlangenmanagerobjektdefinition als auch der Eintrag in der CCDT im Rahmen derselben 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 Clientverbindungskanal 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
```

Sie können die CCDT verwenden, um Clients die Authentifizierungsinformationen zur Verfügung zu stellen, die auf den Widerruf von TLS-Zertifikaten hin überprüft werden sollen. Definieren Sie eine Namensliste, die Authentifizierungsinformationsobjekte enthält, und setzen Sie das WS-Managerattribut **SSLCRLNameList** auf den Namen der Namensliste.

## Standard-CCDT AMQCLCHL . TAB

**Multi**

Unter Multiplatforms wird beim Erstellen eines Warteschlangenmanagers eine CCDT-Standarddatei 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/QUEEMANAGERNAME/&ipcc
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
/prefix/qmgrs/QUEEMANAGERNAME/@ipcc
```

Beim Namen des Verzeichnisses, auf das von *QUEEMANAGERNAME* verwiesen wird, ist auf UNIX and Linux-Systemen die Groß-/Kleinschreibung zu beachten. 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
```

`MQ_INSTALLATION_PATH` steht 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 Verzeichnis `@ipcc` der von Ihnen angegebenen `DataPath`.

Der Pfad zu der CCDT kann durch Einstellung von `MQCHLLIB` geändert werden. Wenn Sie `MQCHLLIB` festlegen, müssen Sie, wenn Sie mehrere WS-Manager auf demselben Server haben, dieselbe CCDT-Position gemeinsam nutzen.

Die CCDT wird beim Erstellen des Warteschlangenmanagers erstellt. Jeder Eintrag einer CCDT stellt eine Clientverbindung zu einem bestimmten Warteschlangenmanager dar. Wenn Sie einen Clientverbindungskanal mit dem Befehl **DEFINE CHANNEL** definieren, wird ein neuer Eintrag hinzugefügt, und der Eintrag wird aktualisiert, wenn Sie die Clientverbindungskanäle mit dem Befehl **ALTER CHANNEL** ändern.

## Positionen für die Definitionstabelle für den Clientkanal

Eine Clientanwendung kann eine CCDT auf verschiedene Arten verwenden. Die CCDT kann auf den Client-Computer kopiert werden. 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.

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.

**V 9.0.0** Ab IBM MQ 9.0 kann die CCDT an einer zentralen Position gehostet werden, auf die über eine URI zugegriffen werden kann, womit die Notwendigkeit entfällt, die CCDT für jeden implementierten Client einzeln zu aktualisieren. In IBM MQ 9.0 wurde für native Anwendungen (C/C++, COBOL und RPG) und nicht verwaltete .NET-Anwendungen die Funktion hinzugefügt, die CCDT von einer URL zu extrahieren, unabhängig davon, ob es sich um eine lokale Datei, eine FTP- oder eine HTTP-Ressource handelt.



**Achtung:** IBM MQ unterstützt das Abrufen einer CCDT aus einer Datei-, FTP- oder HTTP-URL.

**V 9.0.0** 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/CCDTUrlOffset` über `MQCNO`-Struktur wird an `MQCONN` `MQI`-Aufruf übergeben
- `MQCCDTURL`, Umgebungsvariable
- Attribut "ChannelDefinitionDirectory" in der Zeilengruppe "Channels" von `mqclient.ini`

**V 9.0.0** 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
```

**V 9.0.0** Wenn Sie diese Unterstützung mit ftp oder http verwenden wollten, bedeutet dies, dass Sie die CCDT-Datei trotzdem auf einem Server hosten mussten; mit der Unterstützung, die in IBM MQ 9.0 hinzugefügt wurde, konnten hingegen alle Ihre Clientanwendungen Änderungen an Kanaldefinitionen automatisch abholen, ohne dass Aktualisierungen manuell ausgespielt werden müssen oder dass auf jedem Client ein Netzdateisystem angehängt werden muss. Weitere Informationen finden Sie im Abschnitt „Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf Seite 45.

## **runmqsc verwenden, um eine CCDT-Datei direkt auf einer Clientmaschine zu erstellen**

Ab IBM MQ 8.0 können Sie eine CCDT-Datei auf dem Clientsystem direkt mit dem Befehl `runmqsc` mit dem Parameter `-n` erstellen. Die CCDT wird an der durch `MQCHLLIB` angegebenen Position und mit dem Dateinamen erstellt, der durch `MQCHLTAB` angegeben wird. Dies ist `AMQCLCHL.TAB` standardmäßig.

**Wichtig:** Wenn Sie den Parameter `-n` angeben, dürfen Sie keinen anderen Parameter angeben.

Jeder Eintrag einer CCDT stellt eine Clientverbindung zu einem bestimmten Warteschlangenmanager dar. Wenn Sie einen Clientverbindungskanal mit dem Befehl **DEFINE CHANNEL** definieren, wird ein neuer Eintrag hinzugefügt, und der Eintrag wird aktualisiert, wenn Sie die Clientverbindungskanäle mit dem Befehl **ALTER CHANNEL** ändern.

## **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` die Attribute `ChannelDefinitionDirectory`, um das Verzeichnis anzugeben, in dem sich die Tabelle befindet, und `ChannelDefinitionFile`, um den Namen der Datei anzugeben.
- **V 9.0.0** Durch die Bereitstellung einer URL (Datei, FTP oder HTTP) für eine CCDT, die an einer zentralen Position gehostet wird (siehe „Positionen für die Definitionstabelle für den Clientkanal“ auf Seite 44).

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.

**V 9.0.0** 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 „Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf Seite 45.

### **Zugehörige Verweise**

„MQCHLLIB“ auf Seite 84

`MQCHLLIB` gibt den Verzeichnispfad für die 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.

### **Zugehörige Informationen**

Mit widerrufenden Zertifikaten arbeiten

## **V 9.0.0 Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal**

Ab IBM MQ 9.0 bietet das Produkt die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren, entweder durch Programmierung mit `MQCNO`, mithilfe von Umgebungsvariablen oder mithilfe von `mqclient.ini`-Zeilengruppen.



**Achtung:** Sie können die Option "Umgebungsvariable" nur für native Programme verwenden, die als Clients verbunden sind, d. B. 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-, FTP- oder HTTP-URL.

Mit der Umgebungsvariablen „MQCCDTURL“ auf Seite 82 können Sie eine Datei-, FTP- oder HTTP-URL als einen einzigen Wert bereitstellen, von dem aus eine Clientkanaldefinitionstabelle abgerufen werden kann.

Sie können auch „MQCHLLIB“ auf Seite 84 (oder das von **ChannelDefinitionDirectory** unter der „Zeilegruppe 'CHANNELS' in der Clientkonfigurationsdatei“ auf Seite 68 angegebene) verwenden, um eine CCDT-Datei entweder über Datei-, FTP- oder HTTP-URL neben dem vorhandenen lokalen Dateisystemverzeichnis, d. h. /var/mqm, zu finden.

Beachten Sie, dass ein „MQCHLLIB“ auf Seite 84-Wert ein Verzeichnisstamm ist und in Kombination mit „MQCHLTAB“ auf Seite 86 funktioniert, um die vollständig qualifizierte URL abzuleiten.

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 bereitgestellt von **ClientConnOffset** und **ClientConnPtr** in MQCNO.
2. URL, die von **CCDTUrlOffset** und **CCDTUrlPtr** in MQCNO bereitgestellt wird.
3. Umgebungsvariable von „MQSERVER“ auf Seite 88.
4. Wenn eine mqclient.ini -Datei definiert ist und eine ServerConnectionParms-Datei enthält, wird der von dieser Datei definierte Kanal verwendet. Weitere Informationen finden Sie unter „Client mit einer Konfigurationsdatei konfigurieren“ auf Seite 56 und „Zeilegruppe 'CHANNELS' in der Clientkonfigurationsdatei“ auf Seite 68.
5. Umgebungsvariable von „MQCCDTURL“ auf Seite 82.
6. Umgebungsvariable von „MQCHLLIB“ auf Seite 84 und „MQCHLTAB“ auf Seite 86.
7. **ChannelDefinitionDirectory** in der „Zeilegruppe 'CHANNELS' in der Clientkonfigurationsdatei“ auf Seite 68.

**Wichtig:** Der Zugriff auf eine CCDT-Datei unter Verwendung einer URL öffnet immer nur eine schreibgeschützte Kopie der Datei, auch wenn sie das Protokoll file:// verwendet.

Attempting to open a CCDT file for write access, for example when using the **runmqsc** DEFINE CHANNEL command from a client, returns an error message indicating that the file could not be opened for write access.

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

### Zugehörige Konzepte

„Definitionstabelle für den Clientkanal“ auf Seite 43

Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungs-Informationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlan-

genmanager herzustellen. Auf Multiplatforms-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

### Zugehörige Tasks

„Zugreifen auf Clientverbindungskanaldefinitionen“ auf Seite 49

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. **V 9.0.0** Ab IBM MQ 9.0 bietet das Produkt auch die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren.

### Zugehörige Informationen

[CCDTURL](#)

[Verwenden einer CCDT mit IBM MQ classes for JMS](#)

[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/Serververbindung bereitzustellen.

Wenn Clientverbindungskanalobjekte definiert sind, werden sie standardmäßig in eine Clientkanaldefinitionsdatei mit dem Namen AMQCLCHL.TAB geschrieben. 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. MQCD-Datenstruktur MQCONNX
2. Umgebungsvariable MQSERVER
3. Clientkanaldefinitionsdatei
4. Active Directory

Diese Reihenfolge bedeutet, dass alle aktuellen Anwendungen von keiner Änderung betroffen sind. Sie 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`.

Beispiel:

```
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 Kanals, der im vorherigen Schritt definiert wurde.
- 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.

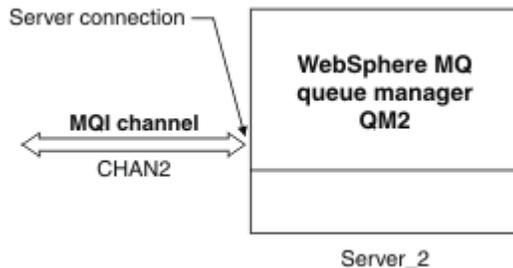


Abbildung 2. Serververbindungskanal definieren

## Definieren des Clientverbindungskanals auf dem Server

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

### Vorbereitende Schritte

Definieren Sie den Serververbindungskanal.

### Vorgehensweise

1. Definieren Sie einen Kanal mit demselben Namen wie der Serververbindungskanal, aber einen Kanaltyp von *Clientverbindung*. Sie müssen den Verbindungsname (CONNNAME) 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) +
CONNNAME(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 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](#).

## 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 „Definitionstabelle für den Clientkanal“ auf Seite 43.

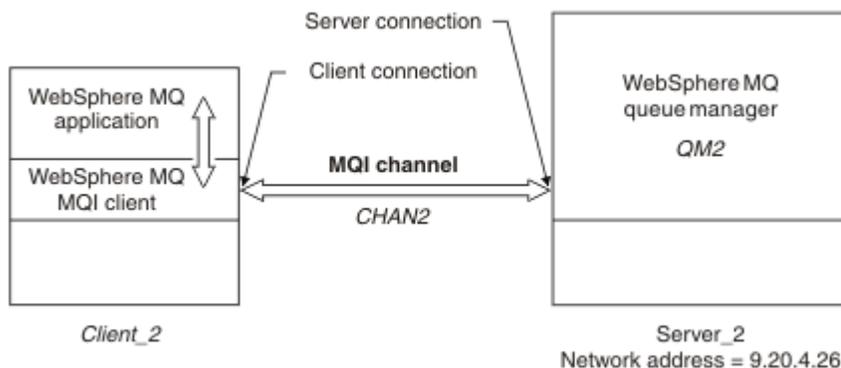


Abbildung 3. Definieren des Clientverbindungskanals

## 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. **V 9.0.0** Ab IBM MQ 9.0 bietet das Produkt auch die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren.

## Vorbereitende Schritte

Sie haben die Clientverbindungskanäle definiert, die Sie benötigen.

**z/OS** Unter z/OS haben Sie eine CCDT erstellt.

**Multi** Unter Multiplatforms wird die CCDT automatisch erstellt und aktualisiert.

## 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.

**V 9.0.0** Ab IBM MQ 9.0 können IBM MQ-, native (C/C++, COBOL und RPG) und nicht verwaltete .NET-Anwendungen die CCDT in einer zentralen Position aus einer URL extrahieren, unabhängig davon, ob es sich um eine lokale Datei, eine FTP- oder eine 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 68](#).
- b) Optional: Verwenden Sie die Umgebungsvariablen MQCHLLIB und MQCHLTAB.

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

- Auf UNIX and Linux-Systemen:

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

-  Unter 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, mithilfe von Umgebungsvariablen oder durch die Verwendung von `mqclient.ini`-Dateizeilengruppen. Weitere Informationen finden Sie unter „Positionen für die Definitionstabelle für den Clientkanal“ auf [Seite 44](#) und „Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf [Seite 45](#).

Wenn die Umgebungsvariable MQSERVER gesetzt ist, hat die von MQSERVER vorgegebene Clientverbindungskanaldefinition Vorrang gegenüber eventuell in der Clientkanal-Definitionstabelle vorhandenen Definitionen und wird vom IBM MQ-Client verwendet.

### Zugehörige Konzepte

„Definitionstabelle für den Clientkanal“ auf [Seite 43](#)

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-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

„Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf [Seite 45](#)

Ab IBM MQ 9.0 bietet das Produkt die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren, entweder durch Programmierung mit MQCNO, mithilfe von Umgebungsvariablen oder mithilfe von `mqclient.ini`-Zeilengruppen.

### Zugehörige Informationen

[MQI-Client: Definitionstabelle für Clientkanäle \(CCDT\)](#)

Es sind drei Typen von Kanalexits für die IBM MQ MQI client-Umgebung unter UNIX, 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 52](#)

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.

### **Zugehörige Verweise**

[„Pfad zu Exits“ auf Seite 51](#)

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 53](#)

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.

### **Zugehörige Informationen**

[Funktionen des Warteschlangenmanagers erweitern](#)

[Benutzerexits, API-Exits und installierbare IBM MQ-Services](#)

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

Auf UNIX, 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.

## 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 53 veranschaulicht die Verwendung von Sicherheitsexits in einer Clientverbindung, wobei der IBM MQ-Objektberechtigungsmanager verwendet wird, um einen Benutzer zu authentifizieren. Entweder 'SecurityParmsPtr' oder 'SecurityParmsOffset' wird in der MQCNO-Struktur auf dem Client definiert, und es sind Sicherheitsexits an beiden Enden des Kanals vorhanden. Nachdem der normale Sicherheitsnachrichtenaustausch beendet wurde und der Kanal für die Ausführung bereit ist, wird die MQCSP-Struktur, auf die über das Feld MQCXP SecurityParms zugegriffen wird, an den Sicherheitsexit auf dem Client übergeben. Der Exittyp wird auf MQXR\_SEC\_PARMS gesetzt. Der Sicherheitsexit kann die Benutzer-ID und das Kennwort nicht ändern, oder es kann eine oder beide der beiden Änderungen ändern. Die vom Exit zurückgegebenen Daten werden dann an das Serververbindungsende des Kanals gesendet. Die MQCSP-Struktur wird auf dem Serververbindungsende des Kanals wiederhergestellt und an den Sicherheitsexit der Serververbindung übergeben, auf den über das Feld MQCXP SecurityParms zugegriffen wird. Der Sicherheitsexit empfängt und verarbeitet diese Daten. Bei dieser Verarbeitung wird normalerweise jede Änderung rückgängig gemacht, die an den Feldern für Benutzer-ID und Kennwort im Clientexit vorgenommen wurde, die dann zur Autorisierung der Verbindung des Warteschlangenmanagers verwendet werden. Die resultierende MQCSP-Struktur wird unter Verwendung von SecurityParmsPtr in der MQCNO-Struktur auf dem WS-Manager-System referenziert.

Die Speicheradresse, die vom MQCXP-Feld SecurityParms übergeben wird, muss bis zum 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 'SecurityParmsPtr' oder 'SecurityParmsOffset' in der MQCNO-Struktur definiert sind und ein Sicherheitsexit nur an einem Ende des Kanals 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 'SecurityParmsPtr' und 'SecurityParmsOffset' nicht in der MQCNO-Struktur definiert sind und ein Sicherheitsexit an einem oder beiden Enden des Kanals vorhanden ist, werden die Sicherheitsexits oder -exits aufgerufen. Der Sicherheitsexit kann seine eigene MQCSP-Struktur zurückgeben, die über die SecurityParmsPtr-Datei adressiert wird. Der Sicherheitsexit wird erst wieder aufgerufen, wenn er beendet wird (ExitReason von MQXR\_TERM). Der Exit-Writer kann den Speicher, der für den MQCSP in dieser Phase verwendet wird, freilassen.

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 Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

Abbildung 4. Clientverbindung-Eingeganger 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 [Kanalsicherheits-Exitprogramme](#).

## 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 Kanalfusses angezeigt werden, wenn ein API-Aufruf verarbeitet wird.

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

<i>Tabelle 7. API-Aufrufe identifizieren</i>		
<b>API-Aufruf</b>	<b>Wert von Byte 10 für Anforderung</b>	<b>Wert von Byte 10 für Antwort</b>
MQCONN „1“ auf Seite 54, „2“ auf Seite 54	X'81 '	X' 91 '
MQDISC „1“ auf Seite 54	X'82 '	X' 92 '
MQOPEN „3“ auf Seite 54	X'83 '	X' 93 '
MQCLOSE	X'84 '	X' 94 '
MQGET „4“ auf Seite 54	X'85 '	X' 95 '
MQPUT „4“ auf Seite 54	X'86 '	X' 96 '
MQPUT1-Anforderung „4“ auf Seite 54	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.

## 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 in [Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange](#).

Ein Client, der Nachrichten in eine gemeinsam genutzte Warteschlange einreicht, 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 Client-Verbindungskanal gespeichert wird, können Sie jetzt eine Verbindung zu jedem Warteschlangenmanager in der Gruppe herstellen und Nachrichten in gemeinsam genutzte Warteschlangen dieser Gruppe einreihen.

## Client mit einer Konfigurationsdatei konfigurieren

Sie konfigurieren Ihre Clients mit Hilfe von Attributen in einer Textdatei. Diese Attribute können durch Umgebungsvariablen oder auf andere plattformspezifische Methoden überschrieben werden.

### Informationen zu diesem Vorgang

Sie konfigurieren Ihre IBM MQ MQI clients mithilfe einer Textdatei, ähnlich der WS-Manager-Konfigurationsdatei `qm.ini`, die auf UNIX and Linux-Plattformen verwendet wird. 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 alle Plattformen und 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 58.

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)

```

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.

## Linux UNIX **Kommentare in der Konfigurationsdatei**

Sie können das Semikolon ';' und den Hashwert '#' 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 für einen Wert eines dieser Zeichen erforderlich ist, müssen Sie dieses Zeichen mit dem Backslash-Zeichen '\' verlassen.

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-
  capped comment characters.

```

## Prozedur

- Verwenden Sie die Informationen in den folgenden Themen, um Sie bei der Konfiguration Ihrer Clients zu unterstützen:
  - [„Position der Clientkonfigurationsdatei“](#) auf Seite 57
  - [„Welche IBM MQ-Clients können die einzelnen Attribute lesen“](#) auf Seite 58

## 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 Position, die durch die Umgebungsvariable MQCLNTCF angegeben wird.

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.

Dabei ist Folgendes zu beachten:

- C-, .NET- und XMS-Clients unterstützen nur das Protokoll `file:`. Das `file:`-Protokoll wird vorausgesetzt, wenn die URL-Zeichenfolge nicht mit `protocol:` beginnt.
  - Um Java 1.4.2 JREs zu ermöglichen, die nicht das Lesen von Umgebungsvariablen unterstützen, kann die Umgebungsvariable `MQCLNTCF` mit einer Systemeigenschaft 'MQCLNTCF Java' 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 Datenverzeichnis von IBM MQ für Windows-, UNIX and Linux-Systeme.

Dabei ist Folgendes zu beachten:

- 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.
- Auf UNIX and Linux-Systemen ist das Verzeichnis `/var/mqm`.
- Auf Windows -Plattformen konfigurieren Sie die Umgebungsvariable `MQ_DATA_PATH` während der Installation so, dass sie auf das Datenverzeichnis verweist. Sie ist normalerweise `C:\ProgramData\IBM\MQ`.

**Anmerkung:** Wenn Sie nur einen Client installieren, kann die Umgebungsvariable `MQ_FILE_PATH` lauten.

- Um Java 1.4.2 JREs zuzulassen, die das Lesen von Umgebungsvariablen nicht unterstützen, können Sie die Umgebungsvariable `MQ_DATA_PATH` manuell mit einer `MQ_DATA_PATH Java` -Systemeigenschaft 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`.
    - Für C-Clients auf UNIX and Linux-Plattformen ist dies der Wert der Umgebungsvariablen `HOME`.
    - Für C-Clients unter Windows sind dies die verketteten Werte der Umgebungsvariablen `HOMEDRIVE` und `HOMEPATH`.

## Welche IBM MQ-Clients können die einzelnen Attribute lesen

Die meisten der Attribute in der IBM MQ MQI client-Konfigurationsdatei können vom C-Client und den nicht verwalteten .NET-Clients verwendet werden. Es gibt jedoch einige Attribute, die von verwalteten .NET- und XMS .NET-Clients nicht gelesen werden, oder von Clients, die entweder IBM MQ classes for Java oder IBM MQ classes for JMS verwenden.

<i>Tabelle 8. Welche Attribute gelten für jeden Typ von Client</i>						
<b>mqclient.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<b>CHANNELS-Zeilengruppe</b>						

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<u>CCSID</u>	Die Nummer des codierten Zeichensatzes, der verwendet werden soll.	Ja	Nein	Nein	Ja	Ja
<u>ChannelDefinitionDirectory</u>	Der Verzeichnispfad zu der Datei, die die Definitionstabelle für den Clientkanal enthält.	Ja	Nein	Nein	Ja	Ja
<u>ChannelDefinitionFile</u>	Der Name der Datei, die die Definitionstabelle für den Clientkanal enthält.	Ja	Nein	Nein	Ja	Ja
<u>ReconDelay</u>	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

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<u>DefRecon</u>	Eine Verwaltungsoption, mit der Clientprogramme automatisch erneuert 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
<u>MQReconnectTimeout</u>	Das Zeitlimit in Sekunden, in dem die Verbindung zu einem Client wieder hergestellt werden kann.	Ja	Nein	Nein	Ja	Nein
<u>ServerConnectionParms</u>	Die Position des IBM MQ-Servers und das zu verwendende Kommunikationsverfahren.	Ja	Nein	Nein	Ja	Ja

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<a href="#">Put1DefaultAlwaysSync</a>	Steuert das Verhalten des Funktionsaufrufs MQPUT1 mit der Option MQPMO_RESPONSE_AS_QDEF.	Ja	Ja	Ja	Ja	Ja
<a href="#">Password-Protection</a>	Hier können Sie geschützte Kennwörter in der MQCSP-Struktur festlegen, statt SSL oder TLS zu verwenden.	Ja	Ja	Ja	Ja	Ja
<b>Zeilengruppe 'ClientExitPath'</b>						
<a href="#">ExitsDefaultPath</a>	Gibt die Position der 32-Bit-Kanal-exits für Clients an.	Ja	Ja	Ja	Ja	Ja
<a href="#">ExitsDefaultPath64</a>	Gibt die Position der 64-Bit-Kanal-exits für Clients an.	Ja	Ja	Ja	Ja	Ja
<a href="#">JavaExitsClassPath</a>	Die Werte, die dem Klassenpfad hinzugefügt werden sollen, wenn ein Java-Exit ausgeführt wird.	Nein	Ja	Ja	Nein	Nein
<b>JMQI-Zeilengruppe</b>						

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<u>useMQCSPauthentication</u>	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
<b>Zeilengruppe 'MessageBuffer'</b>						
<u>Maximum-Size</u>	Größe (in Kilobyte) des Read-Ahead-Puffers im Bereich von 1 bis 999 999.	Ja	Ja	Ja	Ja	Ja
<u>PurgeTime</u>	Intervall (in Sekunden), nach dem Nachrichten, die im Puffer für die Lesepuffer (Read-ahead-Puffer) bleiben, gelöscht	Ja	Ja	Ja	Ja	Ja

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<u>UpdatePercentage</u>	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
<b>Zeilegruppe 'PreConnect'</b>						
<u>Daten</u>	URL des Repositorys, in dem Verbindungsdefinitionen gespeichert werden.	Ja	Nein	Nein	Nein	Nein
<u>Funktion</u>	Der Name des funktionalen Eingangspunkts in der Bibliothek, die den PreConnect-Exit-Code enthält.	Ja	Nein	Nein	Nein	Nein
<u>Modul</u>	Der Name des Moduls, das den API-Exit-Code enthält.	Ja	Nein	Nein	Nein	Nein

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<u>Reihenfolge</u>	Die Sequenz, in der dieser Exit relativ zu anderen Exits aufgerufen wird.	Ja	Nein	Nein	Nein	Nein
<b>Sicherheitszeilengruppe</b>						
<u>DisableClientAMS</u>	Aktiviert oder inaktiviert AMS für Clientverbindungen zu einem Warteschlangenmanager.	Ja	Ja	Ja	Nein	Nein
<b>SSL-Zeilengruppe</b>						
<u>CDPCheckExtensions</u>	Gibt an, ob SSL-oder TLS-Kanäle in diesem Warteschlangenmanager versuchen, CDP-Server zu überprüfen, die in den Zertifikatserweiterungen des CrIDistributionPoint-Zertifikats benannt sind.	Ja	Nein	Nein	Nein	Nein
<u>CertificateLabel</u>	Die Zertifikatsbezeichnung der Kanaldefinition.	Ja	Nein	Nein	Nein	Nein
<u>CertificateValPolicy</u>	Bestimmt den Typ der verwendeten Zertifikatsprüfung.	Ja	Nein	Nein	Nein	Nein

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<a href="#">ClientRevocationChecks</a>	Legt fest, wie die Überprüfung der Zertifikatswiderrufung konfiguriert wird, wenn der Clientverbindungsaufruf einen SSL/TLS-Kanal verwendet.	Ja	Nein	Nein	Nein	Nein
<a href="#">EncryptionPolicySuiteB</a>	Legt fest, ob ein Kanal eine Suite-B-kompatible Verschlüsselung verwendet und welcher Grad der Stärke verwendet werden soll.	Ja	Nein	Nein	Nein	Nein
<a href="#">OCSPAuthentication</a>	Definiert das Verhalten von IBM MQ, wenn OCSP aktiviert ist und die OCSP-Widerrufprüfung nicht in der Lage ist, den Status des Zertifikatswiderrufs zu ermitteln.	Ja	Nein	Nein	Nein	Nein
<a href="#">OCSPCheckExtensions</a>	Steuert, ob IBM MQ auf die AuthorityInfoAccess-Zertifikatserweiterungen einwirkt.	Ja	Nein	Nein	Nein	Nein

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<a href="#">SSLCryptoHardware</a>	Legt die Parameterzeichenfolge fest, die erforderlich ist, um die auf dem System vorhandene Verschlüsselungshardware PKCS #11 zu konfigurieren.	Ja	Nein	Nein	Nein	Nein
<a href="#">SSLFipsRequired</a>	Gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen, wenn die Verschlüsselung in IBM MQ ausgeführt wird.	Ja	Nein	Nein	Nein	Nein
<a href="#">SSLHTTPProxyName</a>	Die Zeichenfolge ist entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers, der von GSKit für OCSP-Prüfungen verwendet werden soll.	Ja	Nein	Nein	Nein	Nein

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

<b>mqcli-ent.ini, Zeilengruppenname und Attribute</b>	<b>Beschreibung</b>	<b>C und nicht verwaltete .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Verwaltete .NET</b>	<b>Verwaltete XMS .NET</b>
<u>SSLKeyRepository</u>	Die Position des Schlüsselrepositorys, in dem das digitale Zertifikat des Benutzers gespeichert ist, im Stammformat.	Ja	Nein	Nein	Nein	Nein
<u>SSLKeyResetCount</u>	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
<b>TCP-Zeilengruppe</b>						
<u>ClntRcvBuffSize</u>	Die Größe des TCP/IP-Empfangspuffers (in Byte), der vom Clientende eines Clientverbindungs-Serververbindungskanal verwendet wird.	Ja	Ja	Ja	Ja	Ja

Tabelle 8. Welche Attribute gelten für jeden Typ von Client (Forts.)

mqcli-ent.ini, Zeilengruppenname und Attribute	Beschreibung	C und nicht verwaltete .NET	Java	JMS	Verwaltete .NET	Verwaltete XMS .NET
<u>ClntSndBuff-Size</u>	Die Größe des TCP/IP-Sendepuffers (in Byte), der vom Clientende eines Clientverbindungs-Serververbindungskanals verwendet wird.	Ja	Ja	Ja	Ja	Ja
<u>Connect_Timeout</u>	Die Anzahl der Sekunden, bevor ein Versuch unternommen wird, das Socket-Zeitlimit zu verbinden.	Ja	Ja	Ja	Nein	Nein
<u>IPAddress-Version</u>	Gibt das IP-Protokoll an, das für eine Kanalverbindung verwendet werden soll.	Ja	Nein	Nein	Ja	Ja
<u>KeepAlive</u>	Schaltet die KeepAlive-Funktion ein oder aus.	Ja	Ja	Ja	Ja	Ja
 <u>Library1</u>	Nur unter Windows: der Name der TCP/IP-Sockets-DLL.	Ja	Nein	Nein	Nein	Nein

## 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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt [Welche IBM MQ-Clients können die einzelnen Attribute lesen.](#)

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 dem Parameter der MQCCSID-Umgebung.

**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 -Protokolldateiverzeichnis (normalerweise C:\ProgramData\IBM\MQ).

**Linux** **UNIX** Auf UNIX and Linux-Systemen ist das Standardverzeichnis /var/mqm.

**V9.0.0** ChannelDefinitionDirectory kann eine URL enthalten, die in Kombination mit dem Attribut "ChannelDefinitionFile" funktioniert (siehe „Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf Seite 45).

Der Pfad "ChannelDefinitionDirectory" entspricht dem Umgebungsparameter 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 ist äquivalent zum Umgebungsparameter 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 Wiederherstellung einer Clientverbindung wird von IBM MQ Classes for Java nicht unterstützt.

Die Interpretation der Optionen DefRecon hängt davon ab, ob ein MQCNO\_RECONNECT\_\* -Wert auch im Clientprogramm festgelegt ist und welcher Wert festgelegt ist.

Wenn das Clientprogramm eine Verbindung mit MQCONNherstellt oder die Option MQCNO\_RECONNECT\_AS\_DEF mit MQCONNXfestlegt, wird der von DefRecon festgelegte Wert für die Verbindungswiederherstellung wirksam. Wenn im Programm kein Wert für die Verbindungswiederherstellung oder die Option DefRecon festgelegt ist, wird das Clientprogramm nicht automatisch erneut verbunden.

**NO**

Sofern nicht von **MQCONNX**überschrieben, wird die Clientverbindung nicht automatisch wiederhergestellt.

**YES**

Wenn nicht durch **MQCONNX**überschrieben, stellt der Client die Verbindung automatisch wieder her.

**QMGR**

Wenn nicht durch **MQCONNX**überschrieben, stellt der Client die Verbindung automatisch wieder her, aber nur mit demselben Warteschlangenmanager. Die Option QMGR hat dieselbe Wirkung wie MQCNO\_RECONNECT\_Q\_MGR.

**INAKTIVIERT**

Die Verbindungswiederholung ist inaktiviert, auch wenn sie vom Clientprogramm mit dem MQI-Aufruf **MQCONNX** angefordert wird.

Die automatische Neuverbindung des Clients hängt von zwei Werten ab:

- Die in der Anwendung festgelegte Verbindungswiederherstellungsoption
- DefRecon -Wert in der mqclient.ini -Datei

*Tabelle 9. Die automatische Verbindungswiederherstellung hängt von den Werten ab, die in der Anwendung und in der Datei mqclient.ini festgelegt sind.*

DefRecon-Wert in der mqclient.ini	In der Anwendung festgelegte Verbindungswiederholungsoptionen			
	MQCNO_RECONNECT	MQCNO_RECONNECT_Q_MGR	MQCNO_RECONNECT_AS_DEF	MQCNO_RECONNECT_DISABLED
NO	YES	QMGR	Nein	Nein
YES	YES	QMGR	YES	Nein
QMGR	YES	QMGR	QMGR	Nein
INAKTIVIERT	Nein	Nein	Nein	Nein

**MQReconnectTimeout**

Das Zeitlimit in Sekunden, in dem die Verbindung zu einem Client wieder hergestellt werden kann. 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

ServerConnectionParms entspricht dem MQSERVER-Umgebungsparameter und gibt den Standort des IBM MQ-Servers und das zu verwendende Kommunikationsverfahren 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 | JA

Steuert das Verhalten des Funktionsaufrufs MQPUT1 mit der Option MQPMO\_RESPONSE\_AS\_Q\_DEF.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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.

#### YES

MQPUT1 verhält sich so, als ob MQPMO\_SYNC\_RESPONSE gesetzt ist, unabhängig davon, ob MQPMO\_SYNCPOINT oder MQPMO\_NO\_SYNCPOINT festgelegt ist.

### PasswordProtection = Kompatibel | immer | optional

Ab IBM MQ 8.0 können Sie geschützte Kennwörter in der MQCSP-Struktur festlegen, anstatt TLS zu verwenden.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java- und IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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 finden Sie im Abschnitt [MQCSP-Kennwortschutz](#).

## Zugehörige Informationen

IBM MQ-MQI-Anwendungen mit Warteschlangenmanagern verbinden

## 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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt Welche IBM MQ-Clients können die einzelnen Attribute lesen.

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." und dieser vollständige Name wird auch in der Systemeigenschaft IBM WebSphere MQ 7.0 oder höher verwendet. In IBM WebSphere MQ 6.0 wurde dieses Attribut mit der Systemeigenschaft 'com.ibm.mq.exitClasspath' angegeben, die in der Readme-Datei von IBM WebSphere MQ 6.0 dokumentiert wurde. Die Verwendung von 'com.ibm.mq.exitClasspath' ist nicht mehr weiter unterstützt. Wenn sowohl 'JavaExitsClassPath' als auch 'exitClasspath' vorhanden sind, wird 'JavaExitsClassPath' berücksichtigt. Wenn nur die Verwendung von exitClasspath vorhanden ist, wird sie immer noch in IBM WebSphere MQ 7.0 oder höher berücksichtigt.

## Zeilengruppe 'JMQUI' der Clientkonfigurationsdatei

Verwenden Sie die Zeilengruppe 'JMQUI', um Konfigurationsparameter für die Java Message Queuing Interface (JMQUI) 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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt Welche IBM MQ-Clients können die einzelnen Attribute lesen.

Das folgende Attribut kann in die Zeilengruppe 'JMQUI' 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 der Standardwert.

**YES**

Verwenden Sie den MQCSP-Authentifizierungsmodus beim Authentifizieren mit einem Warteschlangenmanager.

Weitere Informationen zum Kompatibilitätsmodus und zum MQCSP-Authentifizierungsmodus finden Sie unter [Verbindungsauthentifizierung mit dem Java-Client](#).

## **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.

Dies entspricht dem Umgebungsparameter 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.

## 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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt [Welche IBM MQ-Clients können die einzelnen Attribute lesen](#).

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-, nicht verwalteten .NET-, IBM MQ classes for Java-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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 Lesebuffer (Read-ahead-Puffer) bleiben, gelöscht

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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-, nicht verwalteten .NET-, IBM MQ classes for Java-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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:

$$T = \text{Upper} - \text{Lower}$$

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:

$$\text{Lower} = \text{Upper} \times \text{UpdatePercentage} / 100$$

#### **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

$$T = \text{Upper} - \text{Lower}$$

$T$  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

$$T = \text{Upper} - \text{Lower}$$

$T$  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.

## **Zeilen­gruppe für PreConnect der Clientkonfigurationsdatei**

Verwenden Sie die Zeilen­gruppe 'PreConnect', um den PreConnect-Exit in der Datei `mqclient.ini` zu konfigurieren.

**Anmerkung:** Die Beschreibung jedes Attributs dieser Zeilengruppe zeigt an, welche IBM MQ-Clients dieses Attribut lesen können. Eine Übersichtstabelle aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt Welche IBM MQ-Clients können die einzelnen Attribute lesen.

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 Folgennummer wird vor einem Exit mit einer höheren Folgennummer aufgerufen. Es ist nicht erforderlich, dass die Folgennummerierung 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.

Es können mehrere PreConnect-Zeilengruppen in der Datei `mqclient.ini` definiert werden. Die Verarbeitungsreihenfolge der einzelnen Exit wird durch das Attribut "Sequence" der Zeilengruppe festgelegt.

#### **Zugehörige Informationen**

Verbindungsdefinitionen unter Verwendung eines Vorverbindungsexits aus einem Repository referenzieren

### **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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt Welche IBM MQ-Clients können die einzelnen Attribute lesen.

Das folgende Attribut kann in der Zeilengruppe 'Security' enthalten sein:

#### **DisableClientAMS = NO|YES**

Mit dem Attribut `DisableClientAMS` können Sie IBM MQ Advanced Message Security (AMS) inaktivieren, wenn Sie einen Client von IBM WebSphere MQ 7.5 oder höher verwenden, um eine Verbindung zu einem Warteschlangenmanager von einer früheren Version des Produkts herzustellen, und der Fehler 2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) gemeldet wird.

Ab IBM WebSphere MQ 7.5 wird IBM MQ Advanced Message Security (AMS) automatisch in einem IBM MQ-Client aktiviert, und deshalb versucht der Client standardmäßig, die Sicherheitsrichtlinien für Objekte im Warteschlangenmanager zu prüfen. Auf Servern früherer Versionen des Produkts, z. B. IBM WebSphere MQ 7.1, ist AMS jedoch nicht aktiviert, was dazu führt, dass der Fehler 2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) gemeldet wird.

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.

### Zugehörige Informationen

[Advanced Message Security auf dem Client inaktivieren](#)

## 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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt [Welche IBM MQ-Clients können die einzelnen Attribute lesen](#).

Die folgenden Attribute können in die SSL-Zeilengruppe eingeschlossen werden:

### CDPCheckExtensions = YES | NO

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 : TLS-Kanäle versuchen, die CDP-Server zu überprüfen, um festzustellen, ob ein digitales Zertifikat widerrufen wird.
- 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 Clientverbindungsaufbau einen TLS-Kanal verwendet. Siehe auch **OCSPAAuthentication**.

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 Widerrufskonfiguration 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.

#### **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 Zertifikatswiderrufprüfung 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.

### **OCSPAAuthentication = 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 Widerrufsprüfung 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 'ReasonQualifier' von MQRC\_SSL\_HANDSHAKE\_ERROR generiert. Die Verbindung wurde aufgehoben.

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 MQRC\_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 NO gesetzt ist, ignoriert IBM MQ die AuthorityInfoAccess-Zertifikatserweiterungen und versucht nicht, eine OCSP-Sicherheitsprüfung auszuführen. Der Standardwert ist YES .

## 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;passwd;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. Um auf symmetrische Chiffrieroperationen über GSKit zuzugreifen, geben Sie den Parameter für die symmetrische Verschlüsselungseinstellung 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.

Die maximal zulässige Länge beträgt 256 Zeichen. Standardmäßig erfolgt keine Angabe. Wenn Sie eine Zeichenfolge angeben, die sich nicht im richtigen Format befindet, wird ein Fehler generiert.



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: '\;'

## 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 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.

Dieses Attribut kann von C- und von nicht verwalteten .NET-Clients gelesen werden.

Auf den Plattformen HP-UX PA-RISC und Sun Solaris SPARC und für 32-Bit-Clients unter AIX kann die Netzadresse nur eine IPv4-Adresse sein; auf anderen Plattformen kann es sich um eine IPv4- oder IPv6-Adresse handeln.

Dieses Attribut kann erforderlich sein, wenn z. B. eine Firewall den Zugriff auf die URL des OCSP-Responder verhindert.

**SSLKeyRepository = *Pfadname***

Die Position des Schlüsselrepositorys, in dem das digitale Zertifikat des Benutzers gespeichert ist, im Stammformat. Das heißt, er enthält den vollständigen Pfad und den Dateinamen ohne Erweiterung.

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.

## 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 aller IBM MQ MQI client-Konfigurationsdateizeilen-gruppen finden Sie im Abschnitt [Welche IBM MQ-Clients können die einzelnen Attribute lesen](#).

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-Serververbindungskanal verwendet wird.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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-Serververbindungskanal verwendet wird.

Dieses Attribut kann von C-, nicht verwalteten .NET-, IBM MQ classes for Java-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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. Der Standardwert 0 gibt an, dass kein Verbindungszeitlimit vorhanden ist.

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. 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 Erfolgchancen 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**.

### **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-, nicht verwalteten .NET-, IBM MQ classes for Java-, IBM MQ classes for JMS-, verwalteten .NET- und verwalteten XMS .NET-Clients 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.

## **IBM MQ-Umgebungsvariablen verwenden**

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 Befehle verwenden, um die aktuelle Einstellung anzuzeigen oder um den Wert der Variablen zurückzusetzen. Diese Befehle sind auf allen IBM MQ MQI client-Plattformen verfügbar, sofern nicht anders angegeben.

IBM MQ verwendet Standardwerte für diese Variablen, die Sie nicht festgelegt haben.

**Anmerkung:**  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 die Umgebungsvariablen von IBM MQ weiterhin 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 den Befehl SET MQSERVER=, um den Wert einer Umgebungsvariablen zu entfernen.
  - Verwenden Sie den Befehl SET MQSERVER, um die aktuelle Einstellung einer Umgebungsvariablen anzuzeigen
  - Verwenden Sie den Befehl set, um alle Umgebungsvariablen für die Sitzung anzuzeigen.
-  **Linux**  **UNIX**  
Verwenden Sie in UNIX and Linux für jede Umgebungsvariable die folgenden Befehle, um die aktuelle Einstellung anzuzeigen oder um den Wert einer Variablen zurückzusetzen:
  - Wenn Sie den Wert einer Umgebungsvariablen entfernen möchten, verwenden Sie den Befehl unset MQSERVER.
  - Um die aktuelle Einstellung einer Umgebungsvariable anzuzeigen, verwenden Sie den Befehl echo \$MQSERVER.
  - Wenn Sie alle Umgebungsvariablen für die Sitzung anzeigen möchten, verwenden Sie den Befehl set.

## Zugehörige Tasks

„Client mit einer Konfigurationsdatei konfigurieren“ auf Seite 56

Sie konfigurieren Ihre Clients mit Hilfe von Attributen in einer Textdatei. Diese Attribute können durch Umgebungsvariablen oder auf andere plattformspezifische Methoden überschrieben werden.

## Zugehörige Informationen

[Umgebungsvariablen](#)

## MQCCDTURL

MQCCDTURL bietet die funktional entsprechende Funktionalität zum Festlegen einer Kombination aus den Umgebungsvariablen MQCHLLIB und MQCHLTAB.



**Achtung:** Sie können die Option "Umgebungsvariable" nur für native Programme verwenden, die als Clients verbunden sind, d. B. 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-, FTP- oder HTTP-URL.

MQCCDTURL akzeptiert jedoch nur einen URL-Wert; MQCCDTURL akzeptiert das vorhandene Verzeichnisformat des lokalen Dateisystems nicht.

Wenn Sie MQCCDTURL an Stelle von MQCHLLIB und MQCHLTAB verwenden möchten, um eine lokale Datei zu verwenden, können Sie ein 'file://'-Protokoll verwenden. Daher gilt Folgendes:

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

ist äquivalent zu:

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

## Beispiel für Windows bei Verwendung einer JSON-Datei

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

ist äquivalent zu:

```
set MQCHLLIB=C:\mq-channels  
set MQCHLTAB=CCDT-QMGR1.json
```

### Zugehörige Konzepte

„Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf Seite 45

Ab IBM MQ 9.0 bietet das Produkt die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren, entweder durch Programmierung mit MQCNO, mithilfe von Umgebungsvariablen oder mithilfe von `mqclient.ini`-Zeilengruppen.

„Definitionstabelle für den Clientkanal“ auf Seite 43

Die Definitionstabelle für den Clientkanal (CCDT) bestimmt die Kanaldefinitionen und Authentifizierungsinformationen, die von Clientanwendungen verwendet werden, um eine Verbindung zum Warteschlangenmanager herzustellen. Auf Multiplatforms-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

### Zugehörige Verweise

„MQCHLLIB“ auf Seite 84

MQCHLLIB gibt den Verzeichnispfad für die 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.

„MQCHLTAB“ auf Seite 86

MQCHLTAB gibt den Namen der Datei an, die die Clientkanaldefinitionstabelle (ccdt) enthält. Der Standarddateiname ist AMQCLCHL.TAB.

### Zugehörige Informationen

[CCDTURL](#)

[XMSC\\_WMQ\\_CCDTURL](#)

[IBM MQ-MQI-Anwendungen mit Warteschlangenmanagern verbinden](#)

## MQCCSID

MQCCSID gibt die Nummer des codierten Zeichensatzes an, der verwendet werden soll, und überschreibt den CCSID-Wert, mit dem der Server konfiguriert wurde.

Weitere Informationen hierzu finden Sie im Abschnitt [ID des codierten Zeichensatzes \(CCSID\) des Clients oder des Servers auswählen](#).

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQCCSID=number
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQCCSID=number
```

- **IBM i** Unter IBM i:

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

## MQCERTLABL

MQCERTLABL gibt die Zertifikatsbezeichnung für die Kanaldefinition an.

Weitere Informationen finden Sie unter [Zertifikatsbezeichnung \(CERTLABL\)](#).

## MQCERTVPOL

MQCERTVPOL gibt die Validierungsrichtlinie für Zertifikate an.

Weitere Informationen zu Zertifikatsprüfrichtlinien in IBM MQ finden Sie im Abschnitt [Zertifikatsprüfrichtlinien in IBM MQ](#).

Diese Umgebungsvariable überschreibt die Einstellung *CertificateValPolicy* in der SSL-Zeilengruppe der Client-INI-Datei. Die Variable kann auf einen der folgenden Werte gesetzt werden:

### ANY

Verwenden Sie eine beliebige Zertifikatvalidierungsrichtlinie, die von der zugrunde liegenden Secure Sockets Library unterstützt wird

### RFC5280

Verwenden Sie nur die Zertifikatsprüfung, die mit dem Standard RFC 5280 kompatibel ist.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQCERTVPOL= value
```

- **Linux** **UNIX** Auf UNIX and Linux-Systemen:

```
export MQCERTVPOL= value
```

- **IBM i** Für IBM i:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

## MQCHLLIB

MQCHLLIB gibt den Verzeichnispfad für die 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.

Wenn MQCHLLIB nicht festgelegt ist, nimmt der Pfad für den Client standardmäßig die folgenden Werte an:

- **Windows** Unter Windows: *MQ\_INSTALLATION\_PATH*
- **Linux** **UNIX** Unter UNIX and Linux: */var/mqm/*
- **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.

- **Windows** Unter Windows: *datapath\@ipcc*
- **Linux** **UNIX** Unter UNIX and Linux: *datapath/@ipcc*
- **IBM i** Unter IBM i: *datapath/&ipcc*

Oder:

- **Windows** Unter Windows: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`
- **Linux** **UNIX** Unter UNIX and Linux: `/prefix/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** **UNIX** `/var/mqm` auf UNIX 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:
  - **IBM i** **Linux** **UNIX** In der Datei `mqs.ini` unter IBM i, UNIX und Linux.
  - **Windows** In der Registry unter Windows.

#### Anmerkungen:

1. **z/OS** Wenn Sie IBM MQ für 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. **V 9.0.0** `MQCHLLIB` kann eine URL enthalten, die in Kombination mit der Umgebungsvariablen `MQCHLTAB` funktioniert (siehe „Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf Seite 45).
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, wird `SYSTEM.DEF.CLNTCONN` in der `CCDT` aktualisiert, wenn `MQCHLLIB` festgelegt ist. 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 Clientverbindungsdefinition mit demselben Namen, die zuvor erstellt wurde, jedoch nicht von diesem Warteschlangenmanager, wird unabhängig von der Option `NORE` -

PLACE ersetzt. Wenn die Definition zuvor von demselben WS-Manager erstellt wurde, wird die Definition nicht ersetzt.

- Der Befehl **rczmqobj -t c1chl1tab** 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.
- Andere Befehle, die die CCDT aktualisieren, ändern nur die Clientverbindungskanäle, die denselben Kanalnamen haben. Andere Clientverbindungskanäle in der Datei werden nicht geändert.
- Der Pfad für MQCHLLIB benötigt keine Anführungszeichen.

## Beispiele

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- Windows** Unter Windows:

```
SET MQCHLLIB=pathname
```

Beispiel:

```
SET MQCHLLIB=C:\wmqtest
```

- Linux** **UNIX** Für UNIX and Linux-Systeme:

```
export MQCHLLIB=pathname
```

- IBM i** Für IBM i:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

## Zugehörige Verweise

„MQCHLTAB“ auf Seite 86

MQCHLTAB gibt den Namen der Datei an, die die Clientkanaldefinitionstabelle (ccdt) enthält. Der Standarddateiname ist AMQCLCHL.TAB.

## Zugehörige Informationen

[IBM MQ-MQI-Anwendungen mit Warteschlangenmanagern verbinden](#)

## MQCHLTAB

MQCHLTAB gibt den Namen der Datei an, die die Clientkanaldefinitionstabelle (ccdt) enthält. Der Standarddateiname ist AMQCLCHL.TAB.

Informationen darüber, wo sich die Clientkanaldefinitionstabelle auf einem Server befindet, finden Sie in „Definitionstabelle für den Clientkanal“ auf Seite 43.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- Windows** Unter Windows:

```
SET MQCHLTAB=filename
```

- Linux** **UNIX** Unter UNIX and Linux:

```
export MQCHLTAB=filename
```

- **IBM i** Unter IBM i:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

Beispiel:

```
SET MQCHLTAB=ccdf1.tab
```

In der gleichen Weise wie für den Client gibt die Umgebungsvariable MQCHLTAB auf dem Server den Namen der Definitionstabelle für den Clientkanal an.

### Zugehörige Konzepte

„Webadressierbarer Zugriff auf die Definitionstabelle für den Clientkanal“ auf Seite 45

Ab IBM MQ 9.0 bietet das Produkt die Möglichkeit, eine Clientkanaldefinitionstabelle (CCDT) über eine URL zu lokalisieren, entweder durch Programmierung mit MQCNO, mithilfe von Umgebungsvariablen oder mithilfe von mqclient.ini -Zeilen Gruppen.

### Zugehörige Verweise

„MQCHLLIB“ auf Seite 84

MQCHLLIB gibt den Verzeichnispfad für die 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.

„MQCCDTURL“ auf Seite 82

MQCCDTURL bietet die funktional entsprechende Funktionalität zum Festlegen einer Kombination aus den Umgebungsvariablen MQCHLLIB und MQCHLTAB.

### Zugehörige Informationen

[IBM MQ-MQI-Anwendungen mit Warteschlangenmanagern verbinden](#)

## MQIPADDRV

MQIPADDRV gibt an, welches IP-Protokoll für eine Kanalverbindung verwendet werden soll. It has the possible string values of "MQIPADDR\_IPv4" or "MQIPADDR\_IPv6". These values have the same meanings as IPv4 and IPv6 in ALTER QMGR IPADDRV. Wenn diese Umgebungsvariable nicht gesetzt ist, wird der Wert "MQIPADDR\_IPv4" angenommen.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQIPADDRV=MQIPADDR_IPv4|MQIPADDR_IPv6
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQIPADDRV=MQIPADDR_IPv4|MQIPADDR_IPv6
```

- **IBM i** Unter IBM i:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPv4|MQIPADDR_IPv6)
```

## MQNAME

MQNAME gibt den lokalen NetBIOS-Namen an, den der IBM MQ-Prozess verwenden kann.

Eine vollständige Beschreibung und die Regeln der Vorrangstellung auf dem Client und dem Server finden Sie in „[NetBIOS-Verbindung in Windows definieren](#)“ auf Seite 223 .

Verwenden Sie den folgenden Befehl, um diese Variable zu definieren:

```
SET MQNAME=Your_env_Name
```

Beispiel:

```
SET MQNAME=CLIENT1
```

Für das NetBIOS auf einigen Plattformen ist für jede Anwendung ein anderer Name (festgelegt durch MQNAME) erforderlich, wenn Sie mehrere IBM MQ-Anwendungen gleichzeitig auf dem IBM MQ MQI client ausführen.

## MQSERVER

Die Umgebungsvariable MQSERVER wird verwendet, um einen minimalen Kanal zu definieren. MQSERVER gibt die Position des IBM MQ-Servers sowie das zu verwendende Kommunikationsverfahren an.

Sie können MQSERVER nicht verwenden, um einen TLS-Kanal oder einen Kanal mit Kanalexits zu definieren. Ausführliche Informationen zum Definieren eines TLS-Kanals finden Sie im Abschnitt [Kanäle mit TLS schützen](#).

*ConnectionName* muss ein vollständig qualifizierter Netzname sein. Das *ChannelName* kann den Schrägstrich (/) nicht enthalten, da dieses Zeichen zum Trennen des Kanalnamens, des Transporttyps und des Verbindungsnamens verwendet wird. 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.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

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

- **Linux** **UNIX** Unter UNIX and Linux:

```
export 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)')
```

- **z/OS** unter z/OS

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

*TransportType* kann abhängig von Ihrer IBM MQ-Clientplattform einen der folgenden Werte haben:

- LU62
- TCP
- NETBIOS
- SPX

*ConnectionName* kann eine durch Kommas getrennte Liste mit Verbindungsnamen sein. Die Verbindungsnamen in der Liste werden auf ähnliche Weise für mehrere Verbindungen in einer Clientverbindungstabelle verwendet. Die Liste *ConnectionName* kann als Alternative zu WS-Manager-Gruppen verwendet werden, um mehrere Verbindungen für den Client anzugeben, die versucht werden sollen. Wenn Sie

einen Warteschlangenmanager mit mehreren Instanzen konfigurieren, können Sie unter Umständen eine *ConnectionName* -Liste verwenden, um verschiedene Warteschlangenmanagerinstanzen anzugeben.

### **TCP/IP-Standardport**

Bei TCP/IP geht IBM MQ standardmäßig davon aus, dass der Kanal mit Port 1414 verbunden wird.

Dies kann wie folgt geändert werden:

- Die Anschlussnummer in Klammern als letzter Teil des Verbindungsnamens hinzufügen:

– **Windows** Unter Windows:

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

– **Linux** **UNIX** Unter UNIX and Linux:

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

- Ändern Sie die *mqclient.ini* -Datei, indem Sie die Portnummer zum Protokollnamen hinzufügen, z. B.:

```
TCP:  
port=2001
```

- Hinzufügen von IBM MQ zur Servicedatei wie in „[TCP/IP-Listener unter UNIX and Linux verwenden](#)“ auf Seite 228 beschrieben.

### **SPX-Standardsocket**

Für SPX geht IBM MQ standardmäßig davon aus, dass der Kanal mit dem Socket 5E86 verbunden wird.

Dies kann wie folgt geändert werden:

- Hinzufügen der Socketnummer in eckigen Klammern als letzter Teil des Verbindungsnamens:

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

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.

- Ändern Sie die *qm.ini* -Datei, indem Sie die Portnummer zum Protokollnamen hinzufügen, z. B.:

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

### **Verwenden von MQSERVER**

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 erfolgt kein Verweis auf die Clientkanaldefinitionstabelle (CCDT).

In dieser Situation bestimmt das Empfangsprogramm, das Sie auf der Servermaschine ausgeführt haben, den Warteschlangenmanager, zu dem Ihre Anwendung eine Verbindung herstellt. Es handelt sich um denselben Warteschlangenmanager, mit dem das Empfangsprogramm verbunden ist.

Wenn die *MQCONN*-oder *MQCONNX*-Anforderung einen anderen Warteschlangenmanager als den Listener angibt, mit dem das Empfangsprogramm verbunden ist, oder wenn der *MQSERVER*-Parameter *TransportType* nicht erkannt wird, schlägt die *MQCONN*-oder *MQCONNX*-Anforderung mit dem Rückkehrcode *MQRC\_Q\_MGR\_NAME\_ERROR* fehl.

**Linux** **UNIX** Auf UNIX and Linux-Systemen könnten 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 referenziert, die mit MQCONNX angegeben wurde. In diesem Fall hat der von der MQCD-Struktur angegebene Kanal Vorrang vor allen durch die Umgebungsvariable MQSERVER angegebenen Kanälen.

Die MQSERVER-Umgebungsvariable hat Vorrang vor jeder Clientkanaldefinition, auf die durch MQCHLLIB und MQCHLTAB verwiesen wird.

## MQSERVER wird abgebrochen

Geben Sie Folgendes ein, um MQSERVER abzubrechen und zur Clientkanaldefinitionstabelle zurückzukehren, auf die durch MQCHLLIB und MQCHLTAB verwiesen wird:

- **Windows** Unter Windows:

```
SET MQSERVER=
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
unset MQSERVER
```

## MQSSLCRYP

MQSSLCRYP enthält eine Parameterzeichenfolge, die es Ihnen ermöglicht, die auf dem System vorhandene Verschlüsselungshardware zu konfigurieren. Die zulässigen Werte entsprechen den Werten für den Parameter **SSLCRYP** des Befehls **ALTER QMGR**.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQSSLCRYP=string
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQSSLCRYP=string
```

### Zugehörige Informationen

Befehl **ALTER QMGR**, Parameter **SSLCRYP**

## MQSSLFIPS

MQSSLFIPS gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen, wenn in IBM MQ Verschlüsselung zum Einsatz kommt. Die Werte sind mit dem Wert für den Parameter SSLFIPS des Befehls ALTER QMGR identisch.

Die Verwendung von FIPS-zertifizierten Algorithmen ist von der Verwendung von Verschlüsselungshardware betroffen. Weitere Informationen finden Sie im Abschnitt [Angaben, dass nur FIPS-zertifizierte CipherSpecs zur Laufzeit auf dem MQI-Client verwendet werden](#).

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQSSLFIPS=YES|NO
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQSSLFIPS=YES|NO
```

- **IBM i** Unter IBM i:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

Der Standardwert ist NO.

## MQSSLKEYR

MQSSLKEYR gibt die Position des Schlüsselrepositorys an, in dem das digitale Zertifikat gespeichert ist, das dem Benutzer gehört, im Stammformat. Das Stammformat beinhaltet den vollständigen Pfad und den Dateinamen ohne Erweiterung.

Ausführliche Informationen hierzu finden Sie im Parameter [SSLKEYR](#) des Befehls ALTER QMGR.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQSSLKEYR=pathname
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQSSLKEYR=pathname
```

- **IBM i** Unter IBM i:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

Es gibt keinen Standardwert.

## MQSSLPROXY

MQSSLPROXY gibt den Hostnamen und die Portnummer des HTTP-Proxy-Servers an, der von GSKit für OCSP-Prüfungen verwendet werden soll.

Verwenden Sie einen der folgenden Befehle, um diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQSSLPROXY= string
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQSSLPROXY="string"
```

Die Zeichenfolge ist entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers, 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** **UNIX** Auf UNIX and Linux-Systemen können Sie zum Beispiel einen der folgenden Befehle verwenden:

- ```
export MQSSLPROXY="proxy.example.com(80)"
```
- ```
export MQSSLPROXY="127.0.0.1"
```

## MQSSLRESET

MQSSLRESET stellt die Anzahl der unverschlüsselten Byte dar, die auf einem TLS-Kanal gesendet und empfangen werden, bevor der geheime Schlüssel neu verhandelt wird.

Weitere Informationen zur Neuvereinbarung geheimer Schlüssel finden Sie im Abschnitt [Geheime TLS-Schlüssel zurücksetzen](#).

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 diese Variable zu definieren:

- **Windows** Unter Windows:

```
SET MQSSLRESET=integer
```

- **Linux** **UNIX** Unter UNIX and Linux:

```
export MQSSLRESET=integer
```

- **IBM i** Unter IBM i:

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

## **ULW** MQSUITEB

Sie können IBM MQ so konfigurieren, dass es in Übereinstimmung mit dem Standard NSA Suite B auf den Plattformen UNIX, Linux, and Windows ausgeführt wird.

Suite B beschränkt die Gruppe aktivierter Verschlüsselungsalgorithmen, um eine sichere Sicherheitsstufe zu gewährleisten.

Weitere Informationen finden Sie unter [IBM MQ für Suite B konfigurieren](#).

## MQTCPTIMEOUT

Gibt an, wie lange IBM MQ auf einen TCP-Verbindungsaufwurf wartet.

# Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

## Informationen zu diesem Vorgang

Sie können die Konfigurationsdaten von IBM MQ ändern, indem Sie die Werte bestimmter Konfigurationsattribute (oder -parameter) von IBM MQ ändern.

Sie ändern Attributinformationen, indem Sie die IBM MQ-Konfigurationsdateien bearbeiten. Sie können Konfigurationsdateien entweder automatisch bearbeiten, indem Sie Befehle verwenden, die die Konfiguration von Warteschlangenmanagern auf dem Knoten ändern, oder Sie können sie manuell mit einem Standardtexteditor ändern. Weitere Informationen finden Sie im Abschnitt [„Konfigurationsdateien bearbeiten“](#) auf Seite 95.

**Windows** **Linux** Auf Windows und Linux (x86 und x86-64-Plattformen), können Sie die IBM MQ-Konfigurationsdateien auch mit IBM MQ Explorer editieren.

**Windows** Auf Windows-Systemen können Sie auch **amqmdain** verwenden, um die Konfigurationsinformationen zu ändern (siehe [amqmdain](#)).

## Prozedur

- Weitere Informationen zum Konfigurieren von IBM MQ und Warteschlangenmanagern für Ihre Plattform finden Sie in den folgenden Unterabschnitten:

### Zugehörige Konzepte

„IBM MQ konfigurieren“ auf Seite 5

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.

### Zugehörige Tasks

„Warteschlangenmanager unter z/OS erstellen“ auf Seite 697

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

### Zugehörige Informationen

[Planung](#)

[IBM MQ verwalten](#)

## **ULW** Konfigurationsdaten unter UNIX, Linux, and Windows ändern

Unter UNIX, Linux, and Windows können Sie die IBM MQ-Konfigurationsattribute ändern, die in Konfigurationsdateien enthalten sind, auf der Ebene des Knotens und des Warteschlangenmanagers.

## Informationen zu diesem Vorgang

Auf UNIX, Linux, and Windows-Plattformen können Sie die IBM MQ-Konfigurationsattribute in den folgenden Dateien ändern:

- Eine IBM MQ-Konfigurationsdatei (`mqs.ini`), um Änderungen für IBM MQ auf dem Knoten als Ganzes zu bewirken. Für jeden Knoten gibt es eine `mqs.ini`-Datei. Weitere Informationen zu den Zeilengruppen in `mqs.ini` finden Sie in [„Attribute für die Änderung von IBM MQ-Konfigurationsinformationen“](#) auf Seite 115.
- Eine Warteschlangenmanagerkonfigurationsdatei (`qm.ini`), um Änderungen für bestimmte Warteschlangenmanager zu bewirken. Für jeden WS-Manager auf dem Knoten gibt es eine `qm.ini`-Datei.

Weitere Informationen zu den Zeilengruppen in `qm.ini` finden Sie unter [„Konfigurationsdaten des Warteschlangenmanagers ändern“](#) auf Seite 122.

Clientkonfigurationsoptionen werden separat in der Clientkonfigurationsdatei `mqclient.ini` gehalten. Weitere Informationen finden Sie unter [„Client mit einer Konfigurationsdatei konfigurieren“](#) auf Seite 56.

Eine Konfigurationsdatei (oder Zeilengruppendatei) enthält eine oder mehrere Zeilengruppen, die Gruppen von Zeilen in der Datei `.ini` sind, die zusammen eine allgemeine Funktion haben oder einen Teil eines Systems definieren, wie z. B. Protokollfunktionen, Kanalfunktionen und installierbare Services.

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.

**Wichtig:** Alle Änderungen, die Sie an einer Konfigurationsdatei vornehmen, werden in der Regel erst beim nächsten Start des Warteschlangenmanagers wirksam.

## Prozedur

- Bevor Sie eine Konfigurationsdatei editieren, sichern Sie sie so, dass Sie eine Kopie haben, auf die Sie zurücksetzen können, wenn die Notwendigkeit besteht.
- Editieren Sie die Konfigurationsdateien mit Hilfe von Befehlen oder einem Standardtexteditor. Weitere Informationen finden Sie unter [„Konfigurationsdateien bearbeiten“](#) auf Seite 95.
-  Auf Windows- und Linux-Systemen (x86- und x86-64-Systeme) verwenden Sie IBM MQ Explorer, um Änderungen an den Konfigurationsdateien vorzunehmen. Weitere Informationen finden Sie im Abschnitt [IBM MQ mit MQ Explorer konfigurieren](#).
-  Verwenden Sie auf Windows-Systemen als Alternative zu IBM MQ Explorer den Befehl `amqmdain`, um Änderungen an den Konfigurationsdateien vorzunehmen. Weitere Informationen finden Sie in [amqmdain](#).

## Zugehörige Konzepte

[„IBM MQ konfigurieren“](#) auf Seite 5

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.

## Zugehörige Tasks

[„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“](#) auf Seite 93

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

[„Konfigurationsdaten unter IBM i ändern“](#) auf Seite 102

Sie können das Verhalten von Warteschlangenmanagern an die Anforderungen Ihrer Installation anpassen, indem Sie die Werte ändern, die in einer Gruppe von Konfigurationsattributen (oder Parametern) angegeben sind, die IBM MQ steuern.

[„Konfigurationsdaten des Warteschlangenmanagers ändern“](#) auf Seite 122

Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

## Zugehörige Verweise

[„Attribute für die Änderung von IBM MQ-Konfigurationsinformationen“](#) auf Seite 115

Ändern Sie auf IBM MQ für Windows-Systemen und auf IBM MQ für Linux-Systemen (x86- und x86-64-Plattformen) die Konfigurationsdaten mit IBM MQ Explorer. Auf anderen Systemen ändern Sie die Informationen, indem Sie die Konfigurationsdatei `'mqs.ini'` bearbeiten.

## Zugehörige Informationen

Planung

IBM MQ verwalten

### Konfigurationsdateien bearbeiten

Bearbeiten Sie die Konfigurationsdateien mit Hilfe von Befehlen oder einem Standardtexteditor.

Bevor Sie eine Konfigurationsdatei editieren, sichern Sie sie so, dass Sie eine Kopie haben, auf die Sie zurückgreifen können, wenn die Notwendigkeit besteht.

Sie können die Konfigurationsdateien entweder wie folgt bearbeiten:

- Automatisch unter Verwendung von Befehlen, die die Konfiguration von Warteschlangenmanagern auf dem Knoten ändern
- Manuell unter Verwendung eines Standardtexteditors

Sie können die Standardwerte in den Konfigurationsdateien von IBM MQ nach der Installation bearbeiten.

Wenn Sie einen falschen Wert in einem Konfigurationsdateiattribut festlegen, wird der Wert ignoriert und eine Bedienernachricht ausgegeben, um das Problem anzugeben. (Der Effekt ist der gleiche wie das Fehlen des Attributs vollständig.)

Wenn Sie einen neuen WS-Manager erstellen, gilt Folgendes:

- Sichern Sie die IBM MQ-Konfigurationsdatei
- Sichern Sie die neue WS-Manager-Konfigurationsdatei.

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 ein "\"-Zeichen als Präfix voranstellen und es wird als Teil der Konfigurationsdaten verwendet.

### Wann müssen Sie eine Konfigurationsdatei bearbeiten?

Bearbeiten Sie eine Konfigurationsdatei, um die Sicherung wiederherzustellen, einen WS-Manager zu verschieben, den Standardwarteschlangenmanager zu ändern oder den IBM Support zu unterstützen.

Möglicherweise müssen Sie eine Konfigurationsdatei bearbeiten, wenn Sie zum Beispiel:

- Sie verlieren eine Konfigurationsdatei. (Wiederherstellung von der Sicherung, wenn Sie können.)
- Sie müssen einen oder mehrere WS-Manager in ein neues Verzeichnis verschieben.
- Sie müssen den Standardwarteschlangenmanager ändern. Dies kann passieren, wenn Sie versehentlich den vorhandenen WS-Manager löschen.
- Es wird empfohlen, dies über Ihr IBM Support Center zu tun.

### Konfigurationsdateiprioritäten

Der Wert eines Attributs wird an mehreren Stellen definiert. Attribute, die in Befehlen festgelegt werden, haben Vorrang vor Attributen in Konfigurationsdateien.

Die Attributwerte einer Konfigurationsdatei werden entsprechend den folgenden Prioritäten festgelegt:

- Die in der Befehlszeile eingegebenen Parameter haben Vorrang vor den Werten, die in den Konfigurationsdateien definiert sind.
- Werte, die in den qm.ini-Dateien definiert sind, haben Vorrang vor den Werten, die in der Datei mq5.ini definiert sind.

### Kommentare in den Konfigurationsdateien



Sie können das Semikolon ';' und den Hashwert '#' 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 für einen Wert eines dieser Zeichen erforderlich ist, müssen Sie dieses Zeichen mit dem Backslash-Zeichen '\' verlassen.

## IBM MQ-Konfigurationsdatei, mqs.ini

Die IBM MQ-Konfigurationsdatei `mqs.ini` enthält Informationen, die für alle WS-Manager auf dem Knoten relevant sind. Sie wird automatisch während der Installation erstellt.

### Verzeichnispositionen

  Unter UNIX und Linux lauten das Daten- und das Protokollverzeichnis immer `/var/mqm` bzw. `/var/mqm/log`.

 Auf Windows-Systemen werden der Speicherort des Datenverzeichnisses `mqs.ini` und der Speicherort des Protokollverzeichnisses in der Registry gespeichert, da sie wechselnde Speicherorte haben können. Die Informationen zur Installationskonfiguration, die in `mqinst.ini` auf UNIX- und Linux-Systemen enthalten sind, befinden sich ebenfalls in der Registry, da es unter Windows keine `mqinst.ini`-Datei gibt (siehe „Installationskonfigurationsdatei, `mqinst.ini`“ auf Seite 101).

 Die Datei `mqs.ini` für Windows-Systeme wird durch den Arbeitspfad angegeben, der im Schlüssel `HKLM\SOFTWARE\IBM\WebSphere MQ` angegeben ist. Sie enthält:

- Die Namen der WS-Manager
- Der Name des Standard-WS-Managers.
- Die Position der Dateien, die jedem von ihnen zugeordnet sind.

### Zeilengruppe LogDefaults für eine neue Installation

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 werden für die Zeilengruppe `LogDefaults` in [Abbildung 5 auf Seite 97](#) angezeigt. Der Wert null für das Attribut `LogBufferPages` bedeutet 512.

Wenn Sie einen anderen Wert als den Standardwert benötigen, müssen Sie diesen Wert explizit in der Zeilengruppe `LogDefaults` angeben.

### Beispiel für eine `mqs.ini` -Datei



```

#*****#
#* 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

```

Abbildung 5. Beispiel für eine IBM MQ-Konfigurationsdatei für UNIX

ULW

## Warteschlangenmanagerkonfigurationsdateien, qm.ini

Eine Warteschlangenmanagerkonfigurationsdatei `qm.ini` enthält Informationen, die für einen bestimmten Warteschlangenmanager relevant sind.

Für jeden Warteschlangenmanager gibt es eine WS-Manager-Konfigurationsdatei. Die `qm.ini`-Datei wird automatisch erstellt, wenn der Warteschlangenmanager, dem sie zugeordnet ist, erstellt wird.

**Anmerkung:** Informationen darüber, wann Änderungen, die Sie an der Datei `qm.ini` vornehmen, wirksam werden, finden Sie unter „Konfigurationsdaten unter UNIX, Linux, and Windows ändern“ auf Seite 93.

V 9.0.4

► V 9.0.0.2

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** **UNIX** Auf UNIX and Linux-Systemen befindet sich eine Datei `qm.ini` im Stammverzeichnis der Verzeichnisstruktur, 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
```

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 im Abschnitt [Informationen zu IBM MQ-Dateinamen](#).

## Beispiel für eine `qm.ini`-Datei

**Linux** **UNIX**

Das folgende Beispiel zeigt, wie Gruppen von Attributen in einer Warteschlangenmanagerkonfigurationsdatei auf IBM MQ for UNIX -und Linux -Systemen angeordnet werden können.

**V 9.0.5** Dies ist eine Beispielkonfigurationsdatei für einen Warteschlangenmanager ab IBM MQ 9.0.5:

```
##* 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 ##*
##*           :                                         ##*
##*           :                                         ##*
ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

Service:
  Name=AuthorizationService
  EntryPoints=14

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0 1
  LogPath=/var/mqm/log/saturn!queue!manager/

XAResourceManager:
```

```
Name=DB2 Resource Manager Bank
SwitchFile=/usr/bin/db2swit
XAOpenString=MQBankDB
XACloseString=
ThreadOfControl=THREAD
```

```
Channels: 2
MaxChannels=200
MaxActiveChannels=100
MQIBindType=STANDARD
```

```
TCP:
SndBuffSize=0
RcvBuffSize=0
RcvSndBuffSize=0
RcvRcvBuffSize=0
ClntSndBuffSize=0
ClntRcvBuffSize=0
SvrSndBuffSize=0
SvrRcvBuffSize=0
```

```
QMErrorLog:
ErrorLogSize=262144
ExcludeMessage=7234
SuppressMessage=9001,9002,9202
SuppressInterval=30
```

```
ApiExitLocal:
Name=ClientApplicationAPIchecker
Sequence=3
Function=EntryPoint
Module=/usr/Dev/ClientAppChecker
Data=9.20.176.20
```

```
TuningParameters:
ImplSyncOpenOutput=2
```

Dies ist eine Beispielkonfigurationsdatei für Versionen bis einschließlich IBM MQ 9.0.4 sowie für LTS:

```
## 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 ##
##
##*****#
```

```
ExitPath:
ExitsDefaultPath=/var/mqm/exits
ExitsDefaultPath64=/var/mqm/exits64
```

```
Service:
Name=AuthorizationService
EntryPoints=14
```

```
ServiceComponent:
Service=AuthorizationService
Name=MQSeries.UNIX.auth.service
Module=amqzfu
ComponentDataSize=0
```

```
Log:
LogPrimaryFiles=3
LogSecondaryFiles=2
LogFilePages=4096
LogType=CIRCULAR
LogBufferPages=0 1
LogPath=/var/mqm/log/saturn!queue!manager/
```

```
XAResourceManager:
Name=DB2 Resource Manager Bank
SwitchFile=/usr/bin/db2swit
XAOpenString=MQBankDB
XACloseString=
ThreadOfControl=THREAD
```

```

Channels: 2
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD

TCP:
  SndBuffSize=0
  RcvBuffSize=0
  RcvSndBuffSize=0
  RcvRcvBuffSize=0
  ClntSndBuffSize=0
  ClntRcvBuffSize=0
  SvrSndBuffSize=0
  SvrRcvBuffSize=0

QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30

ApiExitLocal:
  Name=ClientApplicationAPIchecker
  Sequence=3
  Function=EntryPoint
  Module=/usr/Dev/ClientAppChecker
  Data=9.20.176.20

```

### Anmerkungen:

1. Der Wert 0 für LogBufferPages ergibt einen Wert von 512.
2. Weitere Informationen zur Zeilengruppe des 'Channel' finden Sie im Abschnitt „[Initialisierungs- und Konfigurationsdateien](#)“ auf Seite 204.
3. 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.
4.  Ab IBM MQ 8.0 werden für neue Warteschlangenmanager unter 'Multiplatforms' die Standardgrößen für TCP-Sende- und -Empfangspuffer in der TCP-Zeilengruppe des `qm.ini` files festgelegt, die vom Betriebssystem verwaltet werden. Neue Warteschlangenmanager werden automatisch mit den folgenden Standardeinstellungen für die Sendepuffer erstellt:

```

TCP:
  SndBuffSize=0
  RcvBuffSize=0
  RcvSndBuffSize=0
  RcvRcvBuffSize=0
  ClntSndBuffSize=0
  ClntRcvBuffSize=0
  SvrSndBuffSize=0
  SvrRcvBuffSize=0

```

Vor IBM MQ 8.0 war die Standardeinstellung für diese Werte ohne manuelle Optimierung ein 32 KB-Puffer fester Größe.

Diese Änderung gilt nur für neue Warteschlangenmanager. Die Einstellungen der TCP-Sende- und -Empfangspuffer für Warteschlangenmanager, die aus früheren Versionen migriert werden, bleiben erhalten.

Informationen, wie Sie die TCP-Puffergrößen manuell festlegen, damit das Standardverhalten des Betriebssystems für migrierte Warteschlangenmanager gilt, finden Sie im Abschnitt [TCP, LU62, NETBIOS und SPX](#).

### Zeilengruppe AccessMode

 Windows

Die Datei `qm.ini` für Windows enthält eine zusätzliche Zeilengruppe `AccessMode`:

```
AccessMode:  
SecurityGroup=wmq\wmq
```

## Zeilengruppe `APIExitLocal`

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 `_r` an den bereitgestellten Modulnamen anhängt, um die Threadversion des Exits zu identifizieren, aber IBM MQ keinen direkt äquivalenten Mechanismus für die 32 -Bit- und 64-Bit-Varianten bereitstellt.

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
```

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.

### Zugehörige Verweise

„TCP, LU62 und NETBIOS“ auf Seite 134

Verwenden Sie diese WS-Manager-Eigenschaftenseiten oder Zeilengruppen in der Datei `qm.ini`, um Netzprotokollkonfigurationsparameter anzugeben. Sie überschreiben die Standardattribute für Kanäle.

## Installationskonfigurationsdatei, `mqinst.ini`

Unter UNIX oder Linux enthält die Installationskonfigurationsdatei `mqinst.ini` Informationen zu allen Installationen von IBM MQ. Unter Windows befinden sich die Installationskonfigurationsdaten in der Registry.

### Position der `mqinst.ini`-Datei

Die Datei `mqinst.ini` befindet sich auf UNIX and Linux-Systemen im Verzeichnis `/etc/opt/mqm`. 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 fixiert 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.
- [dlmqinst](#) 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 **dlmqinst** nicht zur Verfügung. Die Installations- und Deinstallationsprozesse bearbeiten die Erstellung und das Löschen der erforderlichen Registry-Einträge.

### Windows

## IBM i Konfigurationsdaten unter IBM i ändern

Sie können das Verhalten von Warteschlangenmanagern an die Anforderungen Ihrer Installation anpassen, indem Sie die Werte ändern, die in einer Gruppe von Konfigurationsattributen (oder Parametern) angegeben sind, die IBM MQ steuern.

### Informationen zu diesem Vorgang

Sie ändern die Konfigurationsattribute, indem Sie die IBM MQ-Konfigurationsdateien bearbeiten.

### Prozedur

- Weitere Informationen zum Ändern der Konfigurationswerte unter IBM i finden Sie in folgenden Abschnitten:
  - [„IBM MQ-Konfigurationsdateien für IBM i“ auf Seite 103](#)
  - [„Attribute zum Ändern von Konfigurationsinformationen unter IBM i“ auf Seite 105](#)
  - [„WS-Manager-Konfigurationsdaten unter IBM i ändern“ auf Seite 107](#)
  - [„Beispieldateien mqz.ini und qm.ini für IBM i“ auf Seite 113](#)

### Zugehörige Konzepte

[„IBM MQ konfigurieren“ auf Seite 5](#)

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.

## Zugehörige Tasks

„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93  
Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

„Konfigurationsdaten unter UNIX, Linux, and Windows ändern“ auf Seite 93  
Unter UNIX, Linux, and Windows können Sie die IBM MQ-Konfigurationsattribute ändern, die in Konfigurationsdateien enthalten sind, auf der Ebene des Knotens und des Warteschlangenmanagers.

„Konfigurationsdaten des Warteschlangenmanagers ändern“ auf Seite 122  
Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

## Zugehörige Verweise

„Attribute für die Änderung von IBM MQ-Konfigurationsinformationen“ auf Seite 115  
Ändern Sie auf IBM MQ for Windows -Systemen und auf IBM MQ for Linux -Systemen (x86 -und x86-64 -Plattformen) die Konfigurationsdaten mit IBM MQ Explorer. Auf anderen Systemen ändern Sie die Informationen, indem Sie die Konfigurationsdatei 'mqs.ini' bearbeiten.

## Zugehörige Informationen

[Planung](#)  
[IBM MQ verwalten](#)

## IBM MQ-Konfigurationsdateien für IBM i

Dieser Abschnitt enthält Erläuterungen der Verfahren beim Konfigurieren von IBM MQ for IBM i.

Unter IBM i ändern Sie die IBM MQ-Konfigurationsattribute in:

- Eine IBM MQ-Konfigurationsdatei, `mqs.ini`, bewirkt Änderungen auf dem Knoten als Ganzes. Für jede Installation von IBM MQ gibt es eine `mqs.ini`-Datei.
- Eine WS-Manager-Konfigurationsdatei, `qm.ini`, wirkt sich auf bestimmte WS-Manager aus. Für jeden WS-Manager auf dem Knoten gibt es eine `qm.ini`-Datei.

Beachten Sie, dass INI-Dateien Datenstromdateien sind, die sich im IFS befinden.

Eine Konfigurationsdatei (diese kann auch als **Zeilengruppendatei** bezeichnet werden) enthält mindestens eine Zeilengruppe. Dabei handelt es sich um Gruppen von Zeilen in der Datei '.ini', die alle eine Funktion gemein haben oder einen Systembereich definieren, beispielsweise Protokoll- und Kanalfunktionen. Alle Änderungen, die an einer Konfigurationsdatei vorgenommen werden, werden erst beim nächsten Start des Warteschlangenmanagers wirksam.

## Konfigurationsdateien bearbeiten

Bevor Sie eine Konfigurationsdatei editieren, sichern Sie sie so, dass Sie eine Kopie haben, auf die Sie zurückgreifen können, wenn die Notwendigkeit besteht.

Sie können die Konfigurationsdateien entweder wie folgt bearbeiten:

- Automatisch mithilfe von Befehlen, die die Konfiguration von Warteschlangenmanagern auf dem Knoten ändern
- Manuell mit dem CL-Editor EDTF.

Sie können die Standardwerte in den Konfigurationsdateien von IBM MQ nach der Installation bearbeiten. Wenn Sie einen falschen Wert in einem Konfigurationsdateiattribut festlegen, wird der Wert ignoriert und eine Bedienernachricht ausgegeben, um das Problem anzugeben. (Der Effekt ist der gleiche wie das Fehlen des Attributs vollständig.)

Wenn Sie einen neuen WS-Manager erstellen, gilt Folgendes:

- Sichern Sie die IBM MQ-Konfigurationsdatei.
- Sichern Sie die neue Warteschlangenmanagerkonfigurationsdatei.

## Wann müssen Sie eine Konfigurationsdatei bearbeiten?

Möglicherweise müssen Sie eine Konfigurationsdatei bearbeiten, wenn Sie zum Beispiel:

- Sie verlieren eine Konfigurationsdatei. Wenn möglich, stellen Sie eine Wiederherstellung nach einem Backup wieder her.
- Sie müssen einen oder mehrere WS-Manager in ein neues Verzeichnis verschieben.
- Sie müssen den Standardwarteschlangenmanager ändern. Dies kann passieren, wenn Sie versehentlich den vorhandenen WS-Manager löschen.
- Es wird empfohlen, dies über Ihr IBM Support Center zu tun.

## Konfigurationsdateiprioritäten

Die Attributwerte einer Konfigurationsdatei werden entsprechend den folgenden Prioritäten festgelegt:

- Die in der Befehlszeile eingegebenen Parameter haben Vorrang vor den Werten, die in den Konfigurationsdateien definiert sind.
- Werte, die in den `qm.ini`-Dateien definiert sind, haben Vorrang vor Werten, die in der `mqs.ini`-Datei definiert sind.

## Die IBM MQ-Konfigurationsdatei `mqs.ini`

Die IBM MQ-Konfigurationsdatei, `mqs.ini`, enthält Informationen, die für alle Warteschlangenmanager in einer IBM MQ-Installation relevant sind. Sie wird automatisch während der Installation erstellt. Insbesondere wird die Datei `mqs.ini` verwendet, um die Daten zu lokalisieren, die den einzelnen Warteschlangenmanagern zugeordnet sind.

Die Datei `mqs.ini` ist im Verzeichnis `/QIBM/UserData/mqm` gespeichert

Die Datei `mqs.ini` 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](#)).

## Warteschlangenmanagerkonfigurationsdateien, `qm.ini`

Eine Warteschlangenmanagerkonfigurationsdatei `qm.ini` enthält Informationen, die für einen bestimmten Warteschlangenmanager relevant sind. Für jeden Warteschlangenmanager gibt es eine WS-Manager-Konfigurationsdatei. Die `qm.ini`-Datei wird automatisch erstellt, wenn der Warteschlangenmanager, dem sie zugeordnet ist, erstellt wird.

Eine `qm.ini`-Datei befindet sich im `mqmdata directory/QMNAME/qm.ini`, wobei `mqmdata directory` standardmäßig `/QIBM/UserData/mqm` und `QMNAME` der Name des Warteschlangenmanagers ist, auf den die Initialisierungsdatei angewendet wird.

### Anmerkung:

1. Sie können den `mqmdata directory` in der Datei `mqs.ini` ändern.
2. 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. Weitere Informationen hierzu finden Sie im Abschnitt [Informationen zu den Bibliotheksnamen der IBM MQ for IBM i-Warteschlangenmanager](#).

Verwenden Sie diese Informationen, um die Zeilengruppen für Konfigurationsinformationen zu verstehen.

Die folgenden Attributgruppen treten in `mqs.ini` auf:

- „Die Zeilengruppe 'AllQueueManagers'“ auf Seite 105
- „Zeilengruppe 'DefaultQueueManager'“ auf Seite 106
- „Die ExitProperties-Zeilengruppe“ auf Seite 106
- „Zeilengruppe 'QueueManager'“ auf Seite 107

Es gibt auch zwei Zeilengruppen, die API-Exits zugeordnet sind: `ApiExitCommon` und `ApiExitTemplate`. Weitere Informationen zur Verwendung dieser Informationen finden Sie im Abschnitt [API-Exits konfigurieren](#).

## Die Zeilengruppe 'AllQueueManagers'

Die Zeilengruppe `AllQueueManagers` kann Folgendes angeben:

- Der Pfad zu dem Verzeichnis `qmgrs`, in dem die Dateien gespeichert werden, die einem Warteschlangenmanager zugeordnet sind.
- Der Pfad zur ausführbaren Bibliothek
- Die Methode zum Konvertieren von EBCDIC-Format-Daten in das ASCII-Format

In den Beschreibungen der Zeilengruppen ist der unterstrichene Wert der Standardwert und das Symbol | bedeutet *oder*.

### **DefaultPrefix= directory\_name**

Der Pfad zum Verzeichnis `qmgrs`, in dem die WS-Manager-Daten gespeichert werden. Wenn Sie das Standardpräfix für den WS-Manager ändern, müssen Sie die Verzeichnisstruktur replizieren, die während der Installation erstellt wurde. Insbesondere müssen Sie die `qmgrs`-Struktur erstellen. Stoppen Sie IBM MQ, bevor Sie das Standardpräfix ändern, und starten Sie IBM MQ erst dann erneut, nachdem Sie die Strukturen an die neue Position verschoben und das Standardpräfix geändert haben.

Anstatt das Standardpräfix zu ändern, können Sie die Umgebungsvariable `MQSPREFIX` nutzen, um das Standardpräfix `DefaultPrefix` für den Befehl `crtmqm` zu überschreiben.

### **ConvEBCDICNewline = NL\_TO\_LF | TABLE | ISO**

EBCDIC-Codepages enthalten ein Zeichen `newline` (NL), das von ASCII-Codepages nicht unterstützt wird, obwohl einige ISO-Varianten von ASCII ein Äquivalent enthalten.

Verwenden Sie das Attribut "ConvEBCDICNewline", um die Methode anzugeben, die von IBM MQ bei der Konvertierung des EBCDIC-NL-Zeichens in das ASCII-Format verwendet werden soll.

#### **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 unter IBM i für alle EBCDIC-zu-ASCII-Konvertierungen verwendet werden.

Beachten Sie, dass die Auswirkungen dieser Art der Konvertierung von Sprache zu Sprache variieren können.

#### **ISO**

Geben Sie ISO an, wenn Sie möchten:

- ISO-CCSIDs, die mit der TABLE-Methode konvertiert werden sollen
- Alle anderen CCSIDs, die mit der Methode `NL_TO_CF` konvertiert werden sollen.

Die möglichen ISO-CCSIDs werden im Abschnitt [Tabelle 10 auf Seite 106](#) aufgeführt.

<i>Tabelle 10. Liste der möglichen ISO-CCSIDs</i>	
<b>CCSID</b>	<b>Codetruppe</b>
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 kein ISO-Subset ist, nimmt ConvEBCDICNewline standardmäßig NL\_TO\_LF an.

### **Zeilengruppe 'DefaultQueueManager'**

Die Zeilengruppe `DefaultQueueManager` gibt den Standardwarteschlangenmanager für den Knoten an.

#### **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 auf das Original zurücksetzen möchten, müssen Sie das Attribut `DefaultQueueManager` manuell ändern.

### **Die ExitProperties-Zeilengruppe**

Die Zeilengruppe `ExitProperties` gibt Konfigurationsoptionen an, die von Exitprogrammen des Warteschlangenmanagers verwendet werden.

In den Beschreibungen der Zeilengruppen ist der unterstrichene Wert der Standardwert und das Symbol | bedeutet *oder*.

#### **CLWLMode= SAFE | FAST**

Mit dem Exit für Clusterauslastung (CLWL) können Sie angeben, welche Clusterwarteschlange in dem Cluster als Antwort auf einen MQI-Aufruf geöffnet werden soll (z. B. MQOPEN oder MQPUT). Der CLWL-Exit wird entweder im FAST-Modus oder im SAFE-Modus ausgeführt, abhängig von dem Wert, den Sie im Attribut `CLWLMode` angeben. 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 an den 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 (amqzlw0) 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 fehlerhaften Rückkehrcodes.

Die Integrität des Warteschlangenmanagers bleibt erhalten.

**Anmerkung:** Die Ausführung des CLWL-Exits in einem separaten Prozess kann sich nachteilig 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 dem Betrieb im SAFE-Modus zugeordneten Overheads vermieden werden, dies jedoch zu Lasten der Integrität des Warteschlangenmanagers geht. Führen Sie den CLWL-Exit im FAST-Modus nur aus, wenn Sie davon überzeugt sind, dass es **keine** Probleme mit dem CLWL-Exit gibt, und Sie sind besonders besorgt über Leistungsüberstände.

Tritt ein Problem auf, wenn der CLWL-Exit im FAST-Modus ausgeführt wird, schlägt der Warteschlangenmanager fehl, und Sie laufen Gefahr, die Integrität des Warteschlangenmanagers zu gefährden.

## **Zeilengruppe 'QueueManager'**

Für jeden WS-Manager gibt es eine Zeilengruppe `QueueManager`. Diese Attribute geben den Namen des Warteschlangenmanagers und den Namen des Verzeichnisses an, das die Dateien enthält, die diesem WS-Manager 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 im Abschnitt [Informationen zu den Bibliotheksnamen des IBM MQ for IBM i-Warteschlangenmanagers](#).

#### **Name = `queue_manager_name`**

Der Name des Warteschlangenmanagers.

#### **Präfix = `prefix`**

Gibt an, wo die Warteschlangenmanagerdateien gespeichert sind. Standardmäßig ist dies mit dem Wert identisch, der im Attribut `DefaultPrefix` der Zeilengruppe `AllQueueManager` in der Datei `mq5.ini` angegeben ist.

#### **Verzeichnis = `name`**

Der Name des Unterverzeichnisses unter dem Verzeichnis `prefix\QMGRS`, in dem die WS-Manager-Dateien 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 Dateiname ist.

#### **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.

## **IBM i WS-Manager-Konfigurationsdaten unter IBM i ändern**

Verwenden Sie diese Informationen, um die Konfigurationszeilengruppen des Warteschlangenmanagers zu verstehen.

Es gibt zwei Zeilengruppen, die API-Exits zugeordnet sind, `ApiExitCommon` und `ApiExitTemplate`. Ausführliche Informationen zur Verwendung dieser Zeilengruppen finden Sie im Abschnitt [API-Exits konfigurieren](#).

Die folgenden Attributgruppen können in einer `qm.ini`-Datei für einen bestimmten Warteschlangenmanager auftreten oder verwendet werden, um die in `mq5.ini` festgelegten Werte außer Kraft zu setzen.

In den folgenden Abschnitten finden Sie Informationen zum Ändern von Konfigurationsinformationen für bestimmte Optionen:

- [„Die Zeilengruppe 'Log' unter IBM i“ auf Seite 108](#)
- [„Die Zeilengruppe 'Channels' unter IBM i“ auf Seite 108](#)

- „QMErrorLog-Zeilengruppe unter IBM i“ auf Seite 110
- „Die Zeilengruppe 'TCP' unter IBM i“ auf Seite 111
- „Zeilengruppe für PreConnect der Clientkonfigurationsdatei“ auf Seite 75

## IBM i Die Zeilengruppe 'Log' unter IBM i

Parameter für die Konfiguration der Protokolldatei.

Die Zeilengruppe Log gibt die Protokollattribute für einen bestimmten Warteschlangenmanager an. Diese Attribute werden standardmäßig von den Einstellungen übernommen, die in der Zeilengruppe LogDefaults in der Datei mqz.ini angegeben sind, wenn der Warteschlangenmanager erstellt wird.

Ändern Sie die Attribute dieser Zeilengruppe nur, wenn Sie einen WS-Manager anders als andere konfigurieren möchten.

Die Werte, die in den Attributen in der Datei qm.ini angegeben sind, werden beim Starten des Warteschlangenmanagers gelesen. Die Datei wird beim Erstellen des Warteschlangenmanagers erstellt.

### LogBuffergröße

Die Größe des Journalpuffers in Byte. Geben Sie eine Zahl im Bereich von 32 000-15 761 440 ein. Der Standardwert ist 32 000.

### LogPath= library\_name

Der Name der Bibliothek, die zum Speichern von Journalen und Journalempfängern für diesen WS-Manager verwendet wird.

### LogReceiverSize

Die Größe des Journalempfängers in Kilobyte. Der Standardwert ist 100 000.

## IBM i Die Zeilengruppe 'Channels' unter IBM i

Die Zeilengruppe Channels enthält Informationen zu den Kanälen.

### MaxChannels= 100 | number

Die maximale Anzahl zulässiger *aktueller* Kanäle. Für z/OS muss der Wert zwischen 1 und 9999 liegen, wobei der Standardwert 200 ist. Alle anderen Plattformen haben den Standardwert 100.

### MaxActiveChannels= MaxChannels\_value

Die maximale Anzahl der Kanäle, die zu einem beliebigen Zeitpunkt *aktiv* sein dürfen. Der Standardwert ist der Wert, der im Attribut MaxChannels angegeben ist.

### MaxInitiators= 3 | 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. Das heißt, es gibt keinen Agentenprozess.

#### STANDARD

Kanäle verbinden sich mit STANDARD.

### ThreadedListener= NO |YES

Gibt an, ob RUNMQLSR ( YES ) oder AMQCLMAA ( NO ) als Listener gestartet werden soll.

Wenn Sie ThreadedListener = YES angeben, werden alle Kanäle als Threads in einem einzigen Job ausgeführt. Dadurch wird die Anzahl der Verbindungen zu den Ressourcen begrenzt, die für einen einzelnen Job verfügbar sind.

Wenn Sie ThreadedListener = NO angeben, startet der Nicht-Thread-Listener (AMQCLMAA) einen neuen Responder-Job (AMQCRSTA) für jeden eingehenden TCP/IP-Kanal. Der Nachteil dieses Verfahrens besteht darin, dass es nicht so schnell ist, einen neuen AMQCRSTA-Job zu starten, da ein Thread innerhalb eines RUNMQLSR-Jobs gestartet werden soll. Daher sind Verbindungszeiten für einen Nicht-Thread-Listener langsamer als die Verbindungszeiten für einen Thread-Listener.

## **AdoptNewMCA= NO | SVR | SNDR | RCVR | CLUSRCVR | ALL | FASTPATH**

Wenn IBM MQ eine Anforderung zum Starten eines Kanals empfängt, aber feststellt, dass ein amqcrsta-Prozess für den gleichen Kanal vorhanden ist, muss der vorhandene Prozess gestoppt werden, bevor der neue Prozess gestartet werden kann. Mit dem Attribut AdoptNewMCA können Sie die Beendigung eines vorhandenen Prozesses und den Start eines neuen Prozesses für einen bestimmten Kanaltyp steuern.

Wenn Sie das Attribut AdoptNewMCA für einen bestimmten Kanaltyp angeben, aber der neue Kanal nicht gestartet werden kann, weil der Kanal bereits aktiv ist:

1. Der neue Kanal versucht, die vorherige zu beenden.
2. Wenn der vorherige Kanalserver nicht beendet wird, wenn das Warteintervall für AdoptNewMCATimeout abläuft, wird der Prozess (oder der Thread) 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.

Sie geben 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**

Serverkanäle verwalten

### **SNDR**

Senderkanäle anpassen

### **RCVR**

Empfängerkanäle verwalten

### **CLUSRCVR**

Clusterempfängerkanäle verwalten

### **ALLE**

Alle Kanaltypen mit Ausnahme von FASTPATH-Kanälen anpassen

### **FASTPATH**

Geben Sie den Kanal an, wenn es sich um einen FASTPATH-Kanal handelt. Dies geschieht nur, wenn der entsprechende Kanaltyp ebenfalls angegeben ist, z. B. AdoptNewMCA = RCVR, SVR, FASTPATH

**Achtung!:** Das Attribut "AdoptNewMCA" kann sich aufgrund des internen Designs des Warteschlangenmanagers in unvorhersehbarer Weise mit FASTPATH-Kanälen verhalten. Gehen Sie mit großer Vorsicht vor, wenn Sie das Attribut AdoptNewMCA für FASTPATH-Kanäle aktivieren.

## **AdoptNewMCATimeout= 60 | 1-3600**

Die Zeit in Sekunden, die der neue Prozess wartet, bis der alte Prozess beendet ist. Geben Sie einen Wert in Sekunden im Bereich von 1 bis 3600 an. Der Standardwert ist 60.

## **AdoptNewMCACheck = QM | ADRESSE | NAME | ALL**

Mit dem Attribut AdoptNewMCACheck können Sie die Typüberprüfung angeben, die für die Aktivierung des Attributs AdoptNewMCA erforderlich ist. Es ist wichtig, dass Sie alle drei der folgenden Prüfungen durchführen, wenn möglich, um Ihre Kanäle vor dem Herunterfahren, versehentlich oder böswillig zu schützen. Überprüfen Sie zumindest, ob die Kanalnamen übereinstimmen.

Einen oder mehrere Werte angeben, die durch Kommas oder Leerzeichen voneinander getrennt sind, aus den folgenden Angaben:

### **QM**

Der Empfangsprogrammprozess prüft, ob die Namen des Warteschlangenmanagers übereinstimmen.

### **ADDRESS**

Der Listenerprozess überprüft die Kommunikationsadresse, z. B. die TCP/IP-Adresse.

## NAME

Der Listenerprozess überprüft, ob die Kanalnamen übereinstimmen.

## ALLE

Der Listenerprozess prüft, wie übereinstimmende WS-Manager-Namen, die Kommunikationsadresse und die übereinstimmenden Kanalnamen vorhanden sind.

Der Standardwert ist `AdoptNewMCACheck=NAME , ADDRESS , QM.`

## Zugehörige Konzepte

„Kanalstatus“ auf Seite 186

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.

## IBM i

### QMErrorLog-Zeilengruppe unter IBM i

Verwenden Sie die Zeilengruppe `QMErrorLog` in der Datei `qm.ini`, um die Operation und den Inhalt der Fehlerprotokolle des Warteschlangenmanagers anzupassen.

## V 9.0.4

### 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.

**Wichtig:** Ab IBM MQ 9.0.4 wurde die Standardgröße des Attributs **ErrorLogSize** erhöht. Dies ist eine Änderung gegenüber IBM MQ 9.0.3.

Sie können die Größe des Protokolls mithilfe der Umgebungsvariablen `MQMAXERRORLOGSIZE` festlegen.

### ExcludeMessage= msgIds

Gibt Nachrichten an, die nicht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben werden sollen. `msgIds` enthält eine durch Kommas getrennte Liste mit Nachrichten-IDs aus den folgenden:

- 7163 - Nachricht 'Job gestartet (nur IBM i)
- 7234-Anzahl der geladenen Nachrichten
- 8245
- 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
- 9508-Verbindung zum WS-Manager kann nicht hergestellt werden
- 9524-Ferner WS-Manager nicht verfügbar
- 9528-Benutzer hat das Schließen des Kanals angefordert
- 9558-Ferner Kanal ist nicht verfügbar
- 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. Das Zeitintervall wird durch **SuppressInterval** angegeben. `msgIds` enthält eine durch Kommas getrennte Liste mit Nachrichten-IDs aus den folgenden:

7163 - Nachricht 'Job gestartet (nur IBM i)  
7234-Anzahl der geladenen Nachrichten  
8245  
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  
9508-Verbindung zum WS-Manager kann nicht hergestellt werden  
9524-Ferner WS-Manager nicht verfügbar  
9528-Benutzer hat das Schließen des Kanals angefordert  
9558-Ferner Kanal ist nicht verfügbar  
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.

#### **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.

### **IBM i Die Zeilengruppe 'TCP' unter IBM i**

Verwenden Sie diese WS-Manager-Eigenschaftenseiten oder -zeilengruppen in der Datei `qm.ini`, um Netzprotokollkonfigurationsparameter anzugeben. Sie überschreiben die Standardattribute für Kanäle.

**Anmerkung:** Es müssen nur Attribute angegeben werden, die Änderungen an den Standardwerten darstellen.

#### **TCP**

Die folgenden Attribute können angegeben werden:

##### **Port = 1414 | port\_number**

Die Standardportnummer (in Dezimalschreibweise) für TCP/IP-Sitzungen. Die Standardportnummer von IBM MQ 8.0 lautet 1414.

##### **KeepAlive= NO |YES**

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**

Beim Empfang über TCP/IP wird eine maximale Anzahl ausstehender Verbindungsanforderungen festgelegt. Dies kann als *Rückstand* von Anforderungen betrachtet werden, die auf den TCP/IP-Port warten, bis der Listener die Anforderung akzeptiert hat. Der Standardwert für den Rückstandsprotokollwert für IBM i beträgt 255; das Maximum ist 512. Wenn der Rückstand den Wert 512 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 sollte die Verbindung zu einem späteren Zeitpunkt wiederholen.

Mit dem Attribut `ListenerBacklog` können Sie die Standardanzahl ausstehender Anforderungen für den TCP/IP-Listener außer Kraft setzen.

**Connect\_Timeout= number | 0**

Die Anzahl der Sekunden, bevor ein Versuch unternommen wird, das Socket-Zeitlimit zu verbinden. Der Standardwert 0 gibt an, dass kein Verbindungszeitlimit vorhanden ist.

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.

**SndBuffSize= number | 0**

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.

**RcvBuffSize= number | 0**

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.

**RcvSndBuffSize= number | 0**

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.

**RcvRcvBuffSize= number | 0**

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.

**SvrSndBuffSize= number | 0**

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.

**SvrRcvBuffSize= number | 0**

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.

  **Die Optimierungsparameterzeilengruppe unter IBM i**

Verwenden Sie die Zeilengruppe `TuningParameters` in der Datei `qm.ini`, um Optionen für die Optimierung des Warteschlangenmanagers anzugeben.

**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 ein impliziter Synchronisationspunkt immer berücksichtigt wird.

Sie können einen beliebigen positiven ganzzahligen Wert festlegen. Wenn Sie nie einen impliziten Synchronisationspunkt hinzufügen möchten, setzen Sie ImplSyncOpenOutput = OFF.

## Zugehörige Informationen

[Impliziter Synchronisationspunkt](#)

## IBM i Beispieldateien mqs.ini und qm.ini für IBM i

### Beispiel für eine IBM MQ-Konfigurationsdatei

Das folgende Beispiel zeigt eine mqs.ini-Datei für 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
```

### Beispielkonfigurationsdatei des Warteschlangenmanagers

**V 9.0.5** Ab IBM MQ 9.0.5 zeigt das folgende Beispiel, wie Attributgruppen in einer Konfigurationsdatei des Warteschlangenmanagers für IBM i angeordnet 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

```

Dies ist eine Beispielkonfigurationsdatei für Versionen bis einschließlich IBM MQ 9.0.4 sowie für LTS:

```

#*****#
#* 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

```

### 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.

## Attribute für die Änderung von IBM MQ-Konfigurationsinformationen

Ändern Sie auf IBM MQ for Windows -Systemen und auf IBM MQ for Linux -Systemen (x86 -und x86-64 -Plattformen) die Konfigurationsdaten mit IBM MQ Explorer. Auf anderen Systemen ändern Sie die Informationen, indem Sie die Konfigurationsdatei 'mq.ini' bearbeiten.

Attribute für bestimmte Komponenten finden Sie in den folgenden Unterabschnitten:

### Zugehörige Konzepte

„IBM MQ konfigurieren“ auf Seite 5

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.

### Zugehörige Tasks

„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

„Konfigurationsdaten des Warteschlangenmanagers ändern“ auf Seite 122

Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

### Zugehörige Informationen

Planung

[IBM MQ verwalten](#)

## Alle WS-Manager

Verwenden Sie die Eigenschaftenseite General und Extended IBM MQ aus dem IBM MQ Explorer oder die Zeilengruppe AllQueueManagers in der Datei mq.ini, um die folgenden Informationen zu allen Warteschlangenmanagern anzugeben:

### DefaultPrefix= *directory\_name*

Dieses Attribut gibt den Pfad zum Verzeichnis 'qmgrs' an, in dem die WS-Manager-Daten 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:** Löschen Sie nicht das Verzeichnis /var/mqm/errors auf UNIX and Linux-Systemen oder das Verzeichnis \errors auf Windows-Systemen.

Anstatt das Standardpräfix zu ändern, können Sie die Umgebungsvariable MQSPREFIX nutzen, um das Standardpräfix **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 **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). Mit dem Attribut **ConvEBCDICNewline** können Sie angeben, 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 mit der Methode TABLE
- Alle anderen CCSIDs mit der Methode NL\_TO\_CF

Die möglichen ISO-CCSIDs werden im Abschnitt [Tabelle 11 auf Seite 116](#) aufgeführt.

<i>Tabelle 11. Liste der möglichen ISO-CCSIDs</i>	
<b>CCSID</b>	<b>Codetruppe</b>
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, nimmt **ConvEBCDICNewLine** standardmäßig den Wert NL\_TO\_LF an.

**V 9.0.0.6** Ab IBM MQ 9.0.0 Fix Pack 6 können Sie die **AMQ\_CONVEBCDICNEWLINE** -Umgebungsvariable anstelle des Zeilengruppenattributs **ConvEBCDICNewLine** verwenden, um beispielsweise **ConvEBCDICNewLine** -Funktionalität auf der Clientseite in Situationen bereitzustellen, in denen 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.

## Standard-WS-Manager

Verwenden Sie die Eigenschaftenseite General (Allgemein) von IBM MQ in IBM MQ Explorer oder die Zeilengruppe `DefaultQueueManager` in der Datei `'mqs.ini'`, um den Standardwarteschlangenmanager anzugeben.

### 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.

## Exiteigenschaften

Verwenden Sie die Extended IBM MQ -Eigenschaftenseite aus IBM MQ Explorer oder die Zeilengruppe `ExitProperties` in der Datei `mqs.ini`, um Konfigurationsoptionen anzugeben, die von Exitprogrammen des Warteschlangenmanagers verwendet werden.

### CLWLMode= SAFE | FAST

Mit dem Exit für Clusterauslastung (CLWL) können Sie angeben, welche Clusterwarteschlange in dem Cluster als Antwort auf einen MQI-Aufruf geöffnet werden soll (z. B. `MQOPEN`, `MQPUT`). Der CLWL-Exit wird entweder im FAST-Modus oder im SAFE-Modus ausgeführt, abhängig von dem Wert, den Sie im Attribut `CLWLMode` angeben. 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 (`amqzlw0`) 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 davon überzeugt sind, dass es **keine** Probleme mit dem 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.

## Protokollstandardwerte für IBM MQ

Verwenden Sie die Eigenschaftenseite `Default log settings` (Standardprotokolleinstellungen) von IBM MQ im IBM MQ Explorer oder die Zeilengruppe `LogDefaults` in der Datei `mqs.ini`, um Informationen zu Protokollstandardwerten für alle Warteschlangenmanager anzugeben.

Wenn die Zeilengruppe nicht vorhanden ist, werden die MQ-Standardwerte verwendet. Die Protokollattribute werden bei der Erstellung eines Warteschlangenmanagers als Standardwerte verwendet, können aber überschrieben werden, wenn Sie die Protokollattribute im Befehl `crtmqm` angeben. Ausführliche Informationen zu diesem Befehl finden Sie in `crtmqm`.

Sobald ein WS-Manager erstellt wurde, werden die Protokollattribute für diesen Warteschlangenmanager aus den in „[Warteschlangenmanagerprotokolle](#)“ auf Seite 126 beschriebenen Einstellungen übernommen.

Das Standardpräfix (angegeben in „[Alle WS-Manager](#)“ auf Seite 115 ) und der Protokollpfad, die für den jeweiligen Warteschlangenmanager angegeben sind (in „[Warteschlangenmanagerprotokolle](#)“ auf Seite 126 angegeben), ermöglichen dem Warteschlangenmanager und seinem Protokoll, sich auf verschiedenen physischen Laufwerken zu 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 568.

**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|2-254 ( Windows )|2-510 ( UNIX and Linux-Systeme)**

Die Protokolldateien, die beim Erstellen des Warteschlangenmanagers zugeordnet werden.

Die minimale Anzahl der von Ihnen verwendeten primären Protokolldateien ist 2 und die maximale Anzahl beträgt 254 unter Windows bzw. 510 unter UNIX und Linux. Der Standardwert ist 3.

Die Gesamtzahl der primären und sekundären Protokolldateien darf unter Windows nicht größer als 255 sein (bzw. 511 unter UNIX und Linux) 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.

#### **LogSecondaryFiles= 2|1-253 (Windows)|1-509 (UNIX und Linux)**

Die Protokolldateien, die zugeordnet werden, wenn die Primärdateien erschöpft sind.

Die minimale Anzahl der sekundären Protokolldateien beträgt 1 und die maximale Anzahl unter Windows ist 253 (bzw. 509 unter UNIX und Linux. Die Standardanzahl ist 2.

Die Gesamtzahl der primären und sekundären Protokolldateien darf unter Windows nicht größer als 255 sein (bzw. 511 unter UNIX und Linux) 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 Protokolldateien 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 UNIX und Linux beträgt die Mindestanzahl der Protokolldateiseiten 64, unter Windows ist die Mindestanzahl der 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 | 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 562.

##### **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 562.

Wenn Sie den Standardwert ändern möchten, können Sie entweder das Attribut LogType bearbeiten oder die lineare Protokollierung mit dem Befehl `crtmqm` angeben.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie die Protokollierungsmethode nach der Erstellung eines Warteschlangenmanagers ändern. Weitere Informationen finden Sie im Abschnitt `migmqlog`.

### **LogBufferPages= 0 | 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. In IBM WebSphere MQ 7.1 ist dies 512 (2048 KB).

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:

- `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.
- `/var/mqm/log` für IBM MQ for UNIX- und Linux-Systeme

Alternativ können Sie den Namen eines Verzeichnisses auf dem Befehl `crtmqm` unter Verwendung der Markierung `-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` nicht auf dem Befehl `crtmqm` angeben, wird der Wert des Attributs `LogDefaultPath` in der Datei `mq.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**

Die Methode, die die Protokollfunktion verwendet, um Protokollsätze zuverlässig zu schreiben.

#### **TripleWrite**

Dies ist die Standardmethode.

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.

## Advanced Configuration and Power Interface (ACPI)

Auf der Eigenschaftenseite von ACPI IBM MQ im IBM MQ Explorer können Sie angeben, wie sich IBM MQ verhalten soll, wenn das System eine Aussetzungsanforderung empfängt.

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.

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 Aussetzungsanforderung 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 <u>Anmerkung</u> ; 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 <u>Anmerkung</u> ), akzeptieren Sie die Anforderung.

Y	Y	N	Den Dialog anzeigen ( <a href="#">Hinweis</a> ; die Aussetzungsanforderung verweigern).
Y	Y	Y	Wenn keine Kanäle aktiv sind, akzeptieren Sie die Aussetzungsanforderung; wenn sie den Dialog anzeigen ( <a href="#">Hinweis</a> ; 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.

## API-Exits

Verwenden Sie den IBM MQ Explorer oder den Befehl `amqmdain`, um die Einträge für API-Exits zu ändern.

Verwenden Sie die Exits IBM MQ -Eigenschaftenseite aus IBM MQ Explorer oder die Zeilengruppe `ApiExitTemplate` und `ApiExitCommon` in der Datei `mq.ini`, um API-Exitroutinen für alle Warteschlangenmanager anzugeben. 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 einzelne Warteschlangenmanager wird die Zeilengruppe `ApiExitLocal` wie in [„Warteschlangenmanagerkonfigurationsdateien, qm.ini“](#) auf Seite 97 beschrieben verwendet.)

Eine vollständige Beschreibung der Attribute für diese Zeilengruppen finden Sie im Abschnitt [API-Exits konfigurieren](#).

## Warteschlangenmanager

Für jeden WS-Manager gibt es eine Zeilengruppe `QueueManager`. Verwenden Sie die Zeilengruppe, um die Position des Warteschlangenmanagerverzeichnisses anzugeben.

Auf Windows-, UNIX and Linux-System gibt es für jeden Warteschlangenmanager je eine Zeilengruppe `QueueManager`. Diese Attribute geben den Namen des Warteschlangenmanagers und den Namen des Verzeichnisses an, das die Dateien enthält, die diesem WS-Manager 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 den IBM MQ-Dateinamen](#).

### **Name = *queue\_manager\_name***

Der Name des Warteschlangenmanagers.

### **Präfix = *prefix***

Gibt an, wo die Warteschlangenmanagerdateien gespeichert sind. Standardmäßig ist dieser Wert mit dem Wert identisch, der im Attribut `DefaultPrefix` der Informationen zu allen WS-Managern angegeben ist.

### **Verzeichnis = *name***

Der Name des Unterverzeichnisses unter dem Verzeichnis `prefix\QMGRS`, in dem die WS-Manager-Dateien 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 Dateiname ist.

### **Datenpfad = *path***

Ein expliziter Datenpfad, der beim Erstellen des Warteschlangenmanagers bereitgestellt wurde, überschreibt `Prefix` und `Directory` als Pfad zu den WS-Manager-Daten.

### **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. Wenn kein Wert für `InstallationName` vorhanden ist, wird der Warteschlangenmanager einer Installation des Produkts vor IBM WebSphere MQ 7.1 zugeordnet.

## Zugehörige Tasks

„WS-Manager einer Installation zuordnen“ auf Seite 421

Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den Befehl **crtmqm** ausgegeben hat. Unter UNIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

## Konfigurationsdaten des Warteschlangenmanagers ändern

Die Attribute, die Sie zum Ändern der Konfiguration eines einzelnen Warteschlangenmanagers verwenden können, überschreiben alle Einstellungen für IBM MQ.

### Informationen zu diesem Vorgang

Auf UNIX and Linux-Systemen ändern Sie die Konfigurationsinformationen des Warteschlangenmanagers, indem Sie die `qm.ini`-Konfigurationsdatei bearbeiten. Wenn Sie eine Zeilengruppe in `qm.ini` definieren, müssen Sie die einzelnen Elemente nicht in einer neuen Zeile starten. Sie können entweder ein Semikolon (;) oder ein Nummernzeichen (#) verwenden, um einen Kommentar anzugeben.

Auf Windows- und Linux x86-64-Systemen können Sie einige Konfigurationsinformationen unter Verwendung des IBM MQ Explorer ändern. Da es jedoch erhebliche Auswirkungen auf die Änderung installierbarer Services und ihrer Komponenten hat, sind die installierbaren Services in der IBM MQ Explorer Schreibgeschützt. Sie müssen daher Änderungen an installierbaren Services vornehmen, indem Sie **regedit** in Windows verwenden und die Datei `qm.ini` unter UNIX and Linux bearbeiten.

### Prozedur

- Weitere Informationen zum Ändern der Konfigurationsinformationen des Warteschlangenmanagers finden Sie in den folgenden Unterabschnitten:

#### Zugehörige Konzepte

„IBM MQ konfigurieren“ auf Seite 5

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.

#### Zugehörige Tasks

„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

#### Zugehörige Verweise

„Attribute für die Änderung von IBM MQ-Konfigurationsinformationen“ auf Seite 115

Ändern Sie auf IBM MQ for Windows -Systemen und auf IBM MQ for Linux -Systemen (x86 - und x86-64 -Plattformen) die Konfigurationsdaten mit IBM MQ Explorer. Auf anderen Systemen ändern Sie die Informationen, indem Sie die Konfigurationsdatei 'mqm.ini' bearbeiten.

#### Zugehörige Informationen

[Planung](#)

[IBM MQ verwalten](#)

### **Windows** Zugriffsmodus

**Access Mode** gilt nur für Windows-Server. Die AccessMode-Zeilengruppe wird durch die Option `-a [ r ]` auf dem 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 gemeinsam zu nutzen.

Die zusätzliche Sicherheitszugriffsgruppe kann alternativ zur lokalen Gruppe mqm verwendet werden, um Berechtigungen für den Ordner festzulegen, der Warteschlangenmanagerdaten und Protokolle enthält. Im Gegensatz zur lokalen Gruppe mqm können Sie die zusätzliche Sicherheitszugriffsgruppe zu einer lokalen oder globalen Gruppe machen. 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 das Windows-Betriebssystem die Benutzer-ID, die den Befehl **strmqm** ausgeführt hat.

Die Benutzer-ID muss Mitglied der lokalen Gruppe mqm 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.

### **Zugehörige Konzepte**

[„Nicht gemeinsam genutzte WS-Manager-Daten und -Protokollverzeichnisse und -Dateien unter Windows schützen“ auf Seite 502](#)

[„Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern“ auf Seite 499](#)

### **Zugehörige Tasks**

[„WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen“ auf Seite 472](#)

### **Zugehörige Informationen**

[crtmqm](#)

## **ULW** Installierbare Services konfigurieren

Sie können die installierbaren Services unter Windows mithilfe von **regedit** und unter UNIX and Linux mit der Zeilengruppe Service in der Datei qm.ini ändern.

**Anmerkung:** Es hat erhebliche Auswirkungen auf die Änderung installierbarer Services und ihrer Komponenten. Aus diesem Grund sind die installierbaren Services im IBM MQ Explorer schreibgeschützt.

Wenn Sie installierbare Services auf Windows-Systemen ändern möchten, verwenden Sie **regedit** bzw. auf UNIX and Linux-Systemen die Zeilengruppe `Service` in der Datei `qm.ini`. 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 auf UNIX and Linux-Systemen die Zeilengruppe `ServiceComponent`.

### **Name = AuthorizationService | NameService**

Der Name des erforderlichen Service.

#### **AuthorizationService**

Bei IBM MQ wird die Komponente "Authorization Service" als Objektberechtigungsmanager oder OAM bezeichnet. Die Zeilengruppe `AuthorizationService` und die zugehörige Zeilengruppe `ServiceComponent` werden bei der Erstellung des Warteschlangenmanagers automatisch hinzugefügt. Fügen Sie andere `ServiceComponent` -Zeilengruppen manuell hinzu.

#### **NameService**

Standardmäßig wird kein Namensservice bereitgestellt. Wenn Sie einen Namensservice benötigen, müssen Sie die Zeilengruppe `NameService` manuell hinzufügen.

### **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 es sich bei dem angegebenen Service um den Standardberechtigungs-service handelt, d. h. den OAM. Mit dem Attribut "SecurityPolicy" können Sie die Sicherheitsrichtlinie für die einzelnen Warteschlangenmanager 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 hierzu finden Sie im Abschnitt [Windows-Sicherheitskennungen \(SIDs\)](#).

Siehe auch [Zeilengruppen für Berechtigungsservice konfigurieren: Windows-Systeme](#).

### **Linux UNIX SecurityPolicy = user|group|default**

Auf UNIX and Linux-Systemen gibt der Wert an, ob der WS-Manager eine benutzerbasierte oder gruppenbasierte Berechtigung verwendet. Bei Werten wird die Groß-/Kleinschreibung nicht beachtet.

Wenn Sie dieses Attribut nicht angeben, wird `default` verwendet, bei dem die gruppenbasierte Berechtigung verwendet wird. Starten Sie den Warteschlangenmanager neu, damit die Änderungen wirksam werden. Siehe auch [Zeilengruppen für Berechtigungsservice konfigurieren: UNIX and Linux-Systeme](#).

### **SharedBindingsUserId= user-type**

Das Attribut `SharedBindingsUserId` gilt nur, wenn es sich bei dem angegebenen Service um den Standardberechtigungs-service handelt, d. h. den OAM. Das Attribut `SharedBindingsUserId` wird nur in Verbindung mit gemeinsam genutzten 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 reale Benutzer-ID festgelegt.

**Real**

Der Wert des Feldes *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 es sich bei dem angegebenen Service um den Standardberechtigungs-service handelt, d. h. den OAM. Das Attribut 'FastpathBindingsUserId' wird nur in Verbindung mit den Direktauf-rufbindungen 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 es sich bei dem angegebenen Service um den Standardberechtigungs-service handelt, d. h. den OAM. Das Attribut **IsolatedBindingsUserId** wird nur in Verbindung mit isolierten 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 des Feldes *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 UNIX, Linux, and Windows](#).

Weitere Informationen zu Sicherheitsservices im Allgemeinen finden Sie im Abschnitt [Sicherheit auf UNIX and Linux-Systemen konfigurieren](#).

**Zugehörige Informationen**

[Referenzinformationen zu installierbaren Services](#)

 **Servicekomponenten**

Sie müssen Servicekomponenteninformationen angeben, wenn Sie einen neuen installierbaren Service hinzufügen. Verwenden Sie auf Windows-Systemen **regedit** und auf UNIX and Linux-Systemen die Zeilengruppe **ServiceComponent** in der Datei 'qm.ini'. Die Berechtigungsservice-Zeilengruppe ist standardmäßig vorhanden, und die zugeordnete Komponente (OAM) ist aktiv.

Geben Sie die Servicekomponenten wie folgt an:

**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.

Weitere Informationen zu installierbaren Services und Komponenten finden Sie unter [Installierbare Services und Komponenten für UNIX, Linux und Windows](#).

**Warteschlangenmanagerprotokolle**

Verwenden Sie die Eigenschaftenseite Log des Warteschlangenmanagers im IBM MQ Explorer oder die Zeilengruppe Log in der Datei 'qm.ini', um Informationen zur Protokollierung in einem Warteschlangenmanager anzugeben.

Standardmäßig werden diese Einstellungen aus den Standardprotokolleinstellungen des Warteschlangenmanagers übernommen (Beschreibung siehe „Protokollstandardwerte für IBM MQ“ auf Seite 117). Ändern Sie diese Einstellungen nur, wenn Sie diesen Warteschlangenmanager auf eine andere Weise konfigurieren möchten.

Informationen zum Berechnen der Protokollgrößen finden Sie in „Berechnen der Größe des Protokolls“ auf Seite 568.

**Anmerkung:** Die in der folgenden Parameterliste angegebenen Grenzwerte werden durch IBM MQ festgelegt. Durch die Begrenzung des Betriebssystems kann die maximal mögliche Protokollgröße reduziert werden.

**LogPrimaryFiles= 3|2-254 ( Windows )|2-510 ( UNIX 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 UNIX 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 UNIX 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.

**LogSecondaryFiles= 2|1-253 ( Windows )|1-509 ( UNIX 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 UNIX 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 UNIX 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 UNIX 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 bei der Erstellung des Warteschlangenmanagers angegeben wurden, kann für einen Warteschlangenmanager nicht geändert werden.

**LogType= CIRCULAR | LINEAR**

Der Typ der Protokollierung, der vom Warteschlangenmanager verwendet werden soll. Sie können den Typ der Protokollierung nicht ändern, die nach der Erstellung des Warteschlangenmanagers verwendet werden soll. Informationen zum Erstellen eines Warteschlangenmanagers mit dem Typ der Protokollierung, die Sie benötigen, finden Sie in der Beschreibung des Attributs LogType in „[Protokollstandardwerte für IBM MQ](#)“ auf Seite 117.

**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 562.

**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 562.

**LogBufferPages= 0 | 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. In IBM WebSphere MQ 7.1 ist dies 512 (2048 KB).

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 WS-Manager die angegebene Zahl, um den zugeordneten Speicher 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:

- C:\ProgramData\IBM\MQ\log in IBM MQ for Windows.
- /var/mqm/log in IBM MQ for UNIX- und Linux-Systemen.

Sie können den Namen eines Verzeichnisses auf dem Befehl `crtmqm` unter Verwendung der Markierung `-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` nicht auf dem Befehl `crtmqm` angeben, wird der Wert des Attributs `LogDefaultPath` verwendet.

In IBM MQ for UNIX- und Linux-Systemen müssen die Benutzer-ID `mqm` und die Gruppe `mqm` über die vollständigen 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**

Die Methode, die die Protokollfunktion verwendet, um Protokollsätze zuverlässig zu schreiben.

#### **TripleWrite**

Dies ist die Standardmethode.

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.

V 9.0.2

### **LogManagement= Manuell | Automatisch | Archiv**

Die Methode, mit der Protokoll extents entweder manuell oder vom Warteschlangenmanager verwaltet werden.

Das Attribut gilt nur, wenn **LogType** LINEAR ist.

Wenn Sie den Wert für **LogManagement** ändern, wird die Änderung erst wirksam, wenn der WS-Manager erneut gestartet wird.

Wenn ein nicht erkannter Wert für das Attribut gefunden wird, wird der WS-Manager erst gestartet, wenn der Wert korrigiert wird.

#### **Manuell**

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 Befehl Reset Queue Manager PCF ausgeführt.

Linux

UNIX

## Eingeschränkter Modus

Diese Option gilt nur für UNIX and Linux-Systeme. Die Zeilengruppe `RestrictedMode` wird durch die Option **-g** auf dem 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.

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:** Bei vielen Betriebssystemen muss der Benutzer sich abmelden und erneut anmelden, um die Hinzufügung eines Benutzers zu einer zu erkennbaren Gruppe zu erhalten.

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.

Weitere Informationen zum Konfigurieren des IBM MQ-Berechtigungsservice finden Sie unter [Sicherheit auf Windows-, UNIX and Linux-Systemen konfigurieren](#).

Verwenden Sie nur den eingeschränkten Modus von IBM MQ, wenn die vom Berechtigungsservice bereitgestellte Steuerung keine ausreichende Isolation von Warteschlangenmanagerressourcen bietet.

## XA-Ressourcenmanager

Verwenden Sie die Eigenschaftenseite des XA `resource manager`-Warteschlangenmanagers aus IBM MQ Explorer oder die Zeilengruppe `XAResourceManager` in der Datei `qm.ini`, um die folgenden Informationen zu den Ressourcenmanagern anzugeben, die an globalen Arbeitseinheiten beteiligt sind, die vom Warteschlangenmanager koordiniert werden.

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 `dspmqtin`.

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 IBM MQ for 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.

## **Attribute der Zeilengruppe 'channels'**

Diese Attribute bestimmen die Konfiguration eines Kanals.

**z/OS** Diese Informationen gelten nicht für IBM MQ for z/OS.

Auf der Eigenschaftenseite des Channels -Warteschlangenmanagers aus dem IBM MQ Explorer oder in der Zeilengruppe CHANNELS in der Datei qm.ini können Sie Informationen zu Kanälen angeben.

### **MaxChannels= 100 | *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 | *number***

Die maximale Anzahl der Initiatoren. Der Standardwert und der Maximalwert sind 3.

### **MQBindType = 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.

**AdoptNewMCA= NO | 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 nicht auf diese Anforderung reagiert, wenn das Warteintervall für *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**

Empfänger-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 | 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 aufricht. Alternativ können Sie eine HACMP-Typ-lösung 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`.

### **V 9.0.4**

#### **ChlauthEarlyAdopt = Y | 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 das 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 | immer | optional**

Geben Sie ab IBM MQ 8.0 geschützte Kennwörter in der MQCSP-Struktur an, statt TLS zu verwenden.

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 finden Sie im Abschnitt [MQCSP-Kennwortschutz](#).

### **ChlauthIssueWarn = y**

Legen Sie dieses Attribut fest, wenn die Nachricht AMQ9787 beim Festlegen des Attributs WARN = YES im Befehl SET CHLAUTH generiert werden soll.

### **Zugehörige Konzepte**

„Kanalstatus“ auf Seite 186

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.

## **TCP, LU62 und NETBIOS**

Verwenden Sie diese WS-Manager-Eigenschaftenseiten oder Zeilengruppen in der Datei qm.ini, um Netzprotokollkonfigurationsparameter anzugeben. Sie überschreiben die Standardattribute für Kanäle.

### **TCP**

Verwenden Sie die Eigenschaftenseite des TCP-Warteschlangenmanagers aus der Zeilengruppe IBM MQ Explorer oder der Zeilengruppe TCP in der Datei qm.ini, um TCP/IP-Konfigurationsparameter (TCP/IP-Transmission Control Protocol/Internet Protocol) anzugeben.

#### **Port= 1414 | port\_number**

Die Standardportnummer (in Dezimalschreibweise) für TCP/IP-Sitzungen. Die *Standardportnummer* für IBM MQ ist 1414.

#### **Library1= DLLName1 (nur IBM MQ for Windows)**

Der Name der TCP/IP-Sockets-DLL.

Der Standardwert ist WSOCK32.

**KeepAlive= NO |YES**

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 12 auf Seite 135](#) angezeigt.

<i>Tabelle 12. Ausstehende Standardverbindungsanforderungen (TCP)</i>	
<b>Plattform</b>	<b>Standardwert für ListenerBacklog</b>
Windows-Server	100
Windows-Workstation	5
Linux	100
Solaris	100
HP-UX	20
AIX 5.3 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 12 auf Seite 135](#) 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 | Zahl**

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 Erfolgchancen 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=number|0**

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 „Warteschlangenmanagerkonfigurationsdateien, qm.ini“ auf Seite 97).

**RcvBuffSize=number|0**

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 „Warteschlangenmanagerkonfigurationsdateien, qm.ini“ auf Seite 97).

**RcvSndBuffSize=number|0**

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 „Warteschlangenmanagerkonfigurationsdateien, qm.ini“ auf Seite 97).

**RcvRcvBuffSize=number|0**

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 „Warteschlangenmanagerkonfigurationsdateien, qm.ini“ auf Seite 97).

**SvrSndBuffSize=number|0**

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 „Warteschlangenmanagerkonfigurationsdateien, qm.ini“ auf Seite 97).

**SvrRcvBuffSize=number|0**

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 „Warteschlangenmanagerkonfigurationsdateien, qm.ini“ auf Seite 97).

### Windows **LU62 (nur IBM MQ for Windows)**

Verwenden Sie die Eigenschaftenseite des LU6.2-Warteschlangenmanagers aus der Zeilengruppe IBM MQ Explorer oder der Zeilengruppe LU62 in der Datei qm.ini, um die Konfigurationsparameter für die SNA LU 6.2-Protokolle anzugeben.

#### **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.

### Windows **NETBIOS (nur IBM MQ for Windows)**

Verwenden Sie die Eigenschaftenseite des Netbios-Warteschlangenmanagers aus der Zeilengruppe IBM MQ Explorer oder der Zeilengruppe NETBIOS in der Datei qm.ini, um die Konfigurationsparameter für das NetBIOS-Protokoll anzugeben.

#### **LocalName= name**

Der Name, unter dem diese Maschine im LAN bekannt ist.

#### **AdapterNum= 0| adapter\_number**

Die Nummer des LAN-Adapters. Der Standardwert ist Adapter 0.

#### **NumSess= 1| number\_of\_sessions**

Die Anzahl der Sitzungen, die zugeordnet werden sollen. Der Standardwert ist 1.

#### **NumCmds= 1| number\_of\_commands**

Die Anzahl der Befehle, die zugeordnet werden sollen. Der Standardwert ist 1.

#### **NumNames= 1| number\_of\_names**

Die Anzahl der zuzuordnende Namen. Der Standardwert ist 1.

#### **Library1= DLLName1**

Der Name der NetBIOS-DLL.

Der Standardwert ist NETAPI32.

### Windows **SPX (nur IBM MQ for Windows)**

Verwenden Sie die Eigenschaftenseite des SPX-Warteschlangenmanagers aus der Zeilengruppe IBM MQ Explorer oder der Zeilengruppe SPX in der Datei qm.ini, um SPX-Protokollkonfigurationsparameter anzugeben.

#### **Socket= 5E86| socket\_number**

Die SPX-Socket-Nummer in Hexadezimalschreibweise. Der Standardwert ist X'5E86 '.

#### **BoardNum= 0| adapter\_number**

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 138](#) angezeigt.

<i>Tabelle 13. Ausstehende Standardverbindungsanforderungen (SPX)</i>	
<b>Plattform</b>	<b>Standardwert für ListenerBacklog</b>
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 138](#) 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.

## **Exitpfad**

Verwenden Sie die Eigenschaftenseite des Exits -Warteschlangenmanagers aus IBM MQ Explorer oder die Zeilengruppe ExitPath in der Datei qm.ini, um den Pfad für Benutzerexitprogramme auf dem Warteschlangenmanagersystem anzugeben.

### **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

## **API-Exits**

Verwenden Sie für einen Server die Eigenschaftenseite Exits des Warteschlangenmanagers im IBM MQ Explorer oder die Zeilengruppe ApiExitLocal in der Datei qm.ini, um API-Exitroutinen für einen Warteschlangenmanager zu identifizieren. Ändern Sie für einen Client die Zeilengruppe ApiExitLocal in der Datei mqclient.ini, um API-Exitroutinen für einen Warteschlangenmanager zu identifizieren.

Auf Windows-Systemen können Sie auch den Befehl 'amqmdain' verwenden, um die Einträge für API-Exits zu ändern. (Um API-Exitroutinen für alle Warteschlangenmanager zu identifizieren, verwenden Sie die Zeilengruppen ApiExitCommon und ApiExitTemplate, wie in [„API-Exits“ auf Seite 121](#) 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.

Eine vollständige Beschreibung der Attribute für diese Zeilengruppen finden Sie im Abschnitt [API-Exits konfigurieren](#).

## V 9.0.5 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* bezeichnet, enthalten jedoch immer IBM MQ -Informations- und -Warnungen sowie Fehlernachrichten. 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

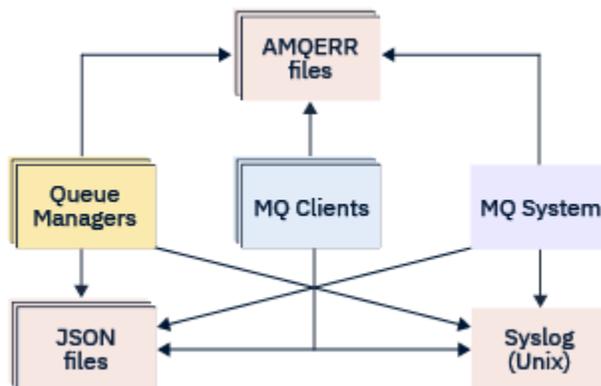
IBM MQ unterstützt die Berichterstellung von Diagnosenachrichten durch eine Reihe unterschiedlicher Methoden, die als *Diagnosenachrichtenservices* bezeichnet werden. Dies ermöglicht einen angepassten Ansatz zum Aufzeichnen und Verarbeiten dieser Informationen:

- AMQERRnn, Protokolldateien
- JSON-formatierte Protokolldateien
-  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 6. 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. Beispiel:

- 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 142 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 142

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 unter UNIX, Linux, and Windows“ auf Seite 141

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.

„Diagnosenachrichtenservices“ auf Seite 145

Die folgenden Diagnosenachrichtenservices und ihre servicespezifischen Attribute, die in den Zeilengruppen DiagnosticSystemMessages, DiagnosticMessages und DiagnosticMessagesTemplate der Konfigurationsdateien angegeben sind, können definiert werden:

### **ULW** **QMErrorLog-Zeilengruppe unter UNIX, Linux, and Windows**

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.

Diese Zeilengruppe ist nicht auf die Konfigurationen der IBM MQ-System- oder Client-ini-Datei anwendbar.

Verwenden Sie die erweiterte WS-Manager-Eigenschaftenseite aus dem IBM MQ Explorer oder die Zeilengruppe 'QMErrorLog' in der Datei 'qm.ini', um die Operation und den Inhalt von IBM MQ-Fehlerprotokollen anzupassen.



**Achtung:** Sie können IBM MQ Explorer nur dann verwenden, um die Änderungen vorzunehmen, wenn Sie einen lokalen Warteschlangenmanager auf der Windows-Plattform verwenden.

### **V 9.0.4** **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.

**Wichtig:** Ab IBM MQ 9.0.4 wurde die Standardgröße des Attributs **ErrorLogSize** erhöht. Dies ist eine Änderung gegenüber IBM MQ 9.0.3.

Sie können die Größe des Protokolls mithilfe 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 142.

### **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 gilt nicht für Diagnosenachrichtenservices, die in MQ `client.ini` definiert sind.

Weitere Informationen hierzu finden Sie unter [SuppressMessage](#) in „Zeilengruppen für Diagnosenachrichtenservice“ auf Seite 142.

### **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.

### **Zugehörige Verweise**

„Zeilengruppen für Diagnosenachrichtenservice“ auf Seite 142

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.

„Diagnosenachrichtenservices“ auf Seite 145

Die folgenden Diagnosenachrichtenservices und ihre servicespezifischen Attribute, die in den Zeilengruppen `DiagnosticSystemMessages`, `DiagnosticMessages` und `DiagnosticMessagesTemplate` der Konfigurationsdateien angegeben sind, können definiert werden:

### V 9.0.5 **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 Dateien `mqs.ini` oder `mqclient.ini`.

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. Dieser Wert ist 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 werden.

Wenn Sie `syslog` beispielsweise als zusätzlichen Service aktivieren möchten, geben Sie Folgendes ein:

```
Service=syslog
```

Siehe „[Diagnosenachrichtenservices](#)“ auf Seite 145 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 144

**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.
- 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.
- 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 gilt nicht für Diagnosenachrichtenservices, die in `MQ client.ini` definiert sind.

### **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.

### **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 `a11`.
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 unter UNIX, Linux, and Windows“ auf Seite 141](#)

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.

[„Diagnosenachrichtenservices“ auf Seite 145](#)

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 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, in den die Protokolldateien geschrieben werden. Der Standardwert ist die gleiche Position wie die AMQERR01.Logdateien, d. -E. der System- oder WS-Manager. Der Pfad muss absolut sein, er kann jedoch austauschbare Einfügungen enthalten. Beispiel:

#### + MQ\_Q\_MGR\_DATA\_PATH +

Der vollständige Pfad zum übergeordneten Verzeichnis des Nachrichtenverzeichnisses des Warteschlangenmanagerdiagnoseprogramms. Standardmäßig auf UNIX-Plattformen: /var/mqm/qmgrs/<QM\_name> und unter Windows C:\Program Data\IBM\MQ\qmgrs\<QM\_name>.

#### + MQ\_DATA\_PATH +

Der vollständige Pfad zum übergeordneten Element des Nachrichtenverzeichnisses für Systemdiagnosen. Standardmäßig auf UNIX-Plattformen: /var/mqm und 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 Fehlerprotokollgröße-Eigenschaft des „QMErrorLog-Zeilengruppe unter UNIX, Linux, and Windows“ auf Seite 141, die semantisch identisch ist.

**Anmerkung:** Die Eigenschaft **ErrorLogSize** gilt nur für den Standardfehlerprotokollservice, nicht für angepasste Diagnoseservices.

Ab IBM MQ 9.0.5 können Sie die Größe des Protokolls mithilfe 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 zum Beispiel die Datei `qm.ini` des Warteschlangenmanagers und fügen Sie die folgende Zeilengruppe hinzu:

```
DiagnosticMessages:
  Service = File
  Name = JSONLogs
  Format = json
  FilePrefix = AMQERR
```

Nach dem Neustart verfügt der Warteschlangenmanager über AMQERR0x.json-Dateien im 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
```

### Systemprotokoll

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 mit Hilfe der JSON-Format Diagnosenachrichten-Spezifikation 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:

Bewertung	Stufe
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  
  Service=Syslog  
  Severities=E +  
  FilePrefix=AllErrors
```

Weitere Informationen zu generischen Zeilengruppenattributen finden Sie unter „Zeilengruppen für Diagnosenachrichtenservice“ auf Seite 142.

#### 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.

## Standardbindungsart des WS-Managers

Verwenden Sie die Eigenschaftenseite `Extended` (Erweitert) für den Warteschlangenmanager im IBM MQ Explorer oder die Zeilengruppe `Connection` in der Datei `qm.ini`, um den Standardbindungstyp anzugeben. Beachten Sie, dass Sie eine Zeilengruppe `Connection` erstellen müssen, wenn Sie eine Zeilengruppe benötigen.

### DefaultBindType= SHARED | ISOLATED

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.

Der Standardwert ist `SHARED`.



**Achtung:** `DefaultBindType` gilt für alle `MQCONN`-Aufrufe und alle `MQCONN`-Aufrufe mit `MQCNO_STANDARD_BINDING`.

Das Ändern von `DefaultBindType` kann dazu führen, dass die Leistung einiger Anwendungen abnimmt.

## SSL-Zeilengruppe der Konfigurationsdatei des Warteschlangenmanagers

Verwenden Sie die SSL-Zeilengruppe der WS-Manager-Konfigurationsdatei, um TLS-Kanäle in Ihrem 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 widerrufenen 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 in 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 Fehlermeldungen geschrieben, die darauf hingewiesen werden, dass ein ungültiger Wert angegeben wurde.

### CDPCheckExtensions = YES | NO

`CDPCheckExtensions` 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 : TLS-Kanäle versuchen nicht, die CDP-Server zu überprüfen. Dieser Wert stellt den Standardwert dar.

#### **OCSPAAuthentication= REQUIRED | WARN | OPTIONAL**

OCSPAAuthentication 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 OCSPAAuthentication verwendet.

- ERFORDERLICH : Wenn der Widerrufsstatus nicht bestimmt wird, 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= JA | NEIN**

OCSPCheckExtensions gibt an, ob TLS-Kanäle in diesem Warteschlangenmanager versuchen, die OCSP-Server zu überprüfen, die in den Erweiterungen von AuthorityInfoAccess-Zertifikaten angegeben sind.

- YES : TLS-Kanäle versuchen, die 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.

#### **SSLHTTPProxyName= *string***

Die Zeichenfolge ist entweder der Hostname oder die Netzadresse des HTTP-Proxy-Servers, 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.

 Solaris  HP-UX  AIX Für 32-Bit-Clients unter AIX, und auf Solaris-SPARC-Plattformen und HP-UX-PA-RISC-Plattformen 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.

## **Exiteigenschaften**

Verwenden Sie die Eigenschaftenseite "Cluster" des Warteschlangenmanagers im IBM MQ Explorer oder der Zeilengruppe 'ExitPropertiesLocal' in der Datei 'qm.ini', um Informationen zu Exiteigenschaften in einem Warteschlangenmanager anzugeben. Alternativ können Sie sie mit dem Befehl **amqmdain** festlegen.

Diese Einstellung wird standardmäßig aus dem Attribut CLWLMode in der Zeilengruppe 'ExitProperties' der maschinenweiten Konfiguration übernommen (siehe „Exiteigenschaften“ auf Seite 117). 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.

#### **CLWLMode= SAFE | FAST**

Mit dem Exit für Clusterauslastung (CLWL) können Sie angeben, welche Clusterwarteschlange in dem Cluster als Antwort auf einen MQI-Aufruf geöffnet werden soll (z. B. MQOPEN , MQPUT ). Der

CLWL-Exit wird entweder im FAST-Modus oder im SAFE-Modus ausgeführt, abhängig von dem Wert, den Sie im Attribut CLWLMode angeben. 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 (amqzlw0) 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 davon überzeugt sind, dass es **keine** Probleme mit dem 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.

## Subpool

Diese Zeilengruppe wird von IBM MQ erstellt. Ändern Sie sie nicht.

Der Zeilengruppensubpool 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 Verzeichnis /var/mqm/sockets erstellt wurde, das IBM MQ für die interne Kommunikation zwischen den aktiven Prozessen verwendet.

### IBM i Linux UNIX Zeilengruppe 'Filesystem' in der Datei 'qm.ini'

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.

Ihr IBM MQ-Administrator möchte jedoch vielleicht die Berechtigungen für die zugehörigen Fehlerprotokolldateien ändern. In diesem Fall müssen sie die Zeilengruppe-Option 'Filesystem' **ValidateAuth=Nein** festlegen, was dazu führt, dass der Warteschlangenmanager die Berechtigungen danach unverändert lassen kann.

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.

## Sicherheit

Verwenden Sie die Zeilengruppe `Security` in der Datei `qm.ini`, um Optionen für den OAM (Object Authority Manager) anzugeben.

### **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.

### **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; weitere Informationen finden Sie in den Informationen zum Parameter **`setmqaut -g`**.

**Anmerkung:** Wenn Sie `ClusterQueueAccessControl=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 `ClusterQueueAccessControl=Xmitq` ein oder führen Sie ein Upgrade des angepassten Berechtigungsservice auf `MQZAS_VERSION_6` oder höher aus.

## **V 9.0.5** Tuning-Parameter

Verwenden Sie die Zeilengruppe `TuningParameters` in der Datei `qm.ini`, um Optionen für die Optimierung des Warteschlangenmanagers anzugeben.

### **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 ein impliziter Synchronisationspunkt immer berücksichtigt wird.

Sie können einen beliebigen positiven ganzzahligen Wert festlegen. Wenn Sie nie einen impliziten Synchronisationspunkt hinzufügen möchten, setzen Sie `ImplSyncOpenOutput = OFF`.

### **V 9.0.0.12** **OAMLdapConnectTimeout=Maximale Zeit**

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.

### **OAMLdapQueryTimeLimit=Maximale Zeit**

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.

#### **Zugehörige Informationen**

Impliziter Synchronisationspunkt

## **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.

### **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 152
  - „Einführung in die verteilte Warteschlangenverwaltung“ auf Seite 174
  - „So senden Sie eine Nachricht an einen anderen Warteschlangenmanager“ auf Seite 177
  - „Ausgelöste Kanäle“ auf Seite 200
  - „Sicherheit von Nachrichten“ auf Seite 198
  -  „Kanäle unter UNIX, Linux, and Windows überwachen und steuern“ auf Seite 208
  -  „Kanäle in IBM i überwachen und steuern“ auf Seite 233

### **Zugehörige Konzepte**

„IBM MQ for z/OS konfigurieren“ auf Seite 702

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“ auf Seite 773

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.

### **Zugehörige Tasks**

„Verbindungen zwischen dem Server und dem Client konfigurieren“ auf Seite 17

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 255

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.

„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93

Sie können das Verhalten von IBM MQ oder eines einzelnen Warteschlangenmanagers an die Anforderungen Ihrer Installation anpassen.

„Warteschlangenmanager unter z/OS erstellen“ auf Seite 697

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

## 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:

- ▶ **ULW** [Beispiel für Nachrichtenkanalplanung für UNIX, 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 Tasks

„Verteilte Warteschlangensteuerung konfigurieren“ auf Seite 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### Zugehörige Informationen

[Kanäle](#)

[Einführung in die Nachrichtenwarteschlangensteuerung](#)

[Verteilte Warteschlangen und Cluster](#)

[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 151 eingeführt wurden. Wenn sich Ihr Warteschlangenmanager in einem Cluster befindet, wird der Nachrichtenfluss unter Verwendung unterschiedlicher Verfahren gesteuert, wie in „Nachrichtenflusssteuerung“ auf Seite 152 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 153

Zielwarteschlangennamen werden mit der Nachricht in den Übertragungsheader übertragen, bis die Zielwarteschlange erreicht ist.

„Vorgehensweise zum Erstellen von Warteschlangenmanagern und Antwortaliasnamen“ auf Seite 153

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 Warteschlangename, der Name der logischen Warteschlange, wird vom WS-Manager in den Namen der Zielwarteschlange aufgelöst. Mit anderen Worten: der Name der physischen Warteschlange. Dieser Zielwarteschlangename 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 Definition-Objekt für remote Warteschlangen wird auf drei verschiedene Arten verwendet. [Tabelle 14 auf Seite 154](#) erläutert, wie Sie jede der drei Arten definieren:

- 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 Warteschlangename 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 Übertragungswarteschlangename angegeben wird, verwenden

det 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 Übertragungswarteschlangennamen 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 einreicht, 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.

Verwendung	Name des Warteschlangenmanagers	Warteschlangennamen	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 7](#) auf Seite 155 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 at QMB' auf. Beispiel: Diese Nachrichten werden unverändert in die Übertragungswarteschlange QMB gestellt.

Der Kanal verschiebt die Nachrichten in einen benachbarten WS-Manager.

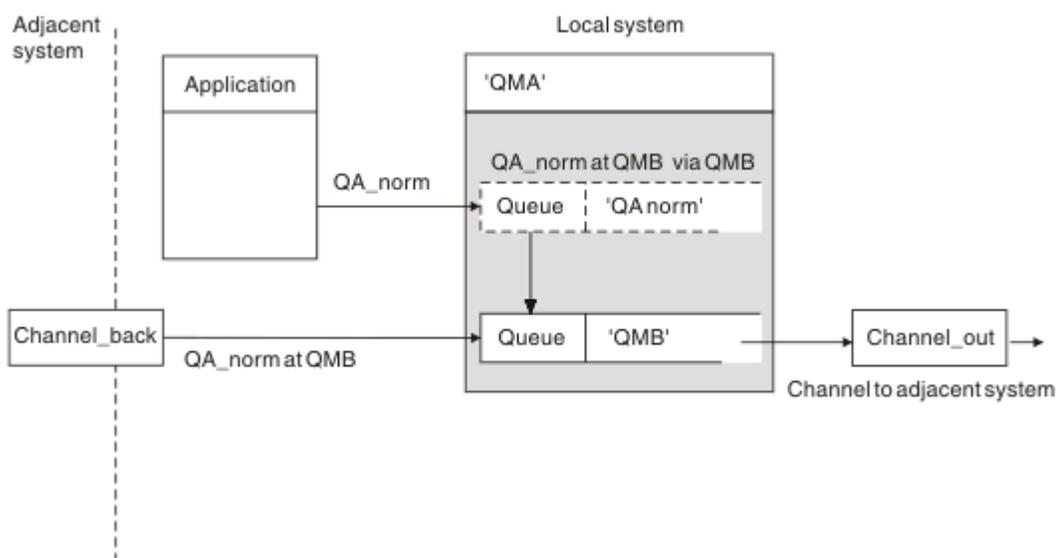


Abbildung 7. Eine Definition einer fernen Warteschlange wird verwendet, um einen Warteschlangennamen in eine Übertragungswarteschlange in einen benachbarten WS-Manager aufzulösen.

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 ferne 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.

## Die Übertragungswarteschlange auswählen

Sie können eine ferne Warteschlangendefinition verwenden, um eine andere Übertragungswarteschlange zuzulassen, um Nachrichten an denselben benachbarten Warteschlangenmanager zu senden.

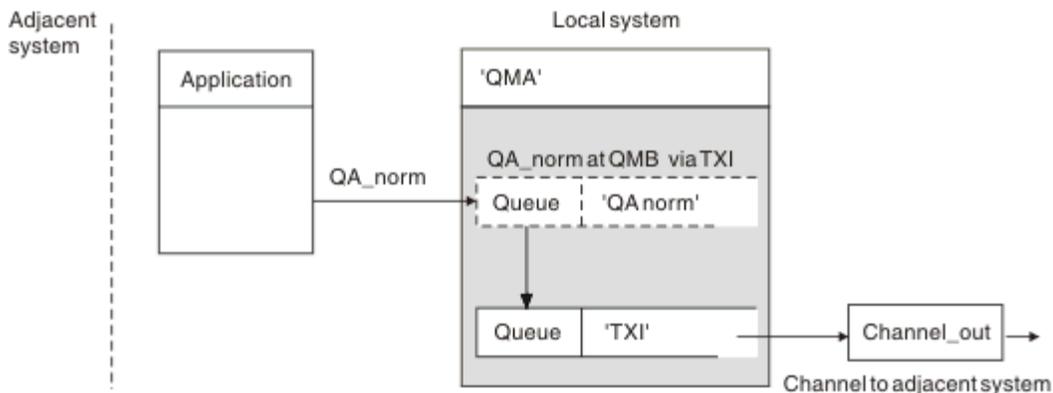


Abbildung 8. Die Definition der fernen Warteschlange ermöglicht, dass eine andere Übertragungswarteschlange verwendet wird.

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 7 auf Seite 155](#) in „Einreihen von Nachrichten in ferne Warteschlangen“ auf Seite 155. [Abbildung 8 auf Seite 156](#) in diesem Abschnitt zeigt, wie Sie die Definition der fernen Warteschlange verwenden, um Nachrichten über eine andere Übertragungswarteschlange und damit über einen anderen Kanal an denselben benachbarten Warteschlangenmanager zu senden.

Für die in [Abbildung 8 auf Seite 156](#) 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 in „[Clusterwarteschlangen definieren](#)“ auf Seite 256.

## Nachrichten empfangen

Sie können den Warteschlangenmanager für den Empfang von Nachrichten von anderen Warteschlangenmanagern konfigurieren. Sie müssen sicherstellen, dass eine unbeabsichtigte Namensauflösung nicht auftritt.

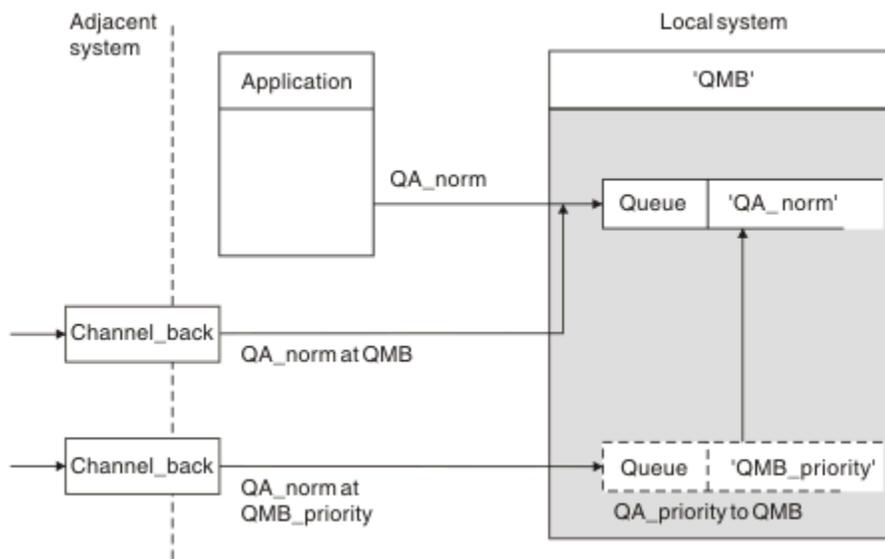


Abbildung 9. Nachrichten direkt empfangen und Aliasnamen-WS-Manager-Namen auflösen

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 9 auf Seite 157](#).

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.

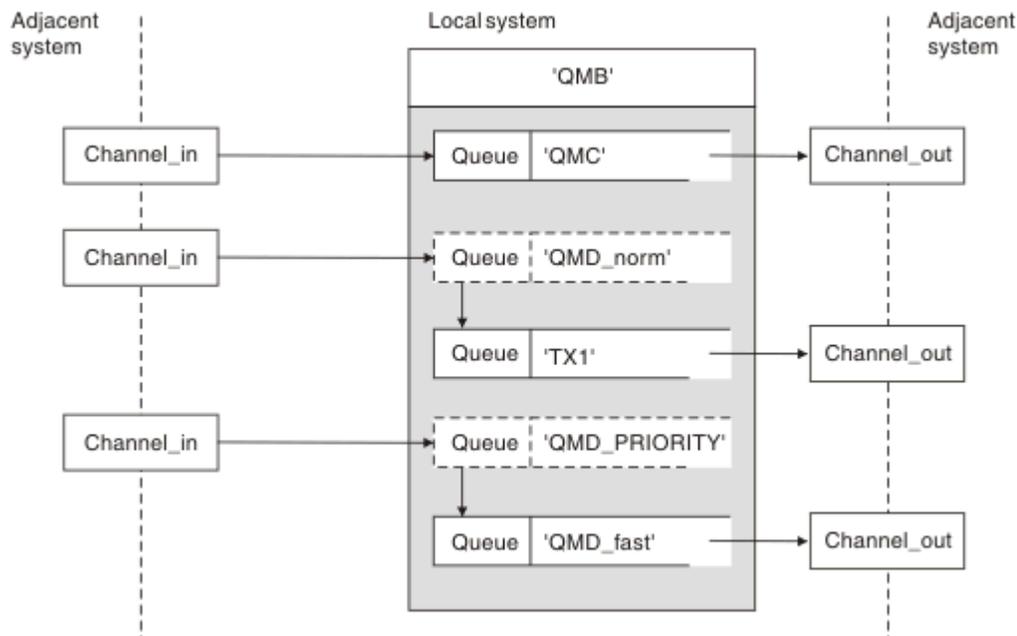


Abbildung 10. Drei Methoden zum Übergeben von Nachrichten durch Ihr System

Die in [Abbildung 9](#) auf Seite 157 in „Nachrichten empfangen“ auf Seite 156 gezeigte Technik hat gezeigt, wie ein Alias-Flow erfasst wird. [Abbildung 10](#) auf Seite 158 veranschaulicht die Art und Weise, wie Netzwerke 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.TRANS-MIT.QUEUE, alle Nachrichten an alle Warteschlangenmanager in allen Clustern, in denen der Warteschlangenmanager Mitglied ist. Informationen hierzu finden Sie im Artikel [Ein Cluster von WS-Managern](#). 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.

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. Beispiel:

- 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.

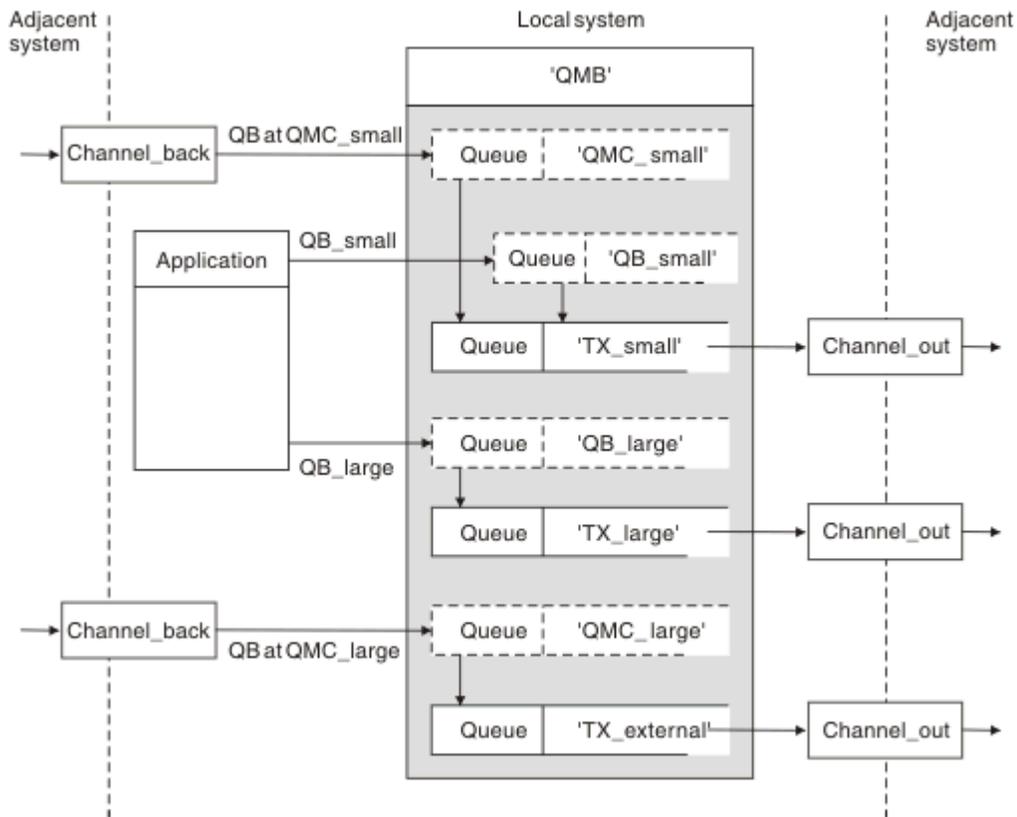


Abbildung 11. Nachrichtenflüsse trennen

In dem in [Abbildung 11](#) auf Seite 160 gezeigten Beispiel werden die beiden eingehenden Datenflüsse den Aliasnamen 'QMC\_small' und 'QMC\_large' des Aliaswarteschlangenmanagers 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.

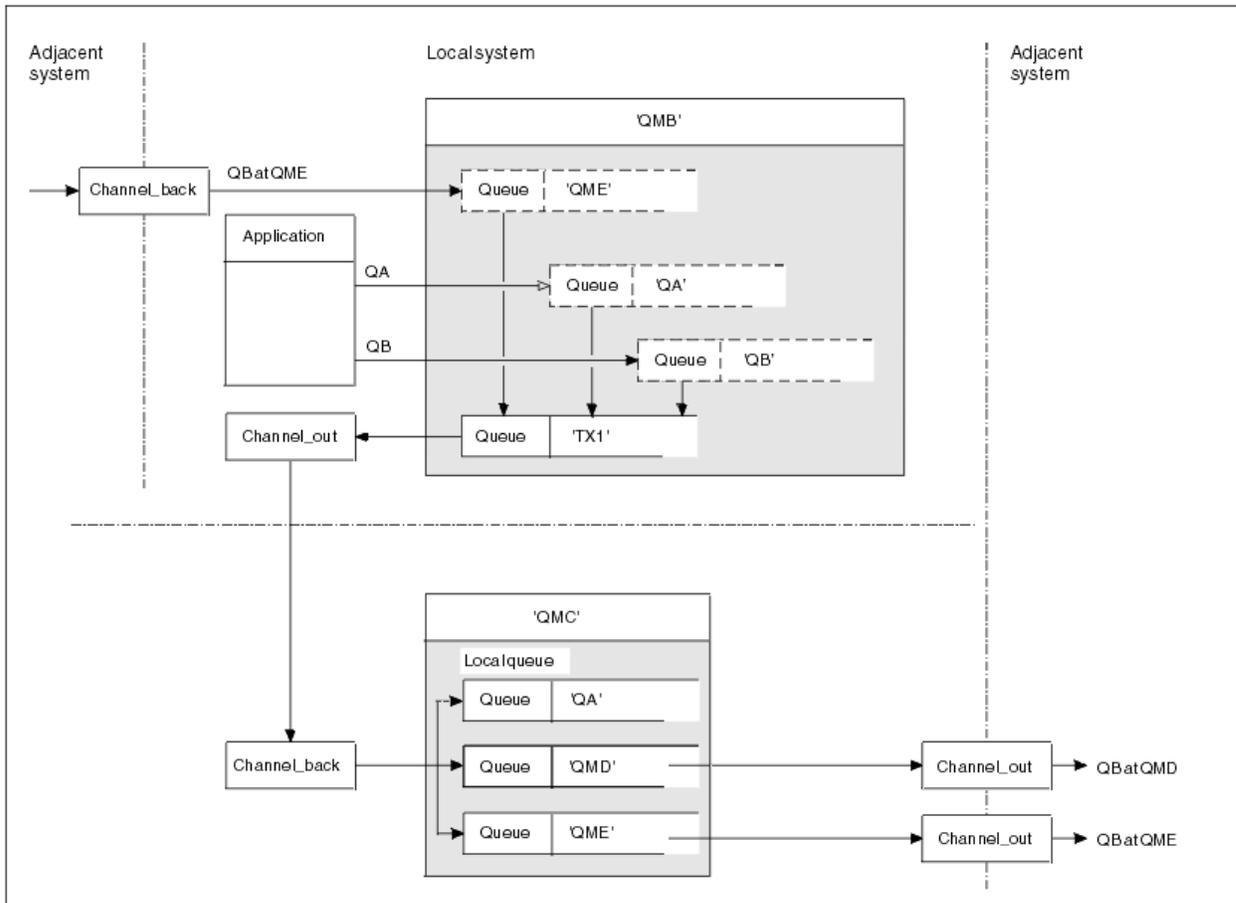


Abbildung 12. Kombinieren von Nachrichtenflüssen auf einem Kanal

Abbildung 12 auf Seite 161 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.

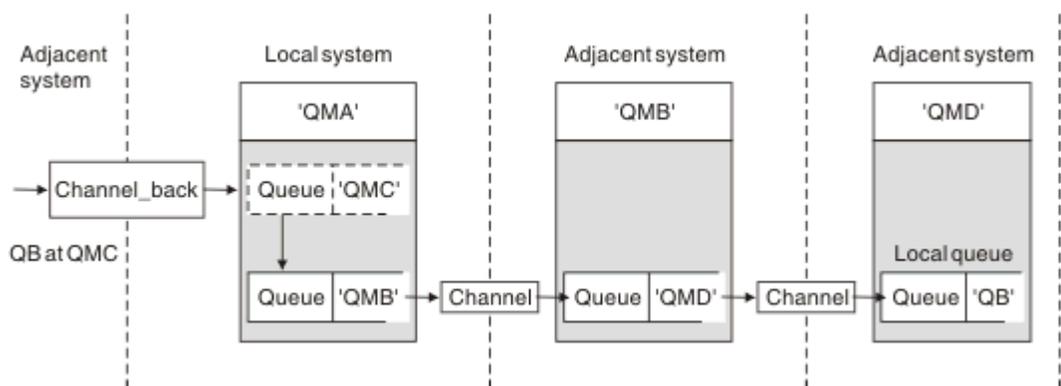


Abbildung 13. Nachrichtenströme zu einem anderen Ziel verwählen

In [Abbildung 13](#) auf Seite 162 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 355.

## 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.

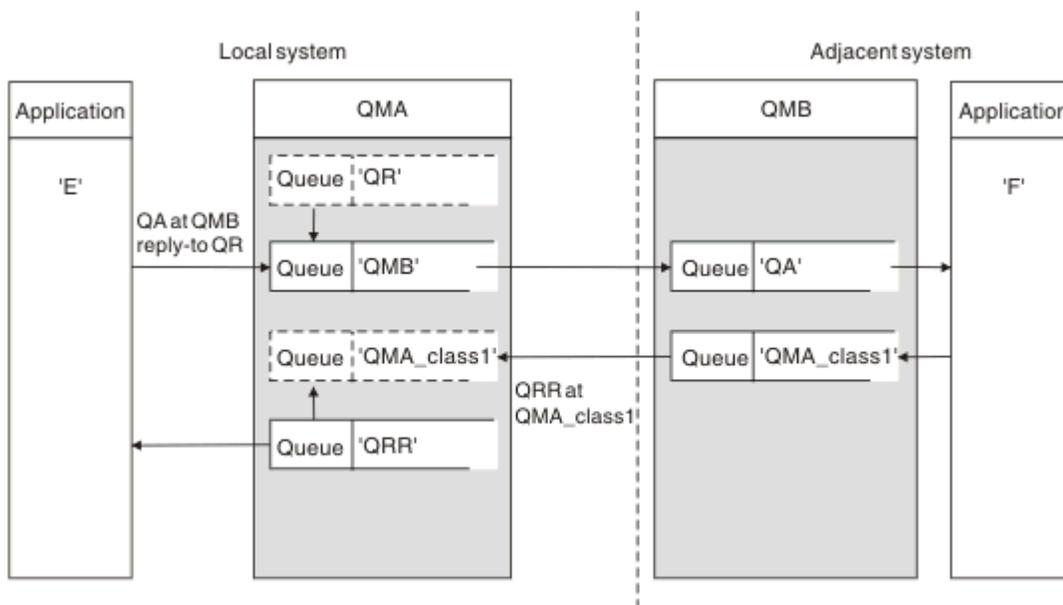


Abbildung 14. Name der Warteschlange für Antwortwarteschlangen beim PUT-Aufruf

In [Abbildung 14 auf Seite 163](#) 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 18 auf Seite 172](#) 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 ändern, so dass die Abläufe getrennt werden, ohne dass die Anwendung einbezogen wird. Die Anwendung wählt immer QR für diese bestimmte Serviceklasse aus. Sie haben die Möglichkeit, die Serviceklasse mit der Antwortwarteschlange-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 Sendechannel
- 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 153  
In diesem Thema werden die drei Möglichkeiten erläutert, wie Sie eine Definition einer fernen Warteschlange erstellen können.

„Beispiel für Antwortwarteschlangenalias“ auf Seite 165

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 167

Eine Erläuterung des Beispiels und die Verwendung des Aliasnamens der Empfangswarteschlange für Antworten durch den Warteschlangenmanager.

„Durchlauf der Warteschlange für Antwortwarteschlangen“ auf Seite 168

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 15 auf Seite 166 gezeigt, muss die Rückgaberoute für die Antwortnachrichten, einschließlich der Übertragungswarteschlange, des Kanals und des Aliasnamens des Warteschlangenmanagers, verfügbar sein.

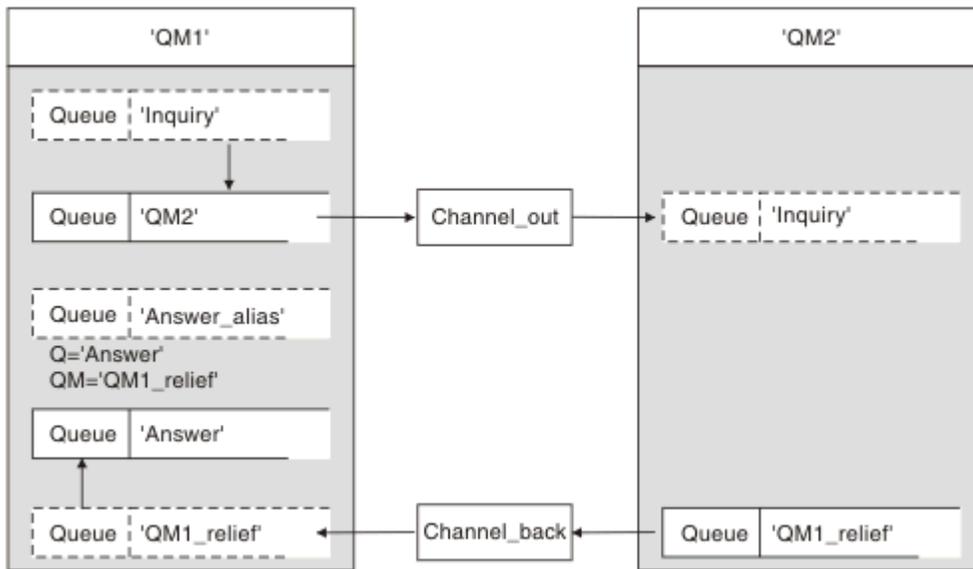


Abbildung 15. Beispiel für Antwortwarteschlangenalias

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.

Object	Definition								
Lokale Übertragungswarteschlange	QM2								
Definition der fernen Warteschlange	<table border="0"> <tr> <td>Objektname</td> <td>Anfrage</td> </tr> <tr> <td>Name des fernen Warteschlangenmanagers</td> <td>QM2</td> </tr> <tr> <td>Name der fernen Warteschlange</td> <td>Anfrage</td> </tr> <tr> <td>Name der Übertragungswarteschlange</td> <td>WSM2 (DEFAULT)</td> </tr> </table>	Objektname	Anfrage	Name des fernen Warteschlangenmanagers	QM2	Name der fernen Warteschlange	Anfrage	Name der Übertragungswarteschlange	WSM2 (DEFAULT)
Objektname	Anfrage								
Name des fernen Warteschlangenmanagers	QM2								
Name der fernen Warteschlange	Anfrage								
Name der Übertragungswarteschlange	WSM2 (DEFAULT)								

<b>Object</b>	<b>Definition</b>	
Aliasname des WS-Managers	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.

<b>Feld</b>	<b>Inhalt</b>
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.

<b>Object</b>	<b>Definition</b>
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.

<b>Feld</b>	<b>Inhalt</b>
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 15. 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 (Warteschlangename = '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 einreicht, 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:

<b>Name der Empfangswarteschlange für Antworten</b>	<b>Antworten</b>
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 Warteschlangename 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](#).

 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 unbekanntes 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 16 auf Seite 171](#), [Tabelle 17 auf Seite 171](#) und [Tabelle 18 auf Seite 172](#).

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 16. Warteschlangennamensauflösung auf WS-Manager QMA*

Quellenwarteschlange angegeben, wenn Warteschlange geöffnet wird	Der Quellenwarteschlangenmanager wurde beim Öffnen der Warteschlange angegeben.	Aufgelöster Warteschlangennamen	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 WS-Managers

*Tabelle 17. Warteschlangennamensauflösung auf WS-Manager WSMB*

Quellenwarteschlange angegeben, wenn Warteschlange geöffnet wird	Der Quellenwarteschlangenmanager wurde beim Öffnen der Warteschlange angegeben.	Aufgelöster Warteschlangennamen	Aufgelöster Name des Warteschlangenmanagers	Name der aufgelösten Übertragungswarteschlange	Auflösungstyp
QA_norm	-	QA_norm	WSMB	-	(keine)
QA_norm	WSMB	QA_norm	WSMB	-	(keine)

Tabelle 17. 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	QMB_PRIORITÄT	QA_norm	WSMB	-	Aliasname des WS-Managers
(beliebig)	QMC	(beliebig)	QMC	QMC	(keine)
(beliebig)	QMD_norm	(beliebig)	QMD_norm	TX1	Aliasname des WS-Managers
(beliebig)	QMD_PRIORITÄT	(beliebig)	QMD_PRIORITÄT	QMD_schnell	Aliasname des WS-Managers
(beliebig)	QMC_small	(beliebig)	QMC_small	TX_klein	Aliasname des WS-Managers
(beliebig)	QMC_large	(beliebig)	QMC_large	TX_extern	Aliasname des WS-Managers
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 WS-Managers
QA	QMC	QA	QMC	TX1	Ferne Warteschlange
QB	WSMD	QB	WSMD	TX1	Ferne Warteschlange

Tabelle 18. 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.

Die Ausgabe eines Befehls RESET CHANNEL führt nicht zu Verlust oder Duplizierung von Nachrichten. Der Befehl RESET bestätigt die Warnung von IBM MQ, dass etwas nicht richtig zu sein scheint. Ein

unbestätigter Kanal, der den persistenten Status verloren hat, kann nach einem RESET-Befehl weiterhin nicht gestartet werden, bis Sie einen Befehl RESOLVE CHANNEL absetzen. Dies ist die Aktion, die einen Stapel verlieren oder duplizieren kann.

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 einreicht, 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 eingestellt 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:

-  „Kanäle unter UNIX, Linux, and Windows überwachen und steuern“ auf Seite 208
-  „Kanäle unter z/OS überwachen und steuern“ auf Seite 776
-  „Kanäle in IBM i überwachen und steuern“ auf Seite 233

Informationen zu den Planungsbeispielen für Nachrichtenkanäle finden Sie unter:

-  [Beispiel für Nachrichtenkanalplanung für UNIX, Linux, and Windows](#)
-  [Beispiel für Nachrichtenkanalplanung für IBM i](#)
-  [Beispiel für Nachrichtenkanalplanung für 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 175

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 183

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 198

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 204

Die Verarbeitung von Kanalinitialisierungsdaten hängt von Ihrer IBM MQ-Plattform ab.

„Datenkonvertierung“ auf Seite 205

IBM MQ-Nachrichten erfordern möglicherweise eine Datenkonvertierung, wenn sie zwischen Warteschlangen in verschiedenen Warteschlangenmanagern gesendet werden.

„Schreiben eigener Nachrichtenkanalagenten“ auf Seite 205

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 206

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 Informationen**

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.

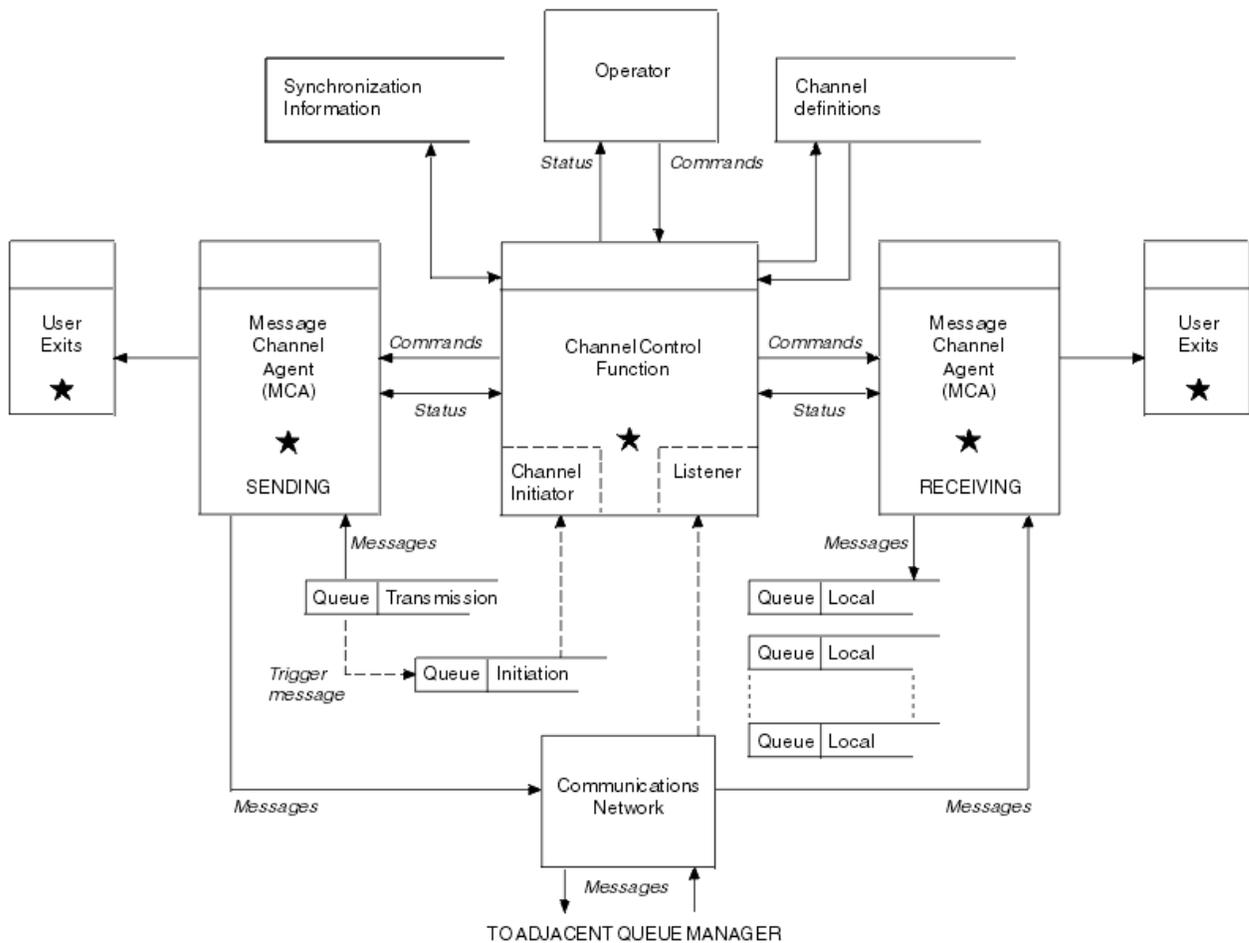


Abbildung 16. Verteiltes Warteschlangenmanagementmodell

**Anmerkung:**

1. Abhängig von der Plattform gibt es je einen MCA pro Kanal. Es können eine oder mehrere Kanalsteuerfunktionen für einen bestimmten Warteschlangenmanager vorhanden sein.
2. Die Implementierung von MCAs und Kanalsteuerfunktionen 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 ausgewähltes 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 (  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 152

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 174

DQM (Distributed Queue Management) wird zum Definieren und Steuern der Kommunikation zwischen Warteschlangenmanagern verwendet.

[„Ausgelöste Kanäle“](#) auf Seite 200

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 198

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.

[„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“](#) auf Seite 773

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.

### Zugehörige Tasks

[„Warteschlangenmanager auf Multiplatforms erstellen und verwalten“](#) auf Seite 5

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 unter UNIX, Linux, and Windows überwachen und steuern“](#) auf Seite 208

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 233

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 dem Server und dem Client konfigurieren“](#) auf Seite 17

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 255

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.

## **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 für lokale 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 180](#)

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 181](#)

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 17 auf Seite 181](#).

MQM-Warteschlange erstellen (CRTMQMQ)

Auswahl eingeben und Eingabetaste drücken.

Warteschlangenname.....

Warteschlangentyp..... \_\_\_\_ \*ALS, \*LCL, \*MDL, \*RMT

Name des Nachrichten-WS-Managers... \*DFT\_\_\_\_\_

-----

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige

F24 = Weitere Tasten

+

*Abbildung 17. 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 18 auf Seite 182](#)).

MQM-Warteschlange erstellen (CRTMQMQ)

Auswahl eingeben und Eingabetaste drücken.

```
Warteschlangenname..... > HURS.2.HURS.PRIORIT
Warteschlangentyp..... > *LCL *ALS, *LCL, *MDL, *RMT
Name des Nachrichten-WS-Managers... *DFT
Ersetzen..... *NO *NO, *YES
Text 'Beschreibung'..... ''
Put aktiviert..... *YES *SYSDFTQ, *NO, *YES
Standardnachrichtenpriorität... 0 0-9, *SYSDFTQ
Standardnachrichtenpersistenz.. *NO *SYSDFTQ, *NO, *YES
Prozessname..... ''
Auslösung aktiviert..... *NO *SYSDFTQ, *NO, *YES
Abrufen aktiviert..... *YES *SYSDFTQ, *NO, *YES
Freigabe aktiviert..... *YES *SYSDFTQ, *NO, *YES
```

Mehr ...

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

*Abbildung 18. 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 19 auf Seite 182](#)).

MQM-Warteschlange erstellen (CRTMQMQ)

Auswahl eingeben und Eingabetaste drücken.

```
Standardfreigabeoption..... *YES *SYSDFTQ, *NO, *YES
Nachrichtenübermittlungsfolge... *PTY *SYSDFTQ, *PTY, *FIFO
Anzahl der Ausrückungsrückstand.... *NO *SYSDFTQ, *NO, *YES
Auslösertyp..... *FIRST *SYSDFTQ, *FIRST, *ALL ...
Auslösertiefe..... 1 1-999999999, *SYSDFTQ
Auslösernachrichtenpriorität... 0 0-9, *SYSDFTQ
Auslöserdaten..... ''
Aufbewahrungsintervall..... 999999999 0-999999999, *SYSDFTQ
Maximale Warteschlangenlänge..... 5000 1-24000, *SYSDFTQ
Maximale Nachrichtenlänge..... 4194304 0-4194304, *SYSDFTQ
Rücksetzschwellenwert..... 0 0-999999999, *SYSDFTQ
Warteschlange für Zurücksetzungswarteschlange.... ''
Initialisierungswarteschlange..... ''
```

Mehr ...

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

*Abbildung 19. 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.

MQM-Warteschlange erstellen (CRTMQMQ)

Auswahl eingeben und Eingabetaste drücken.

```
Verwendung..... *TMQ *SYSDFTQ, *NORMAL, *TMQ
Der obere Schwellenwert für die Warteschlangenlänge... 80 0-100, *SYSDFTQ
Der untere Schwellenwert für die Warteschlangenlänge... 20 0-100, *SYSDFTQ
Vollständige Warteschlangenereignisse aktiviert... *YES *SYSDFTQ, *NO, *YES
Hohe Warteschlangenereignisse aktiviert... *YES *SYSDFTQ, *NO, *YES
Niedrige Warteschlangenereignisse aktiviert... *YES *SYSDFTQ, *NO, *YES
Serviceintervall..... 99999999 0-99999999, *SYSDFTQ
Serviceintervallereignisse... *NONE *SYSDFTQ, *HIGH, *OK, *NONE
Unterstützung für Verteilerlisten... *NO *SYSDFTQ, *NO, *YES
Clustername..... *SYSDFTQ
Clusternamensliste..... *SYSDFTQ
Standardbindung..... *SYSDFTQ *SYSDFTQ, *OPEN, *NOTFIXED
```

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

*Abbildung 20. 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 Kanalen 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 assemblerassembliert. 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 unter UNIX, Linux, and Windows überwachen und steuern“ auf Seite 208 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)
- Nachrichtenfolgennummern zurücksetzen
- Auflösen eines unbestätigten Nachrichtenstaus 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

Weitere Informationen zum Definieren, Steuern und Überwachen von Kanälen finden Sie in den folgenden Unterabschnitten:

### ***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 Kanaldefinition 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 151 beschriebenen Szenarios zu implementieren.

Informationen zum Definieren von MQI-Kanälen finden Sie in [„Definieren von MQI-Kanälen“](#) auf Seite 32.

## 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 unbestätigten Fall an der sendenden Seite hinterlässt. Weitere Informationen finden Sie in [„Unbestätigte Kanäle“](#) auf Seite 196.

## 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 21 auf Seite 186 zeigt die Hierarchie aller möglichen Kanalstatus und die Unterzustände, die für jeden Kanalstatus gelten.

Abbildung 22 auf Seite 187 zeigt die Links zwischen Kanalstatus. Diese Links gelten für alle Arten von Nachrichtenkanal- und Serververbindungskanälen.

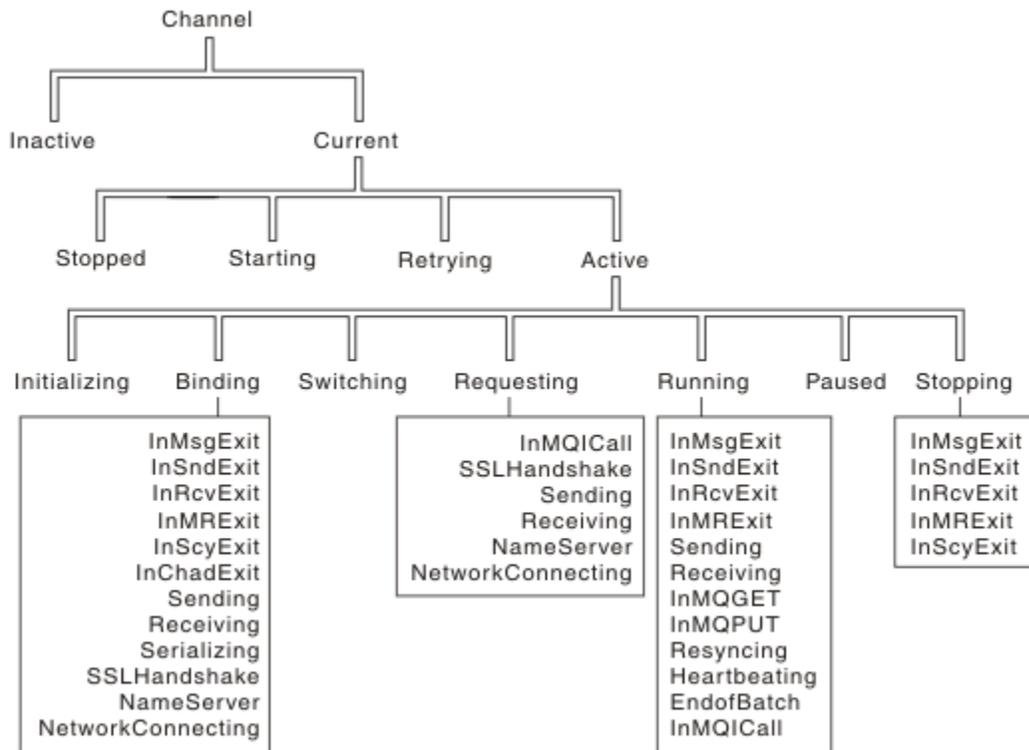


Abbildung 21. Kanalstatus und Unterzustände

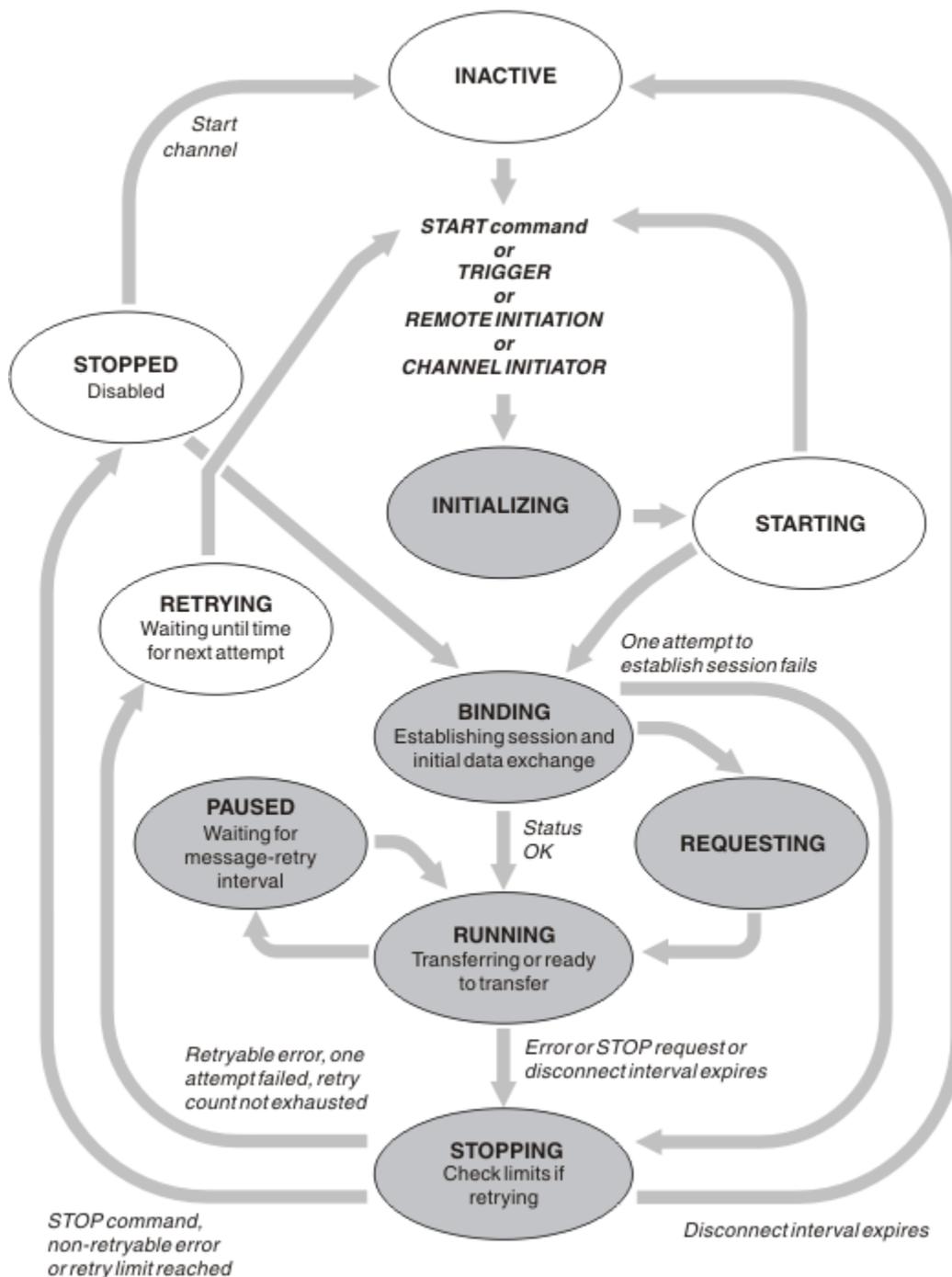


Abbildung 22. Flows zwischen Kanalstatus

### 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 22](#) auf Seite 187 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 21](#) auf Seite 186 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 RE TRYING, 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.

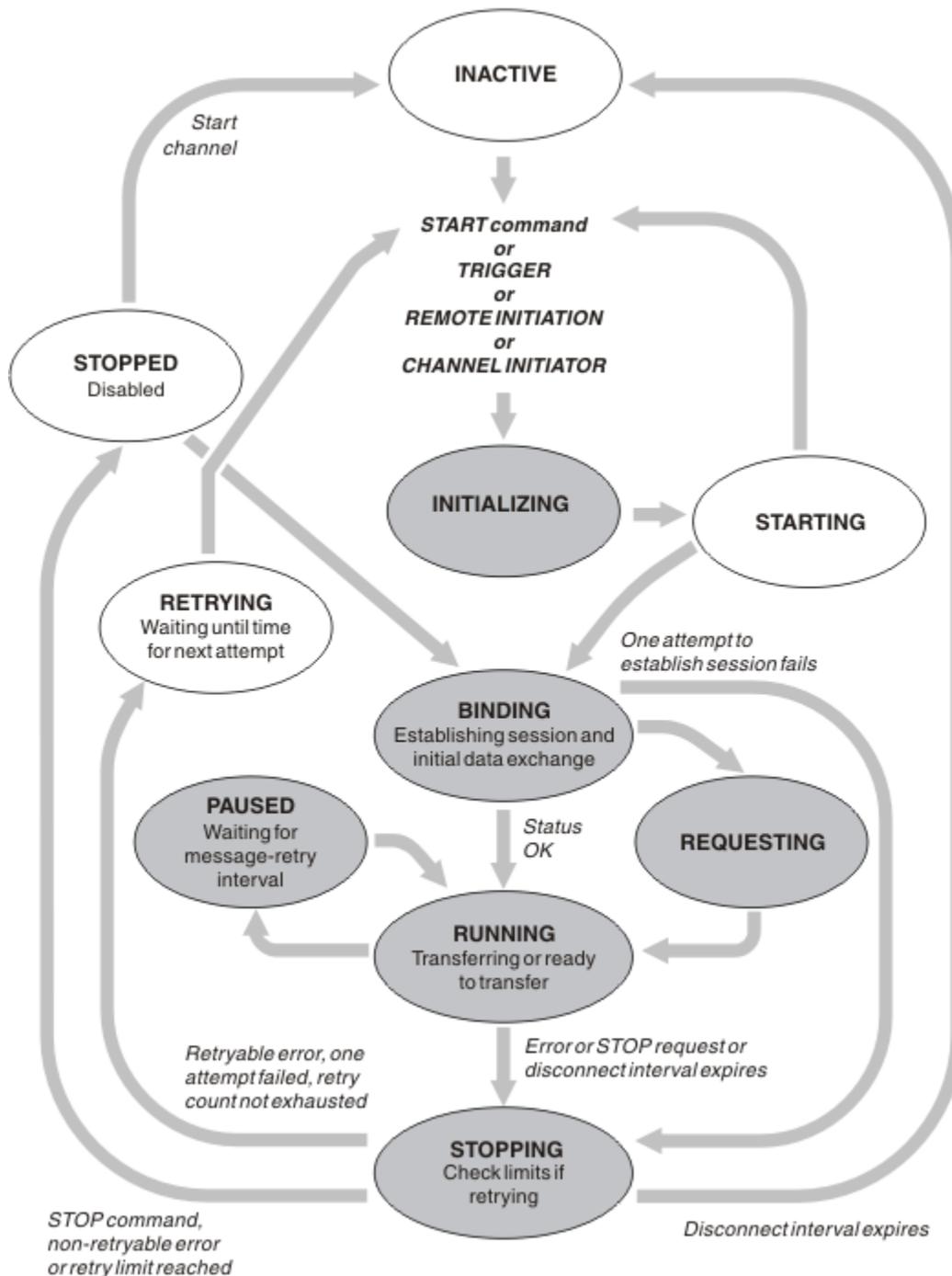


Abbildung 23. Flows zwischen Kanalstatus

### Anmerkung:

1. Wenn sich ein Kanal in einem der sechs in Abbildung 23 auf Seite 188 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** ▶ **UNIX** Bearbeiten Sie die Konfigurationsdatei des Warteschlangenmanagers.
- Verwenden Sie 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:

- ▶ **ULW** [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  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:

-  [IBM MQ verwalten](#)
-  [IBM MQ for IBM i verwalten](#)
-  [IBM MQ for z/OS verwalten](#)

### Kanalfehler

Fehler auf Kanälen bewirken, dass der Kanal weitere Übertragungen stoppen kann. 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 REPLY 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 Bedieneingriffe, bevor sie erneut gestartet werden können (siehe „Gestoppte Kanäle erneut starten“ auf Seite 195).

**Anmerkung:** Für  IBM i, UNIX, 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 gestoppter 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.

 Wenn der Kanalinitiator gestoppt wird, während sich ein Kanal im Status REPLYING 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.

 Wenn der Warteschlangenmanager gestoppt wird, während sich ein Kanal im Status REPLYING 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 und Kanalexitprogramme für Messaging-Kanäle finden Sie unter [Kanalattribute](#) . Informationen zum Nachrichtenwiederholungsexit finden Sie im Abschnitt.

### ***Grenzwerte für Serververbindungskanäle***

Sie können Grenzwerte für Serververbindungskanäle festlegen, um zu verhindern, dass Clientanwendungen die Kanalressourcen des Warteschlangenmanagers (**MAXINST**) ausschöpfen, und um zu verhindern, dass eine einzelne Clientanwendung die Kapazität des Serververbindungskanals (**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 Informationen**

[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

Wenn Sie in IBM MQ for z/OS TCP/IP als Transportprotokoll verwenden, können Sie auch einen Wert für das Attribut "**Keepalive** interval channel" (KAIN) angeben. Es wird empfohlen, dem **Keepalive**-Intervall einen höheren Wert als das Intervall der Überwachungssignale zu geben und einen kleineren Wert als den Unterbrechungswert zu verwenden. Sie können dieses Attribut verwenden, um einen Zeitüberschreitungswert für jeden Kanal anzugeben, wie in [Keepalive Interval \(KAIN\)](#) beschrieben.

Wenn Sie in IBM MQ for IBM i- und UNIX, Linux, and Windows -Systemen TCP als Transportprotokoll verwenden, können Sie `keepalive=yes` festlegen. 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 \(KAIN\)](#) 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** enthält der Parameter [KAIN](#) 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 *Zeitlimitüberschreitung bei Empfang*, wird gemäß dem Wert von HBINT (Überwachungssignalintervall) bestimmt.

In IBM MQ for IBM i- und UNIX, Linux, and Windows -Systemen wird der Wert für *Empfangszeitlimit* wie folgt festgelegt:

1. Für eine anfängliche Anzahl von Datenflüssen, bevor eine Vereinbarung stattfindet, ist der Wert *receive time-out* doppelt so oft wie der Wert für HBINT aus der Kanaldefinition.
2. Wenn die Kanäle einen HBINT-Wert aushandeln, wenn HBINT auf weniger als 60 Sekunden gesetzt ist, wird der Wert von *receive time-out* auf das Doppelte dieses Werts gesetzt. Wenn HBINT auf 60 Sekunden oder mehr gesetzt ist, wird der Wert von *receive time-out* auf 60 Sekunden größer als der Wert von HBINT gesetzt.

In IBM MQ for z/OS wird der Wert *Zeitlimitüberschreitung bei Empfang* wie folgt eingestellt:

1. Für eine anfängliche Anzahl von Datenflüssen, bevor eine Vereinbarung stattfindet, ist der Wert *receive time-out* doppelt so oft wie der Wert für HBINT aus der Kanaldefinition.
2. Wenn RCVTIME festgelegt ist, wird das Zeitlimit auf einen der folgenden Wert gesetzt:
  - das ausgehandelte HBINT multipliziert mit einer Konstanten
  - das vereinbarte HBINT plus eine konstante Anzahl von Sekunden
  - eine konstante Anzahl von Sekunden

in Abhängigkeit vom Parameter RCVTTYPE und vorbehaltlich der von RCVTMIN vorgegebenen Grenzen, wenn dies zutrifft. RCVTMIN wird nicht angewendet, wenn RCVTTYPE (EQUAL) konfiguriert ist. Wenn Sie einen konstanten Wert von RCVTIME verwenden und ein Überwachungssignalintervall verwenden, geben Sie keine RCVTIME an, die kleiner als das Intervall der Überwachungssignale ist.

Ausführliche Informationen zu den Attributen RCVTIME, RCVTMIN und RCVTTYTYPE finden Sie im Befehl [ALTER QMGR](#).

### Anmerkung:

1. Wenn einer der Werte null ist, gibt es kein Zeitlimit.
2. Bei Verbindungen, die keine Überwachungssignale unterstützen, wird der Wert von HBINT in Schritt 2 mit null ausgehandelt, und daher gibt es kein Zeitlimit. Daher müssen Sie TCP/IP KEEPALIVE verwenden.
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 Festbeschreibung 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. -er., dass der Client während aller MQI-Aufrufe Überwachungssignale (und macht) kann.

5. In IBM WebSphere MQ 7-Clientkanälen können Überwachungssignale sowohl von der Server- als auch der Clientseite fließen. Das Zeitlimit an beiden Enden basiert auf  $2 * HBINT$  für HBINTs von weniger als 60 Sekunden und  $HBINT + 60$  für HBINTs von mehr als 60 Sekunden.
6. 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 ist und ein Sende- oder Empfangsexit verwendet wird, wartet das empfangende Ende 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

### IBM MQ for z/OS

Als Startpunkt können Sie Folgendes verwenden:

```
/cpl ALTER QMGR TCPKEEP(YES) RCVTTYTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

Dabei steht `cpI` 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 Absenders in mehr als eine Adressen übersetzt werden kann, müssen Sie ADOPTCHK möglicherweise auf QMNAME anstelle von ALL setzen.

### 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 für Konfigurationsdateien für verteilte Steuerung von Warteschlangen](#) und [„Attribute der Zeilengruppe 'channels'“](#) auf Seite 131.

Wenn die IP-Adresse des Senders in mehrere Adressen übersetzt werden kann, müssen Sie möglicherweise **AdoptNewMCACheck** auf QMName anstelle von ALLsetzen.

### **MCA-Adopting**

Die Funktion "Adopt MCA" ermöglicht es IBM MQ, einen Empfängerkanal abubrechen 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 Systemkeepalives zu beheben.

Mit der Funktion "Adopt MCA" wird das Problem automatisch gelöst. Sie ermöglicht es IBM MQ, einen Empfängerkanal abubrechen und an seiner Stelle einen neuen zu starten.

### **Zugehörige Informationen**

[IBM MQ verwalten](#)

[IBM MQ for z/OS verwalten](#)

[IBM MQ for IBM i verwalten](#)

### **Kanäle stoppen und in den Quiescemode versetzt**

Sie können einen Kanal stoppen und in den Quiescemode versetzt, bevor das Unterbrechungszeitintervall abläuft.

Nachrichtenkanäle sind so konzipiert, dass sie lange Verbindungen zwischen Warteschlangenmanagern mit ordnungsgemäße gesteuerte 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
-   andere plattformspezifische Mechanismen, wie folgt:

 **z/OS:**

Die Anzeige 'Kanal stoppen'

 **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 Nachrichtenneuordnung, 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 Nachrichtenneuordnung, 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 sendenden 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.

 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.

 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](#)
  -  Unter z/OS über die Anzeige [Kanal starten](#).
  -  Unter IBM i entweder über die Befehlszeile mit dem Befehl [STRMQMCHL](#) oder mit der Option START in der Anzeige [WRKMQMCHL](#).

## Unbestätigte Kanäle

Ein unbestätigter Kanal ist ein Kanal, der sich im Zweifel mit einem fernen Kanal befindet, über den Nachrichten gesendet und empfangen wurden.

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, einen Kanal in Frage zu stellen, durch die Verwendung des Kanalparameters 'Batch Heartbeat' (BATCHHB) verringern. Wenn ein Wert für diesen Parameter angegeben wird, überprüft ein Senderkanal, ob der ferne Kanal noch aktiv ist, bevor er weitere Maßnahmen ergreift. 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 Nachrichtenstapel im Sender in Zweifel steht, wenn der Empfangsstatus unbekannt ist, wird die Situation behoben, wenn die Kommunikation erneut aufgebaut wird. Die Folgennummer und die LUWID-Datensätze werden zu diesem Zweck aufbewahrt. Der Kanal steht im Zweifel, bis LUWID-Informationen ausgetauscht wurden und nur ein Nachrichtenstapel für den Kanal unbestätigt sein kann.

Sie können den Kanal bei Bedarf manuell resynchronisieren. Der Begriff *manual* 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.

1. Verwenden Sie den Befehl [DISPLAY CHSTATUS](#), um die ID der letzten logischen Arbeitseinheit (LUWID) für **jede** Seite des Kanals zu finden. Verwenden Sie dazu die folgenden Befehle:

- Für die unbestätigte Seite des Kanals:

```
DISPLAY CHSTATUS( name ) SAVED CURLUWID
```

Sie können die Parameter CONNAME und XMITQ verwenden, um den Kanal weiter zu identifizieren.

- Für die Empfängerseite des Kanals:

```
DISPLAY CHSTATUS( name ) SAVED LSTLUWID
```

Sie können den Parameter CONNAME verwenden, um den Kanal weiter zu identifizieren.

Die Befehle sind unterschiedlich, da nur die sendende Seite des Kanals unbestätigt sein kann. Die empfangende Seite ist nie im Zweifel.

Unter IBM MQ for IBM i kann der Befehl DISPLAY CHSTATUS aus einer Datei heraus mit dem Befehl STRMQMMQSC ausgeführt werden oder mit dem CL-Befehl WRKMQMCHST ('mit MQM-Kanalstatus arbeiten').

2. 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 unbestätigten Nachrichten aus der Übertragungswarteschlange entfernen und sie erneut aktivieren. Dies wird mit dem folgenden Kanal RESOLVE ausgeführt:

```
RESOLVE CHANNEL( name ) ACTION(COMMIT)
```

3. 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 unbestätigten Nachrichten in der Übertragungswarteschlange aufbewahren und sie erneut senden. Dies wird mit dem folgenden Kanal RESOLVE ausgeführt:

```
RESOLVE CHANNEL( name ) ACTION(BACKOUT)
```

 Unter IBM MQ for IBM i können Sie den Befehl "Resolve MQM Channel", RSVMQMCHL, verwenden.

Wenn dieser Prozess abgeschlossen ist, ist der Kanal nicht mehr im Zweifel. Die Übertragungswarteschlange kann jetzt, falls erforderlich, von einem anderen Kanal verwendet werden.

## Fehlerbestimmung

Es gibt zwei unterschiedliche Aspekte bei der Fehlerbestimmung: Probleme, die bei der Übergabe eines Befehls erkannt werden, und Probleme, die während des Betriebs der Kanäle erkannt werden.

## Befehlsüberprüfung

Befehle und Anzeigendaten müssen von Fehlern frei sein, bevor sie zur Verarbeitung akzeptiert werden. Alle Fehler, die von der Validierung festgestellt werden, werden dem Benutzer unverzüglich durch Fehlermeldungen mitgeteilt.

Die Problemdiagnose beginnt mit der Interpretation dieser Fehlermeldungen und ergreift Korrekturmaßnahmen.

## Fehler werden verarbeitet

Probleme, die während des normalen Betriebs der Kanäle gefunden werden, werden an die Systemkonsole oder das Systemprotokoll gemeldet. Die Problemdiagnose beginnt mit der Erfassung aller relevanten Informationen aus dem Protokoll und wird mit der Analyse fortgesetzt, um das Problem zu identifizieren.

Bestätigungs- und Fehlermeldungen werden an das Terminal zurückgegeben, das die Befehle nach Möglichkeit eingeleitet hat.

IBM MQ erstellt Buchhaltungs- und Statistikdaten, die Sie verwenden können, um Trends in Auslastung und Leistung zu identifizieren.  Auf Multiplatforms werden diese Informationen als PCF-Datensätze erstellt. Weitere Informationen finden Sie im Abschnitt [Strukturdatentypen](#).  Unter z/OS werden diese Informationen als SMF-Datensätze erstellt (siehe [Überwachung der Leistung und Ressourcennutzung](#)).

## Nachrichten und Codes

Informationen zu Nachrichten und Codes, die bei der primären Diagnose des Problems hilfreich sind, finden Sie unter [Nachrichten und Ursachencodes](#).

## 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 198.

## 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 24 auf Seite 199 zeigt die Verarbeitung an, die auftritt, wenn ein MCA keine Nachricht in die Zielwarteschlange einlegen kann. (Die angezeigten Optionen gelten nicht für alle Plattformen.)

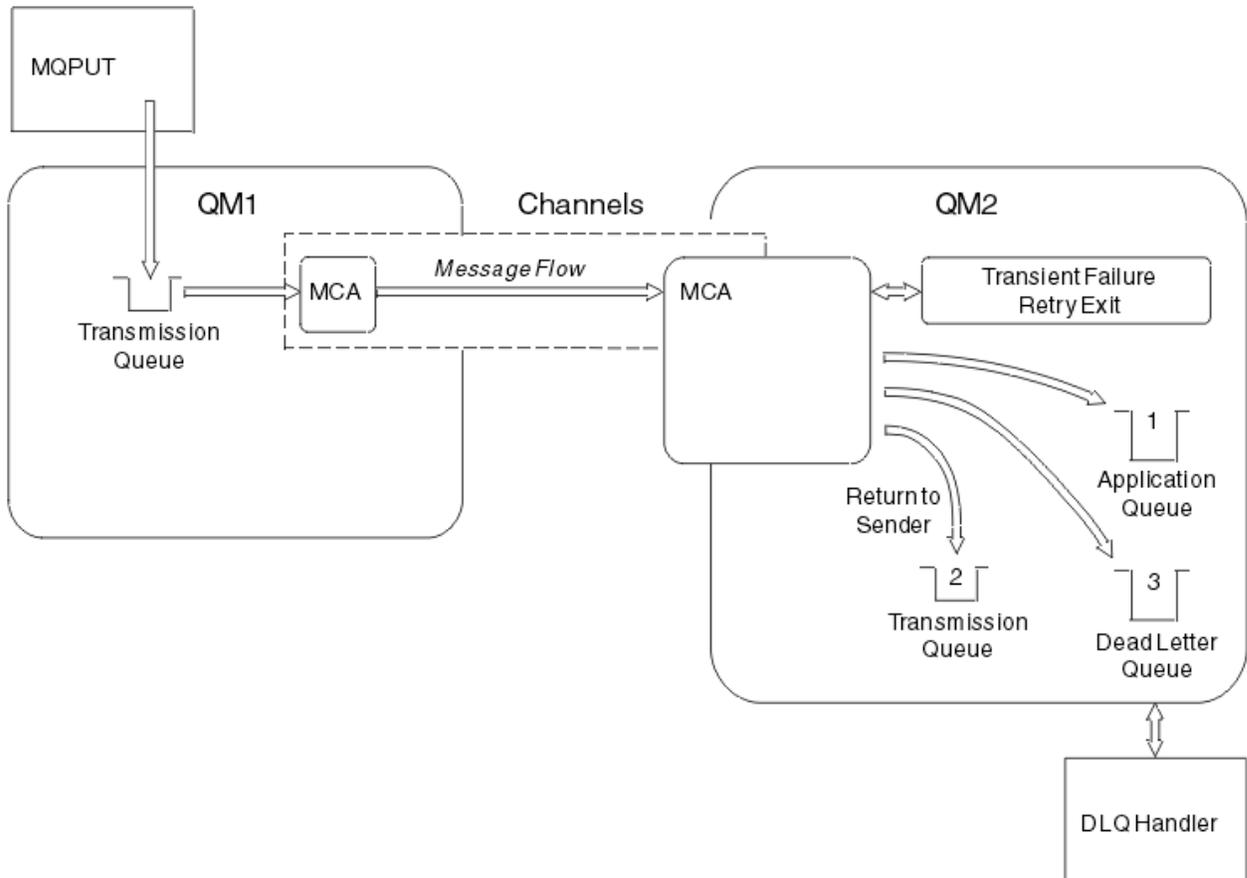


Abbildung 24. Was passiert, wenn eine Nachricht nicht zugestellt werden kann

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 Zeitpunkt 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 Nachrichtendescriptor 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 Ausnah-  
mebericht, der die ursprüngliche Nachricht enthält, und stellt ihn in eine Übertragungswarteschlange,  
die an die in der ursprünglichen Nachricht angegebene Warteschlange für Antwortnachrichten gesen-  
det 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 verarbei-  
ten. 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 [Steueroutine 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 send-  
enden 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 defi-  
niert ist, die ferne Warteschlange nicht erreichbar oder nicht definiert ist und keine ferne Warteschlan-  
ge für nicht zustellbare Nachrichten vorhanden ist, schaltet der Sendekanal in den RETRY-Modus um  
und Nachrichten werden automatisch in die Übertragungswarteschlange zurückgesetzt.

### Zugehörige Informationen

[Warteschlange für nicht zustellbare Mail verwenden \(USEDLQ\)](#)

### Ausgelöste Kanäle

IBM MQ stellt eine Funktion bereit, um eine Anwendung automatisch zu starten, wenn bestimmte Bedin-  
gungen 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 Windows siehe [UNIX and Linux-Systeme, „Kanäle unter UNIX, Linux, and Windows auslösen“ auf Seite 202](#)
-  Für IBM i siehe [„Kanäle in IBM MQ for IBM i auslösen“ auf Seite 202](#)
-  Für z/OS siehe [„Übertragungswarteschlangen und Auslöserkanäle“ auf Seite 775](#)

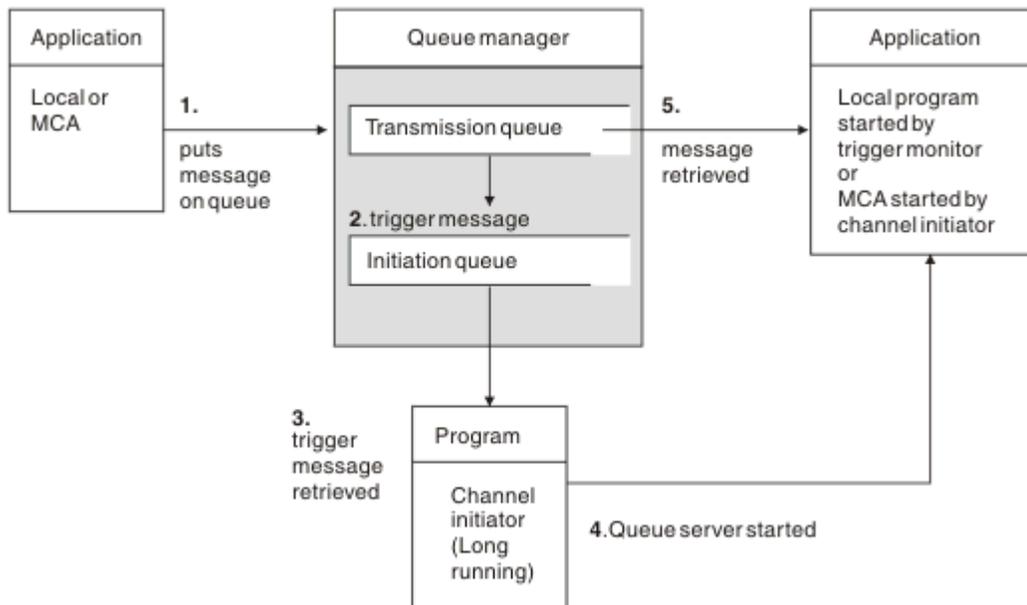


Abbildung 25. Die Konzepte des Triggerns

Die für die Auslösung erforderlichen Objekte werden in [Abbildung 25 auf Seite 201](#) 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, UNIX, 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 UNIX, 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)
```

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. Beispiel:

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(MYINITQ)
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

### Zugehörige Konzepte

„Kanalinitiator starten und stoppen“ auf Seite 203

Die Triggerung wird mithilfe des Kanalinitiatorprozesses implementiert.

### Zugehörige Tasks

„Verteilte Warteschlangensteuerung konfigurieren“ auf Seite 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### Zugehörige Informationen

[Kanalprogramme unter UNIX, Linux, and Windows](#)

 [Jobs für übergreifende Kommunikation unter 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 UNIX and Linux-Systeme und in der Registry für Windows-Systeme ändern.

Weitere Informationen zum Kanalinitiatorausführungsbefehl **runmqchi** und zu den anderen Steuerbefehlen finden Sie im Abschnitt [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.

### z/OS-Systeme



In IBM MQ for z/OS werden Initialisierungs- und Konfigurationsinformationen mit dem MQSC-Befehl ALTER QMGR angegeben. Wenn Sie ALTER QMGR-Befehle in die Initialisierungseingabedatei CSQINP2 eingeben, werden sie jedes Mal verarbeitet, wenn der Warteschlangenmanager gestartet wird.

Um MQSC-Befehle wie START LISTENER bei jedem Starten des Kanalinitiators auszuführen, setzen Sie sie in den Initialisierungseingabedatensatz CSQINPX und geben Sie die optionale DD-Anweisung CSQINPX in der Task-Prozedur für den Kanalinitiator an.

Weitere Informationen zu CSQINP2 und CSQINPX finden Sie im Abschnitt [Initialisierungseingabedaten-gruppen anpassen und ALTER QMGR](#).

### Windows-, IBM i-, UNIX and Linux-Systeme

In IBM MQ for Windows-,  IBM i-, UNIX and Linux-Systemen gibt es *Konfigurationsdateien*, um grundlegende Konfigurationsinformationen für die IBM MQ-Installation zu speichern.

Es gibt zwei Konfigurationsdateien: eine gilt für die Maschine, die andere für einen einzelnen WS-Manager.

#### IBM MQ-Konfigurationsdatei

Diese Datei enthält Informationen, die für alle WS-Manager auf dem IBM MQ-System relevant sind. Die Datei heißt `mqc.ini`. Eine vollständige Beschreibung hierzu finden Sie in [Verwalten für IBM MQ for Windows](#), und in [Verwaltung von IBM i](#) und in UNIX and Linux-Systemen.

#### Warteschlangenmanagerkonfigurationsdatei

Diese Datei enthält Konfigurationsdaten, die sich auf einen bestimmten Warteschlangenmanager beziehen. Die Datei heißt `qm.ini`.

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 Attribute "DefaultPath" müssten die WS-Manager-Konfigurationsdateien für einen Warteschlangenmanager mit dem Namen QMNAME beispielsweise wie folgt sein:

Für UNIX and Linux-Systeme:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Es folgt ein Auszug aus einer `qm.ini`-Datei. Sie gibt an, dass der TCP/IP-Listener an Port 2500 empfangsbereit sein soll, die maximale Anzahl der aktuellen Kanäle 200 beträgt und die maximale Anzahl aktiver Kanäle 100 sein soll.

```
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 Datei `qm.ini` zu verwenden, um den Anfang und das Ende eines Bereichs von Portwerten anzugeben. Das folgende Beispiel zeigt eine Datei `qm.ini`, 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.

 Für IBM i:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

Für Windows-Systeme:

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

Weitere Informationen zu `qm.ini`-Dateien finden Sie unter [Zeilengruppen für Konfigurationsdateien für die verteilte Steuerung von Warteschlangen](#).

## Datenkonvertierung

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 eigenen MCA-Programms 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 abrufen, 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, Nachrichtewarteschlangenschnittstelle) und nicht alle speziellen Schnittstellen.

Sie müssen sicherstellen, dass Nachrichten nur einmal zugestellt werden. Die Synchronisationspunkt-Koordination 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 eine 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 einreicht, 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 einreicht, 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.

 Ein DLQ-Handler wird mit IBM MQ auf UNIX and Linux-Systemen bereitgestellt. Weitere Informationen hierzu finden Sie im Abschnitt [The sample DLQ handler, amqsdq](#).

**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

MCA's 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`. 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 der 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:

<b>MQIBindType</b>	<b>Umgebungsvariable</b>	<b>Ergebnis</b>
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. Geben Sie MQIBindType=FASTPATH in `qm.ini` oder in der Registrierungsdatenbank an und geben Sie die Umgebungsvariable nicht an.
2. Geben Sie MQIBindType=FASTPATH in `qm.ini` oder Registry an und setzen Sie die 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).

## ULW Kanäle unter UNIX, 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 UNIX, 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, Kanalfolgennummern 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 in [IBM MQ-Steuerbefehlsreferenz](#).

#### 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 19 auf Seite 210](#).

## Prozedur

- [„Erforderliche Funktionen für die Einrichtung und Steuerung von Kanälen“ auf Seite 209](#)
- [„Erste Schritte mit Objekten“ auf Seite 211](#)
- [„Kommunikation unter Windows einrichten“ auf Seite 218](#)
- [„Kommunikation unter UNIX and Linux einrichten“ auf Seite 226](#)

## Zugehörige Tasks

[„Kanäle in IBM i überwachen und steuern“ auf Seite 233](#)

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 Informationen

 [Kanalprogramme unter UNIX, Linux, and Windows](#)

 [Beispiel für Nachrichtenkanalplanung für UNIX, Linux, and Windows](#)

[Beispielkonfigurationsdaten](#)

[Kanalattribute](#)

## **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-Objekt-namen gültig sind.

Informationen zu anderen kanalbezogenen Funktionen finden Sie in den folgenden Abschnitten:

- [„Erste Schritte mit Objekten“ auf Seite 211](#)
- [„Erstellen von zugeordneten Objekten“ auf Seite 212](#)
- [„Standardobjekte erstellen“ auf Seite 212](#)
- [„Kanal erstellen“ auf Seite 212](#)
- [„Anzeigen eines Kanals“ auf Seite 213](#)
- [„Kanalstatus anzeigen“ auf Seite 213](#)
- [„Links mit Ping überprüfen“ auf Seite 214](#)
- [„Kanal starten“ auf Seite 214](#)
- [„-Kanal stoppen“ auf Seite 216](#)
- [„Kanal umbenennen“ auf Seite 217](#)
- [„Kanal zurücksetzen“ auf Seite 217](#)
- [„Unbestätigte Nachrichten in einem Kanal auflösen“ auf Seite 218](#)

In [Tabelle 19 auf Seite 210](#) ist die komplette Liste der IBM MQ-Funktionen angezeigt, die Sie eventuell benötigen.

<i>Tabelle 19. In UNIX, Linux, and Windows-Systemen erforderliche Funktionen</i>			
<b>Funktion</b>	<b>Steuerbefehle</b>	<b>MQSC</b>	<b>IBM MQ Explorer-Entsprechung?</b>
WS-Managerfunktionen			
Warteschlangenmanager ändern		<a href="#">ALTER QMGR</a>	Ja
Warteschlangenmanager erstellen	<a href="#">crtmqm</a>		Ja
WS-Manager löschen	<a href="#">dlmqm</a>		Ja
WS-Manager anzeigen		<a href="#">ANZEIGEN QMGR</a>	Ja
WS-Manager beenden	<a href="#">endmqm</a>		Ja
Ping-WS-Manager		<a href="#">PING QMGR</a>	Nein
WS-Manager starten	<a href="#">strmqm</a>		Ja
Befehlsserverfunktionen			
Befehlsserver anzeigen	<a href="#">dspmqcsv</a>		Nein
Befehlsserver beenden	<a href="#">endmqcsv</a>		Nein
Befehlsserver starten	<a href="#">strmqcsv</a>		Nein
Warteschlangenfunktionen			
Warteschlange ändern		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE Siehe, <a href="#">ALTER-Warteschlangen</a> .	Ja
Warteschlange löschen		<a href="#">CLEAR QLOCAL</a>	Ja
Warteschlange erstellen		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE Siehe, <a href="#">DEFINE Queues</a> .	Ja
Warteschlange löschen		DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE Siehe <a href="#">DELETE-Warteschlangen</a> .	Ja
Warteschlange anzeigen		<a href="#">ANZEIGEN QUEUE</a>	Ja
Prozessfunktionen			
Änderungsprozess		<a href="#">ALTER PROCESS</a>	Ja
Prozess erstellen		<a href="#">DEFINE PROCESS</a>	Ja

Tabelle 19. In UNIX, Linux, and Windows-Systemen erforderliche Funktionen (Forts.)			
Funktion	Steuerbefehle	MQSC	IBM MQ Explorer-Entsprechung?
Prozess löschen		<u>DELETE PROCESS</u>	Ja
Anzeigeprozess		<u>ANZEIGEN PROZESS</u>	Ja
Kanalfunktionen			
Kanal ändern		<u>ALTER CHANNEL</u>	Ja
Kanal erstellen		<u>DEFINE CHANNEL</u>	Ja
Kanal löschen		<u>DELETE CHANNEL</u>	Ja
Kanal anzeigen		<u>ANZEIGEN CHANNEL</u>	Ja
Kanalstatus anzeigen		<u>ANZEIGEN CHSTATUS</u>	Ja
Kanal beenden		<u>STOP CHANNEL</u>	Ja
Pingkanal		<u>Pingkanal</u>	Ja
Kanal zurücksetzen		<u>Kanal zurücksetzen</u>	Ja
Kanal auflösen		<u>Auflösungskanal</u>	Ja
Kanal ausführen	<u>runmqchl</u>	<u>START CHANNEL</u>	Ja
Kanalinitiator ausführen	<u>runmqchi</u>	<u>START CHINIT</u>	Nein
Listener ausführen <sup>1</sup>	<u>runmqlsr</u>	<u>START LISTENER</u>	Nein
Listener beenden	<u>endmqlsr</u> (nur Windows-Systeme, AIX, HP-UX und Solaris)		Nein

#### Anmerkung:

1. Ein Listener kann beim Start des Warteschlangenmanagers automatisch gestartet werden.

### 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:

### **ULW** 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 UNIX, Linux, and Windows](#).

### **ULW** 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 im Abschnitt [IBM MQ-Steuerbefehle-Referenz](#).

## 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.

### **ULW** 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') +
CONNNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

In allen MQSC-Beispielen wird der Befehl in einer Datei mit Befehlen angezeigt und wie in UNIX, 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:

-  Verwenden Sie unter Windows die Tastenkombination Strg + Z, um den Eintrag in der Befehlszeile zu beenden.
-   Verwenden Sie unter UNIX and Linux die Tastenkombination Strg + d.
- Alternativ dazu können Sie unter UNIX, Linux, and Windows den Befehl **end** verwenden.

### **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
```

### **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
-  Thread-ID (nur Windows)

## Kanalstatusbeispiele anzeigen

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

Der gespeicherte Status gilt erst dann, wenn mindestens ein Nachrichtensatz 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.

### **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.

### **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** **UNIX** Für TCP unter UNIX and Linux kann inetd so konfiguriert werden, dass ein Kanal gestartet wird. inetd wird als separater Prozess gestartet.
- **Linux** **UNIX** Für LU 6.2 in UNIX 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** **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.

#### **Linux** **UNIX** **UNIX 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 UNIX, Linux, and Windows den Befehl **runmqclsr**, 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.

```
runmqclsr -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 195.

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 Nachrichtenstapel 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.

## Stoppspiel 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.

### **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.

### **Kanal zurücksetzen**

Verwenden Sie den MQSC-Befehl RESET CHANNEL, um die Nachrichtenfolgennummer 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 211

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 183

Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

### **Zugehörige Tasks**

„Verteilte Warteschlangensteuerung konfigurieren“ auf Seite 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### **Zugehörige Informationen**

RESET CHANNEL

## 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 unbestätigte\_Nachrichten darstellt. Anschließend gibt es, wie angefordert, folgende Probleme aus:

- BACKOUT, um die Nachrichten in die Übertragungswarteschlange zurückzuspeichern; 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 211

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 183

Die Kanalsteuerfunktion stellt Funktionen zur Verfügung, mit der Sie Kanäle definieren, überwachen und steuern können.

### Zugehörige Tasks

„Verteilte Warteschlangensteuerung konfigurieren“ auf Seite 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### Zugehörige Informationen

[GELÖST-CHANNEL](#)

## 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, ist es erforderlich, dass die Verbindung definiert und verfügbar ist. 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.

### 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 219
  - [„LU 6.2-Verbindung unter Windows definieren“](#) auf Seite 221
  - [„NetBIOS-Verbindung in Windows definieren“](#) auf Seite 223

## Zugehörige Tasks

[„Kanäle unter UNIX, Linux, and Windows überwachen und steuern“](#) auf Seite 208

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 dem Server und dem Client konfigurieren“](#) auf Seite 17

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 UNIX and Linux einrichten“](#) auf Seite 226

DQM ist eine ferne Warteschlangenfunktion für IBM MQ. Es stellt Kanalsteuerprogramme für den Warteschlangenmanager zur Verfügung, die die Schnittstelle zu Kommunikationsverbindungen bilden, die vom Systembediener gesteuert werden können. Die Kanaldefinitionen, die von der Verwaltung der verteilten Steuerung von Warteschlangen gehalten werden, verwenden diese Verbindungen.

## Zugehörige Verweise

[„Zu verwendende Übertragungsart“](#) auf Seite 18

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.

## 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('OS2R0G3(1822)') +
  XMITQ('XMITQ name') +
  REPLACE
```

Dabei steht OS2R0G3 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 Attribut "port", das in der TCP-Zeilengruppe der Datei `.ini` 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 `runmq1sr`.

Ein grundlegendes Beispiel für die Verwendung des Befehls `runmq1sr`:

```
runmq1sr -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 Attribut "port", das in der TCP-Zeilengruppe der Datei `.ini` 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 207 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 191.

Unter Windows steuert der Registrywert 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 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 Windows finden Sie unter „Verwenden der Backlog-Option für den TCP-Listener unter UNIX and Linux“ auf Seite 229 .

### **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.

<b>Ferne Plattform</b>	<b>TPNAME</b>	<b>TPPATH</b>
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.	-
UNIX and Linux-Systeme	Entsprechendes gilt für die entsprechenden Nebeninformationen zum fernen Warteschlangenmanager.	<i>MQ_INSTALLATION_PATH</i> /bin/amqcrs6a
Windows	Wie im Windows-Befehl 'Listener ausführen' angegeben, oder dem aufrufbaren Transaktionsprogramm, das mit TpSetup unter Windows definiert wurde.	<i>MQ_INSTALLATION_PATH</i> \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 222

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 222

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. Beispiel:

```
Partner LU Name      OS2R0G2
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. Das letzte Teil in eckigen Klammern ist optional und ist 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. Beispiel:

```
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 224](#)

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 224](#)

Die Grenzwerte des Warteschlangenmanagers für NetBIOS-Sitzungen, -Befehle und -Namen können auf zwei Arten angegeben werden.

[„LAN-Adapternummer erstellen“ auf Seite 225](#)

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 225](#)

Definieren der Schritte, die zum Einleiten einer Verbindung erforderlich sind.

„Ziellistener für die NetBIOS-Verbindung definieren“ auf Seite 225

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, z. B.:

```
RUNMQLSR -t NETBIOS -l my_station
```

2. Die Umgebungsvariable MQNAME mit einem Wert, der durch den folgenden Befehl festgelegt wird:

```
SET MQNAME= my_station
```

Sie können den MQNAME-Wert 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 Befehl SET MQNAME absetzen. Der MQNAME-Wert ist beliebig, aber er muss für jeden Prozess eindeutig sein.

3. Die NETBIOS-Zeilengruppe in der WS-Managerkonfigurationsdatei qm.ini. Beispiel:

```
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 Fehlermeldungen 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. Beispiel:

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 "runmqsr" 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) +  
CONNNAME(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. Beispiel:

```
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-Stationenname 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 207 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

UNIX

## Kommunikation unter UNIX and Linux einrichten

DQM ist eine ferne Warteschlangenfunktion für IBM MQ. Es stellt Kanalsteuerprogramme für den Warteschlangenmanager zur Verfügung, die die Schnittstelle zu Kommunikationsverbindungen bilden, die vom Systembediener gesteuert werden können. Die Kanaldefinitionen, die von der Verwaltung der verteilten Steuerung von Warteschlangen gehalten werden, verwenden diese Verbindungen.

### Vorbereitende Schritte

Es kann hilfreich sein, auf die folgenden Abschnitte zu verweisen:

- ▶ **AIX** [Beispielkonfiguration - IBM MQ for AIX](#)
- ▶ **HP-UX** [Beispielkonfiguration - IBM MQ for HP-UX](#)
- ▶ **Solaris** [Beispielkonfiguration - IBM MQ for Solaris](#)
- ▶ **Linux** [Beispielkonfiguration - IBM MQ for Linux](#)

### 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 UNIX 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. Weitere Informationen zu IBM MQ MQI clients finden Sie unter [Übersicht über IBM MQ MQI clients](#).

## Vorgehensweise

Informationen zum Konfigurieren der Kommunikation für Ihr UNIX and Linux-System finden Sie im Unterabschnitt für den ausgewählten Kommunikationstyp:

- „[TCP-Verbindung unter UNIX and Linux definieren](#)“ auf Seite 227
- „[LU 6.2-Verbindung unter UNIX and Linux definieren](#)“ auf Seite 231

## Zugehörige Tasks

„[Kanäle unter UNIX, Linux, and Windows überwachen und steuern](#)“ auf Seite 208

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 dem Server und dem Client konfigurieren](#)“ auf Seite 17

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 218

Wenn ein Verwaltungskanal für die verteilte Steuerung von Warteschlangen gestartet wird, versucht er, die in der Kanaldefinition angegebene Verbindung zu verwenden. Damit dies gelingt, ist es erforderlich, dass die Verbindung definiert und verfügbar ist. 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 18

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

## **TCP-Verbindung unter UNIX 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.

## 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). Informationen zur Verwendung des erweiterten inet-Dämons auf einem Linux-System finden Sie im Abschnitt [TCP-Verbindung auf Linux aufbauen](#).

### Zugehörige Konzepte

[„TCP/IP-Listener unter UNIX and Linux verwenden“](#) auf Seite 228

Um Kanäle in UNIX and Linux zu starten, müssen die Datei `/etc/services` und die Datei `inetd.conf` bearbeitet werden.

[„Verwenden der Backlog-Option für den TCP-Listener unter UNIX and Linux“](#) auf Seite 229

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 230

Verwenden Sie den Befehl `runmqtsr`, um den Listener auszuführen, der mit IBM MQ bereitgestellt wird und neue Kanäle als Threads startet.

[„Verwendung der Option "TCP/IP SO\\_KEEPALIVE"“](#) auf Seite 231

Auf einigen UNIX 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** → **UNIX** [TCP/IP-Listener unter UNIX and Linux verwenden](#)

Um Kanäle in UNIX and Linux zu starten, müssen die Datei `/etc/services` und die Datei `inetd.conf` bearbeitet werden.

Befolgen Sie diese Anweisungen:

1. Bearbeiten Sie die Datei `/etc/services` :

**Anmerkung:** Um die Datei `/etc/services` bearbeiten zu können, müssen Sie als Superuser oder als 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:

- Unter AIX:

```
refresh -s inetd
```

- Unter HP-UX von der mqm-Benutzer-ID:

```
inetd -c
```

- Unter Solaris 10 oder höher:

```
inetconv
```

- Auf anderen UNIX and Linux-Systemen (einschließlich Solaris 9):

```
kill -1 process_number
```

Wenn das von inetd gestartete Empfangsprogramm 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 Empfangsprogramm auf.
3. Ändern Sie die Datei `inetd.conf` so, dass das Shell-Script an Stelle des Listenerprogramms aufgerufen wird.

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. Beispiel:

```
MQSeries1 1414/tcp
MQSeries2 1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

`MQ_INSTALLATION_PATH` steht 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 UNIX and Linux](#)“ auf Seite 229.

Linux

UNIX

Verwenden der Backlog-Option für den TCP-Listener unter UNIX and Linux

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 21](#) auf Seite 229 angezeigt.

Tabelle 21. Maximale Anzahl ausstehender Verbindungsanforderungen, die an einem TCP/IP-Port in die Warteschlange	
Serverplattform	Maximale Verbindungsanforderungen
AIX	100

Tabelle 21. Maximale Anzahl ausstehender Verbindungsanforderungen, die an einem TCP/IP-Port in die Warteschlange (Forts.)

Serverplattform	Maximale Verbindungsanforderungen
▶ <b>HP-UX</b> HP-UX	20
▶ <b>Linux</b> Linux	100
▶ <b>IBM i</b> IBM i	255
▶ <b>Solaris</b> Solaris	100
▶ <b>Windows</b> Windows Server	100
▶ <b>Windows</b> Windows Workstation	100

Wenn der Rückstand die in Tabelle 21 auf Seite 229 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 21](#) auf Seite 229) 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 `runmqclsr -b` oder
- Verwenden Sie den MQSC-Befehl **DEFINE LISTENER** mit dem Attribut `BACKLOG`, das auf den erforderlichen Wert gesetzt ist.

Weitere Informationen zum Befehl `runmqclsr` finden Sie im Abschnitt [runmqcls](#). Weitere Informationen zum Befehl `DEFINE LISTENER` finden Sie in [DEFINE LISTENER](#).

▶ **Linux** ▶ **UNIX** *IBM MQ-Listener verwenden*

Verwenden Sie den Befehl `runmqclsr`, um den Listener auszuführen, der mit IBM MQ bereitgestellt wird und neue Kanäle als Threads startet.

Beispiel:

```
runmqclsr -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 207 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** → **UNIX** *Verwendung der Option "TCP/IP SO\_KEEPALIVE"*

Auf einigen UNIX 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 191), müssen Sie den folgenden Eintrag in Ihrer Warteschlangenmanagerkonfigurationsdatei (qm.ini) hinzufügen:

```
TCP:
KeepAlive=yes
```

Weitere Informationen finden Sie in der Dokumentation für Ihr UNIX and Linux-System.

**Linux** → **UNIX** **LU 6.2-Verbindung unter UNIX 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 Online-dokumentation 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.

<i>Tabelle 22. Einstellungen auf dem lokalen UNIX and Linux-System für eine ferne Warteschlangenmanagerplattform</i>		
<b>Ferne Plattform</b>	<b>TPNAME</b>	<b>TPPATH</b>
z/OS ohne CICS	Dasselbe gilt für den entsprechenden TPName-Wert in den Nebeninformatio- nen zum fernen Warteschlangenmana- ger.	-
z/OS mit CICS	CKRC (Sender) CKSV (Requester) CKRC (Server)	-
IBM i	Entsprechendes gilt für den Vergleichs- wert im Routing-Eintrag auf dem IBM i- System.	-
UNIX and Linux- Systeme	Dasselbe gilt für den entsprechenden TPName-Wert in den Nebeninformatio- nen zum fernen Warteschlangenmana- ger.	<i>MQ_INSTALLATION_PATH/bin/ amqcrs6a</i>

Tabelle 22. Einstellungen auf dem lokalen UNIX and Linux-System für eine ferne Warteschlangenmanagerplattform (Forts.)

Ferne Plattform	TPNAME	TPPATH
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 UNIX and Linux“ auf Seite 232

Erstellen Sie auf UNIX 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 UNIX and Linux“ auf Seite 232

Erstellen Sie auf UNIX and Linux-Systemen einen empfangsbereiten Anschluss auf der Empfangsseite, ein logisches LU 6.2-Verbindungsprofil und ein TPN-Profil.

#### Linux → UNIX Sendeseite für LU 6.2 unter UNIX and Linux

Erstellen Sie auf UNIX 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. Beispiel:

```
Partner LU Name          REMHOST
Remote TP Name          recv
Service Transaction Program no
Mode Name               #INTER
```

Verwenden Sie unter HP-UX die Umgebungsvariable APPCLLU, um die lokale LU zu benennen, die der Sender verwenden soll. Auf Solaris setzen Sie die Umgebungsvariable APPC\_LOCAL\_LU auf den lokalen LU-Namen.

SECURITY PROGRAM wird verwendet, sofern Unterstützung durch CPI-C besteht, wenn IBM MQ versucht, eine SNA-Sitzung aufzubauen.

#### Linux → UNIX Empfangen auf LU 6.2 unter UNIX and Linux

Erstellen Sie auf UNIX 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. Geben Sie unter AIX, Solaris und HP-UX 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:

```
amqcis6a -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, Kanalfolgennummern 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 Folgennummern 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 (Bedienerrführung) verwenden, können Sie den entsprechenden Warteschlangenmanager angeben. Wenn Sie die Eingabeaufforderung nicht verwenden, wird der Standardwarteschlangenmanager angenommen. Mit F4 = Bedienerrführung 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 151.

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 234
  - [„Kanal unter IBM i erstellen“](#) auf Seite 234
  - [„Kanal unter IBM i starten“](#) auf Seite 236
  - [„Kanal unter IBM i auswählen“](#) auf Seite 237
  - [„Durchsuchen eines Kanals unter IBM i“](#) auf Seite 237
  - [„Kanal unter IBM i umbenennen“](#) auf Seite 239
  - [„Mit Kanalstatus unter IBM i arbeiten“](#) auf Seite 239
  - [„Auswahlmöglichkeiten für Arbeit mit Kanal unter IBM i“](#) auf Seite 240

### Zugehörige Konzepte

[„Kommunikation für IBM i konfigurieren“](#) auf Seite 246

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 dem Server und dem Client konfigurieren“](#) auf Seite 17

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 Informationen

[#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 [IBM MQ for IBM i-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.

```
MQM-Kanal erstellen (CRTMQMCHL)

Auswahl eingeben und Eingabetaste drücken.

Kanalname..... > CHANNAME_____
Kanaltyp..... > *SDR__ *RCVR, *SDR, *SVR, *RQSTR ...
Name des Nachrichtenwarteschlangenmanagers *DFT_____

Ersätzen..... *NO_ *NO, *YES
Transportart..... *TCP____ *LU62, *TCP, *SYSDFTCHL
Text 'Beschreibung'..... > 'Beispielkanaldefinition' _____

Verbindungsname..... *SYSDFTCHL_____
_____
_____
_____
_____
_____
Mehr ...
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

Abbildung 26. Kanal erstellen (1)

```
MQM-Kanal erstellen (CRTMQMCHL)

Auswahl eingeben und Eingabetaste drücken.

Übertragungswarteschlange.....TRANSMISSION_QUEUE_NAME ' _____

Nachrichtenkanalagent..... *NONE_____ Name, *SYSDFTCHL, *NONE
Bibliothek..... _____ Name
Benutzer-ID des Nachrichtenkanalagenten. *SYSDFTCHL__ Zeichenwert ...
ID des codierten Zeichensatzsets *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Stapelgröße..... 50 _____ 1-9999, *SYSDFTCHL
Intervall für Verbindungstrennung..... 6000 _____ 1-999999, *SYSDFTCHL
Kurzes Wiederholungsintervall..... 60 _____ 0-99999999, *SYSDFTCHL
Kurzer Wiederholungszähler..... 10 _____ 0-99999999, *SYSDFTCHL
Langes Wiederholungsintervall..... 1200 _____ 0-99999999, *SYSDFTCHL
Zähler für lange Wiederholungsversuche..... 99999999 _____ 0-99999999, *SYSDFTCHL
Sicherheitsexit..... *NONE_____ Name, *SYSDFTCHL, *NONE
Bibliothek..... _____ Name
Benutzerdaten für Sicherheitsexit.... *SYSDFTCHL_____

Mehr ...
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

Abbildung 27. Kanal erstellen (2)

MQM-Kanal erstellen (CRTMQMCHL)

Auswahl eingeben und Eingabetaste drücken.

```
Sendeexit..... *NONE_____ Name, *SYSDFTCHL, *NONE
Bibliothek..... _____ Name
+ für weitere Werte _____
Exitbenutzerdaten senden..... _____
+ für weitere Werte _____
Empfangsexit..... *NONE_____ Name, *SYSDFTCHL, *NONE
Bibliothek..... _____ Name
+ für weitere Werte _____

-----
Benutzerdaten des Empfangsexits.... _____
+ für weitere Werte _____
Nachrichtenexit..... *NONE_____ Name, *SYSDFTCHL, *NONE
Bibliothek..... _____ Name
+ für weitere Werte _____

-----
Mehr ...
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

Abbildung 28. Kanal erstellen (3)

MQM-Kanal erstellen (CRTMQMCHL)

Auswahl eingeben und Eingabetaste drücken.

```
Nachrichtenexitbenutzerdaten..... _____
+ für weitere Werte _____
Nachricht konvertieren..... *SYSDFTCHL_ *YES, *NO, *SYSDFTCHL
Folgenummernwicklung..... 999999999__ 100-999999999, *SYSDFTCHL
Maximale Nachrichtenlänge..... 4194304__ __ 0-4194304, *SYSDFTCHL
Überwachungssignalintervall..... 300__ __ 0-999999999, *SYSDFTCHL
Nicht persistente Nachrichtengeschwindigkeit.. *FAST_____ *FAST, *NORMAL, *SYSDFTCHL
Kennwort..... *SYSDFTCHL_ Zeichenwert, *BLANK ...
Taskbenutzerprofil..... *SYSDFTCHL_ Zeichenwert, *BLANK ...
Transaktionsprogrammname.... *SYSDFTCHL

Unten
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

Abbildung 29. Kanal erstellen (4)

## 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.

Mit MQM-Kanälen arbeiten

Name des Warteschlangenmanagers.: CNX

Geben Sie Optionen ein, und drücken Sie die Eingabetaste

2 = Ändern 3 = Kopieren 4 = Löschen 5 = Anzeigen 8 = Arbeiten mit Status 13 = Ping  
14 = Start 15 = Ende 16 = Zurücksetzen 17 = Auflösen

Opt Name Typ Transportstatus

CHLNIC \*RCVR \*TCP INAKTIV

CORSAIR.TO.MUSTANG \*SDR \*LU62 INAKTIV

FV.CHANNEL.MC.DJE1 \*RCVR \*TCP INAKTIV

FV.CHANNEL.MC.DJE2 \*SDR \*TCP INAKTIV

FV.CHANNEL.MC.DJE3 \*RQSTR \*TCP INAKTIV

FV.CHANNEL.MC.DJE4 \*SVR \*TCP INAKTIV

FV.CHANNEL.PETER \*RCVR \*TCP INAKTIV

FV.CHANNEL.PETER.LU \*RCVR \*LU62 INAKTIV

FV.CHANNEL.PETER.LU1 \*RCVR \*LU62 INAKTIV

Mehr ...

Parameter oder Befehl

==>

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F6 = Erstellen F9 = Auffinden F12 = Abbrechen

F21=Print

Abbildung 30. 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.

```
MQM-Kanal anzeigen

Kanalname..... : ST.JST.2T01
Name des Warteschlangenmanagers..... : QMREL
Kanaltyp..... : *SDR
Transportart..... : *TCP
Text 'description'..... : Johns Absender an WINSDOA1

Verbindungsname..... : MUSTANG

Übertragungswarteschlange..... : WINSDOA1

Nachrichtenkanalagent....:
Bibliothek.....:
Benutzer-ID des Nachrichtenkanalagenten: *NONE
Stapelintervall..... : 0
Stapelgröße..... : 50
Intervall für Verbindungstrennung.... : 6000

F3 = Verlassen F12 = Abbrechen F21 = Drucken
```

Abbildung 31. TCP/IP-Kanal anzeigen (1)

```
MQM-Kanal anzeigen

Kurzer Wiederholungsintervall..... : 60
Kurzer Wiederholungszähler..... : 10
Langes Wiederholungsintervall..... : 6000
Zähler für lange Wiederholungsversuche..... : 10
Sicherheitsexit.....:
Bibliothek.....:
Benutzerdaten für Sicherheitsexit...:
Sendeexit.....:
Bibliothek.....:
Benutzerdaten des Sendeexits....:
Empfangsexit.....:
Bibliothek.....:
Benutzerdaten des Empfangsexits...:
Nachrichtenexit.....:
Bibliothek.....:
Benutzerdaten des Nachrichtenexits...:
Mehr ...

F3 = Verlassen F12 = Abbrechen F21 = Drucken
```

Abbildung 32. TCP/IP-Kanal anzeigen (2)

MQM-Kanal anzeigen

```
Folgenummernwicklung..... : 999999999
Maximale Nachrichtenlänge... : 10000
Nachricht konvertieren..... : *NO
Überwachungssignalintervall..... 300
Nicht persistente Nachrichtengeschwindigkeit.. *FAST
```

Unten

F3 = Verlassen F12 = Abbrechen F21 = Drucken

Abbildung 33. TCP/IP-Kanal anzeigen (3)

### 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 Kanälen gleichzeitig umbenannt werden.

### 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-Arbeiten mit Kanalstatus

Geben Sie Optionen ein, und drücken Sie die Eingabetaste

5 = Anzeigen 13 = Ping 14 = Start 15 = Ende 16 = Zurücksetzen 17 = Auflösen

```
Opt Name Verbindung Unbestätigt letzte 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.2T01.CHL 9.20.2.213 NO 216
```

Unten

Parameter oder Befehl

==>

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F6 = Erstellen F9 = Auffinden F11 = Ansicht ändern

F12=Cancel F21=Print

Abbildung 34. 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.



## 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
<u>„2=Change“ auf Seite 241</u>	Ändert die Attribute eines Kanals.
<u>„3=Copy“ auf Seite 241</u>	Kopiert die Attribute eines Kanals in einen neuen Kanal.
<u>„4=Delete“ auf Seite 241</u>	Löscht einen Kanal.
<u>„5=Display“ auf Seite 241</u>	Zeigt die aktuellen Einstellungen für den Kanal an.
<u>„6=Create“ auf Seite 242</u>	Zeigt die Anzeige 'Kanal erstellen' an
<u>„8 = Mit Status arbeiten“ auf Seite 242</u>	Zeigt die Kanalstatusanzeigen an.

<b>Menüoption</b>	<b>Beschreibung</b>
<u>„13=Ping“ auf Seite 243</u>	Führt die Ping-Funktion aus, um die Verbindung zum benachbarten System zu testen, indem eine feste Datennachricht mit dem fernen Ende ausgetauscht wird.
<u>„14=Start“ auf Seite 243</u>	Startet den ausgewählten Kanal oder setzt einen inaktivierten Empfängerkanal zurück.
<u>„15=End“ auf Seite 244</u>	Fordert den Kanal zum Schließen an.
<u>„16=Reset“ auf Seite 245</u>	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.
<u>„17=Resolve“ auf Seite 245</u>	Fordert den Kanal an, um unbestätigte Nachrichten aufzulösen, ohne dass eine Verbindung zum anderen Ende hergestellt wird.
<u>„18 = Berechtigung anzeigen“ auf Seite 246</u>	Zeigt die IBM MQ-Objektberechtigung an
<u>„19 = Berechtigung erteilen“ auf Seite 246</u>	Erteilt die IBM MQ-Objektberechtigung
<u>„20 = Berechtigung entziehen“ auf Seite 246</u>	Widerruft die IBM MQ-Objektberechtigung
<u>„21 = Objekt wiederherstellen“ auf Seite 246</u>	Stellt das IBM MQ-Objekt wieder her
<u>„22 = Satzabbild“ auf Seite 246</u>	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 26 auf Seite 235](#) 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 i bereitgestellt 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 Informationen

[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. Siehe [„Mit Kanalstatus unter IBM i arbeiten“ auf Seite 239](#).

- Kanalname
- Kanaltyp
- Kanalstatus
- Kanalinstanz
- Ferner Warteschlangenmanager
- Name der Übertragungswarteschlange
- Kommunikationsverbindungsname
- Status des Kanals im Status 'Unbestätigt'
- Letzte Folgenummer
- Anzahl der unbestätigten Nachrichten
- Folgenummer in unbestätigter 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 Nachrichtenfolgennummer geändert. Verwenden Sie diese Option mit Vorsicht, und erst nachdem Sie die Option Resolve verwendet haben, um alle unbestätigten 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 unbestätigten 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 unbestätigte\_Nachrichten darstellt. Anschließend gibt es, wie angefordert, folgende Probleme aus:

- BACKOUT, um die Nachrichten in die Übertragungswarteschlange zurückzuspeichern; 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 GRMQMAUT 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 Widerrufsberechtigung, 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 247

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 248

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 i finden Sie im Handbuch *IBM i Communication Configuration Reference*.

Darüber hinaus muss bei Bedarf die Auslöseranordnung mit der Definition der erforderlichen Prozesse und Warteschlangen vorbereitet werden.

### **Zugehörige Tasks**

[„Kanäle in IBM i überwachen und steuern“](#) auf Seite 233

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 Informationen**

[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.

## Verwendung der Option TCP-Listener-Backlog

In TCP werden die Verbindungen nur unvollständig behandelt, wenn zwischen dem Server und dem Client ein Dreizeige-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 UNIX and Linux“](#) auf Seite 229 .

### Zugehörige Konzepte

[„Empfang auf TCP“](#) auf Seite 248

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.

#### 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. Beispiel:

```
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 191), 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.

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 UNIX and Linux	Das aufrufbare Transaktionsprogramm, das in der fernen LU 6.2-Konfiguration definiert ist.

Tabelle 23. Einstellungen auf dem lokalen IBM i-System für eine ferne Warteschlangenmanagerplattform (Forts.)

Ferne Plattform	TPNAME
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 250

Verwenden Sie den Befehl CRTMQMCHL, um einen Kanal des Transporttyps \*LU62 zu definieren.

„Initiiertes Ende (Empfänger)“ auf Seite 253

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 in IBM MQ for IBM i 5.3 oder höher optional.

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 35. 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.

### **Mode**

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 Nebeninformati-  
onen 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 36. 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 Leitweeintrag 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 Initialisierungs-ende (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           'QNPSEVR'     37
70   *RTGDTA           'QNMAPPINGD'  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 37. 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 Leitweeintrag 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 Leitweeintrag 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 Leitweeintrag 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 [Konstruktive Cluster](#).

### Zugehörige Tasks

„[Clusterthemendefinition in einen anderen WS-Manager verschieben](#)“ auf Seite 390

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 Informationen

[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 Tasks

„[Neuen Cluster einrichten](#)“ auf Seite 268

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 280

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.

### Zugehörige Informationen

[Komponenten eines Clusters](#)

[Clusterkanäle](#)

[Clusterthemen definieren](#)

## 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.

 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 Warteschlangenmanager-Attribut `DEFCLXQ` auf `CHANNEL` setzen, wird für jeden Clustersenderkanal auf Basis von `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` automatisch eine eigene Clusterübertragungswarteschlange 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 7.5 durchgeführt 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.

Standardmäßig wird (außer unter z/OS) in jedem Warteschlangenmanager eine Definition für die einzelne Warteschlange SYSTEM . CLUSTER . TRANSMIT . QUEUE erstellt.  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. Beispiel:

```
DISPLAY QUEUE(*) CLUSINFO
```

### Zugehörige Informationen

[Clusterwarteschlangen](#)

[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 Kanäle auf unterschiedlichen Plattformen liegen.

Das Format der Exitnamen ist auf verschiedenen Plattformen unterschiedlich. Beispiel:

- **z/OS** Auf der z/OS-Plattform ist das Format des Parameters SCYEXIT (*security exit name*) `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.

Beispiel:

- `/var/mqm/exits/myExit.so(MsgExit)` wird zu `MSGEXIT`
- `/var/mqm/exits/myExit` wird zu `MYEXIT`

- /var/mqm/exits/myExit.so(ExitLongName) wird zu EXITLONG

- Für Warteschlangenmanager vor IBM WebSphere MQ 7 setzen Sie das Attribut **PROPCTL** auf den Wert NONE.

Jeder automatisch definierte Clustersenderkanal basiert auf dem entsprechenden Clusterempfängerkanal. Vor IBM MQ Version 7 verfügt der Clusterempfängerkanal nicht über ein Attribut **PROPCTL**, daher wird dieses Attribut im automatisch definierten Clustersenderkanal auf COMPAT gesetzt.

Wenn der Cluster **PROPCTL** verwenden muss, um Anwendungsheader wie RFH2 aus Nachrichten zu entfernen, die von einem Warteschlangenmanager in IBM WebSphere MQ 7 oder höher zu einem Warteschlangenmanager auf einer früheren Version von IBM MQ gehen, müssen Sie einen Exit für die 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 Informationen

[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 außer für SYSTEM.CLUSTER.HISTORY.QUEUE keine Standardwarteschlangendefinitionen.

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 besitzt 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 Protokoll Daten 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` , außer unter z/OS ändern. Die Warteschlange `SYSTEM.CLUSTER.TRANSMIT.QUEUE` kann nicht gelöscht werden. Es wird auch verwendet, um Berechtigungsprüfungen zu definieren, ob die verwendete Standardübertragungswarteschlange `SYSTEM.CLUSTER.TRANSMIT.QUEUE` oder `SYSTEM.CLUSTER.TRANSMIT.ChannelName` ist.

### **Zugehörige Informationen**

[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)  
CONNNAME (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 262 wird der Prozess der Auswahl einer Standardübertragungswarteschlange beschrieben; [„Auswahl von manuell definierten Übertragungswarteschlangen nach Clustersenderkanälen“](#) auf Seite 263 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 264 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. Bis IBM WebSphere MQ 7.5 haben Clusterverbindungen die Übertragungswarteschlange für einzelne Cluster (SYSTEM.CLUSTER.TRANSMIT.QUEUE) gemeinsam genutzt. Ab IBM WebSphere MQ 7.5 haben Sie die Option "Clustersenderkanäle", die keine Übertragungswarteschlangen 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. Weitere Informationen finden Sie im Abschnitt [Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen](#) und „[Cluster und Cluster-Übertragungswarteschlange hinzufügen, um den Datenverkehr der Clusternachrichten zu isolieren, die von einem Gateway-Warteschlangenmanager gesendet werden](#)“ auf Seite 319.

Um einen Clustersenderkanal für die Verwendung einer anderen Übertragungswarteschlange als SYSTEM.CLUSTER.TRANSMIT.QUEUE unter z/OS zu konfigurieren, müssen Sie die neue Funktion der Version 8 mit dem Systemparameter "mode of operation (OPMODE)" im Makro CSQ6SYSP aktivieren.

## 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.QUEUE erstellt. 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 38 auf Seite 263](#) 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.

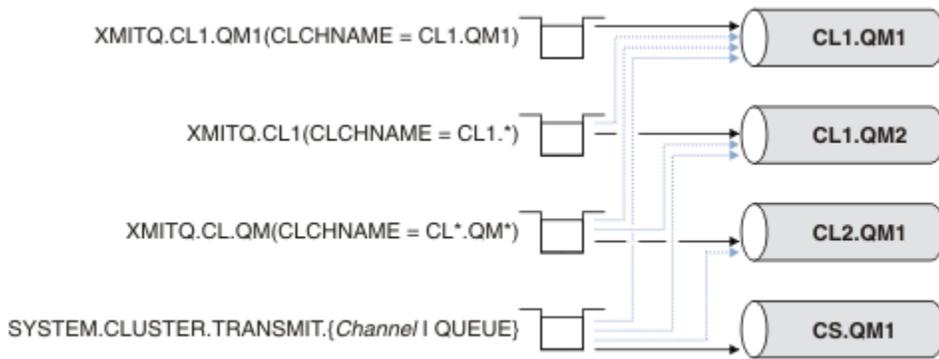


Abbildung 38. Übertragungswarteschlange/Clustersenderkanalvorrangstellung

## 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 38](#) auf Seite 263 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 38](#) auf Seite 263 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 spezifischeren 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 *Warteschlangenname* ."

## **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 39 auf Seite 267](#) 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 39 auf Seite 267](#) werden zwei Clustersenderkanäle, CS.QM1 und CS.QM2, der Standard-systemübertragungswarteschlange 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.

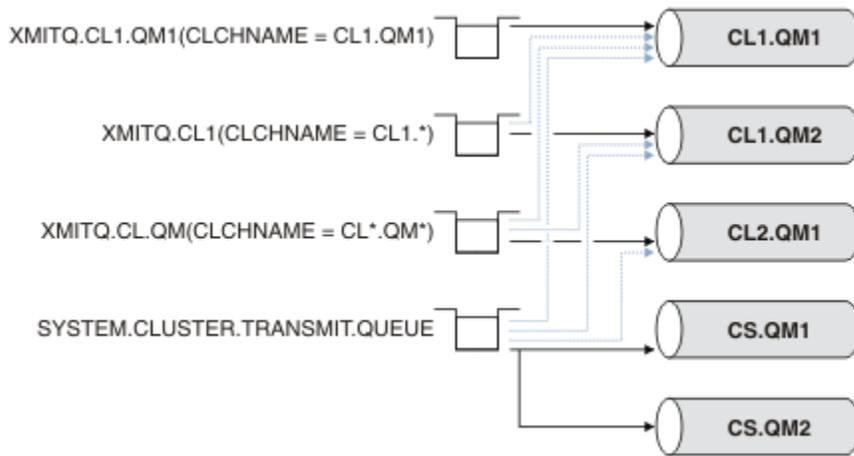


Abbildung 39. Mehrere Clustersenderkanäle

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 264).

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

„Berechnen der Größe des Protokolls“ auf Seite 568

Schätzen der Größe des Protokollwarteschlangenmanagers.

### Zugehörige Tasks

„Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager“ auf Seite 307  
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 282  
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 316  
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 319  
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.

### Zugehörige Informationen

[Clusterkanäle](#)

[Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen](#)

[Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen](#)

## 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

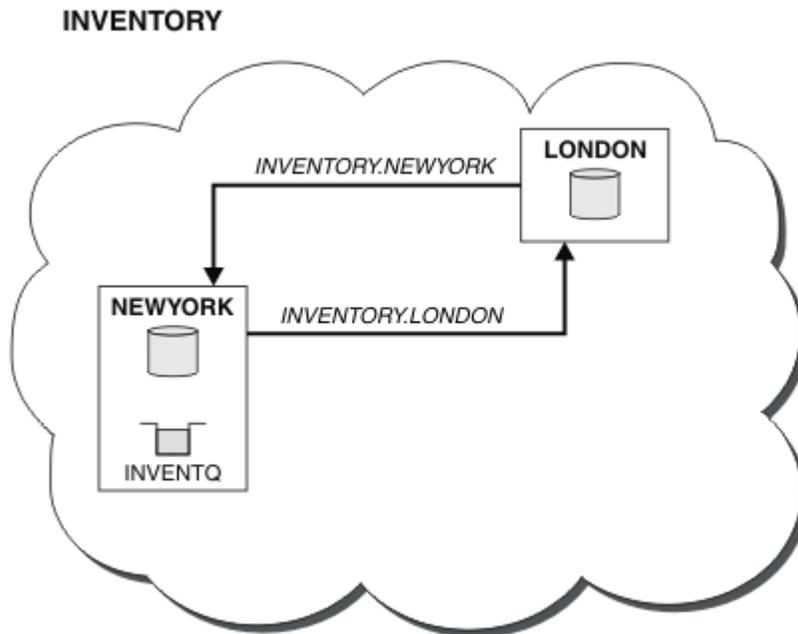
- Anstatt diese Anweisungen zu befolgen, können Sie einen der Assistenten verwenden, die mit IBM MQ Explorer bereitgestellt werden, um einen Cluster wie den von dieser Task erstellten Cluster zu erstellen. Klicken Sie auf den Ordner WS-Manager-Cluster, klicken Sie dann auf **New > Queue Manager Cluster** 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 256, [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 270](#)
- [„Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren“ auf Seite 273](#)
- [„Cluster mit Logical Unit 6.2 unter z/OS einrichten“ auf Seite 276](#)
- [„Cluster verifizieren“ auf Seite 278](#)

## Zugehörige Tasks

[„WS-Manager-Cluster konfigurieren“ auf Seite 255](#)

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.

## Zugehörige Informationen

[Cluster](#)

[Vergleich von Clustering und verteilter Steuerung von Warteschlangen  
Komponenten eines Clusters](#)

## 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 268.

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.  Unter z/OS müssen Sie die Anweisungen in „[TCP-Verbindung unter z/OS definieren](#)“ auf Seite 795 befolgen, um die TCP/IP-Verbindung einzurichten, anstatt die Listener in Schritt „4“ auf Seite 271 zu 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 zerrüttet; siehe „[WS-Manager zu einem Cluster hinzufügen](#)“ auf Seite 280.

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 Repositories 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 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 unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren](#)“ auf Seite 273

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 276

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 268.

### **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 282 aus.

### **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 zerrüttet; siehe [„WS-Manager zu einem Cluster hinzufügen“](#) auf Seite 280.

Die einzige Anwendung, die Sie gerade ausführen, ist die Bestandsanwendung. Der Clustername lautet INVENTORY.

2. Entscheiden Sie, welche WS-Manager vollständige Repositories enthalten sollen.

In jedem Cluster müssen Sie mindestens einen Warteschlangenmanager (oder vorzugsweise zwei) für die Aufnahme von vollständigen Repositories benennen. In diesem Beispiel gibt es nur zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositories 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 280 und „WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen“ auf Seite 282.

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 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 (CON-NAME) 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 Repositories 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 270

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 276

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 268.

## 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 zerrüttet; siehe [„WS-Manager zu einem Cluster hinzufügen“](#) auf Seite 280.

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.

 Siehe [Kanalinitiator](#) unter [z/OS](#) und „Empfangen auf LU 6.2“ auf Seite 799.

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)
CONNNAME(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)
CONNNAME(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 270

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 273

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 270.
- [„Cluster unter TCP/IP mit mehreren Übertragungswarteschlangen pro WS-Manager konfigurieren“](#) auf Seite 273.
- [„Cluster mit Logical Unit 6.2 unter z/OS einrichten“](#) auf Seite 276.

Eine Übersicht über den Cluster, der erstellt wurde, finden Sie unter „[Neuen Cluster einrichten](#)“ auf Seite 268.

## 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( )
CONNNAME(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)
CONNNAME(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 INVENTQ zu 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 INVENTQ zu öffnen.
  - c) Geben Sie den MQGET-Aufruf aus, um die Nachricht von INVENTQ abzurufen.
  - 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 LONDON zu öffnen.
  - h) Geben Sie den Aufruf MQPUT aus, um die Nachricht in LONDON\_reply einzureihen.
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\_reply zu öffnen.
  - c) Geben Sie den Aufruf MQGET aus, um die Nachricht von LONDON\_reply abzurufen.

## 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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wird wie in „[Neuen Cluster einrichten](#)“ auf Seite 268 beschrieben konfiguriert. Es enthält zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositories 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 zu konfigurieren, 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 Filialgeschäftlers 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 Repositories beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Wählen Sie eines der Repositories 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 Repositories 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. 

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 WS-Manager 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.

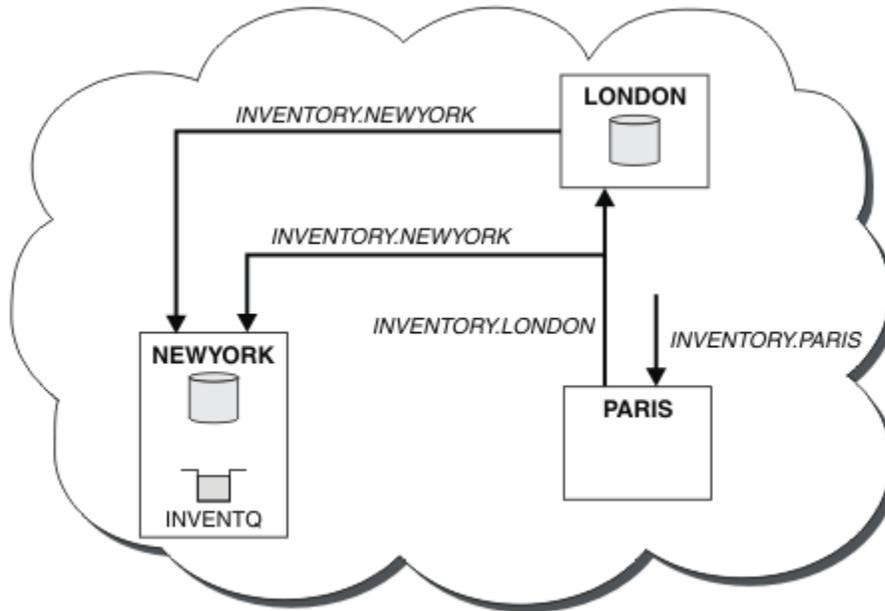


Abbildung 40. Der INVENTORY -Cluster mit drei Warteschlangenmanagern

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.

### **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 268.

### **Informationen zu diesem Vorgang**

Diese Task ist eine Alternative zu [„WS-Manager zu einem Cluster hinzufügen“](#) auf Seite 280, 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.TRANS-MIT.MODEL.QUEUE` mit 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 7.5 durchgeführt 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 Repositories beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Wählen Sie eines der Repositories 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 Repositories 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)
CONNNAME(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.

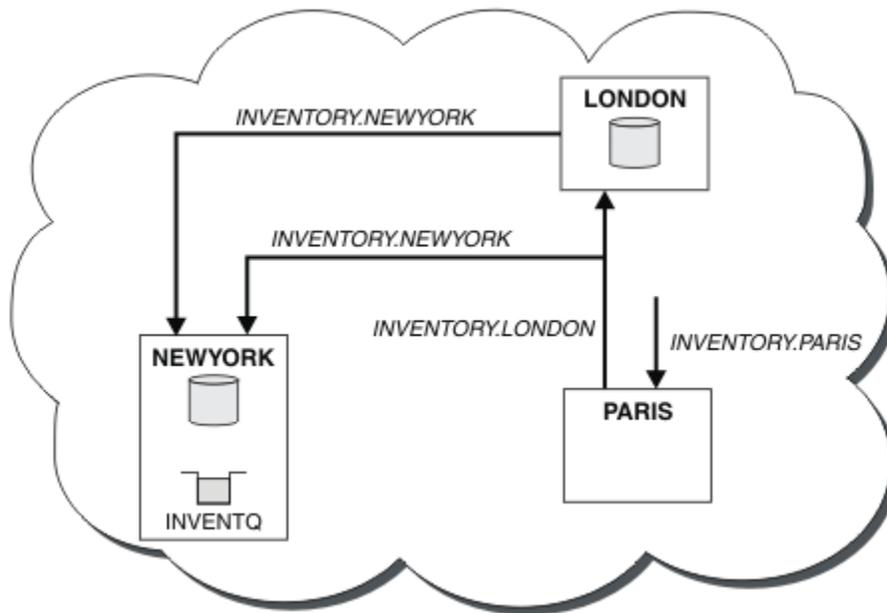


Abbildung 41. Der INVENTORY -Cluster mit drei Warteschlangenmanagern

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 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 Repositories 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 268 beschrieben konfiguriert. Es enthält zwei WS-Manager, LONDON und NEWYORK, die beide vollständige Repositories enthalten.
- Eine neue Filiale des Filialgeschäftlers 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 Netzwerkprotokoll 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 Repositories beziehen. Es sammelt Informationen über den Cluster aus einem vollständigen Repository und erstellt so ein eigenes Teilrepository. Wählen Sie eines der Repositories 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 Repositories 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 indem Sie CONNAME ( ' ' ) angeben. IBM MQ generiert den CONNAME-Wert unter Verwendung der aktuellen IP-Adresse des Systems. Weitere Informationen finden Sie unter 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 Definition 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 WS-Manager 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 280:

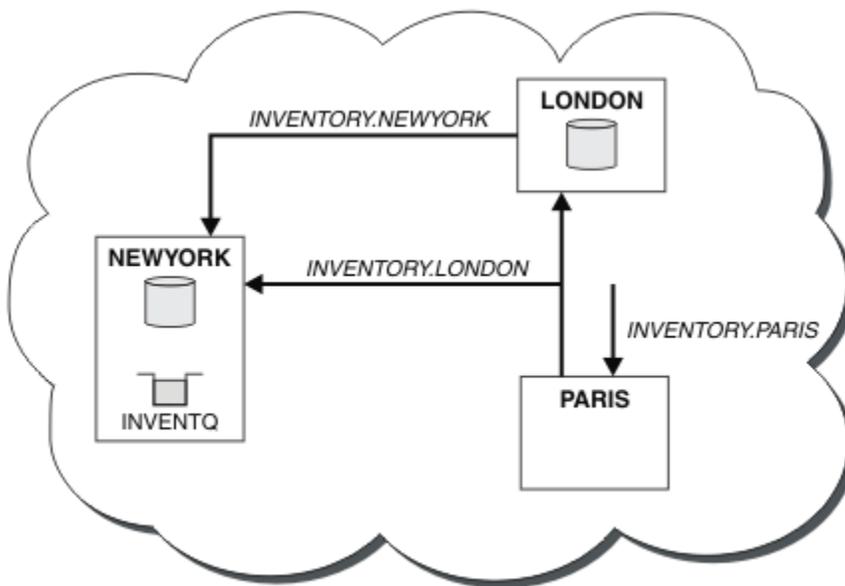


Abbildung 42. Cluster BESTANDSFÜHRUNG mit drei Warteschlangenmanagern

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 Empfangwarteschlange 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 Informationen

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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen“ auf Seite 280 beschrieben konfiguriert. Es enthält drei Warteschlangenmanager; LONDON und NEWYORK enthalten beide vollständige Repositories, 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 Inventuranwendung auf dem System in Toronto sowie in New York ausführen.
- Die Netzkonnektivität besteht zwischen allen vier Systemen.
- Das Netzwerkprotokoll 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 „Vollrepositories in einen anderen WS-Manager verschieben“ auf Seite 293.

### 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 Repositories 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 Repositories.

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)
CONNNAME(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 WS-Manager 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 43 auf Seite 290](#) wird der INVENTORY -Cluster angezeigt, der von dieser Task eingerichtet wurde.

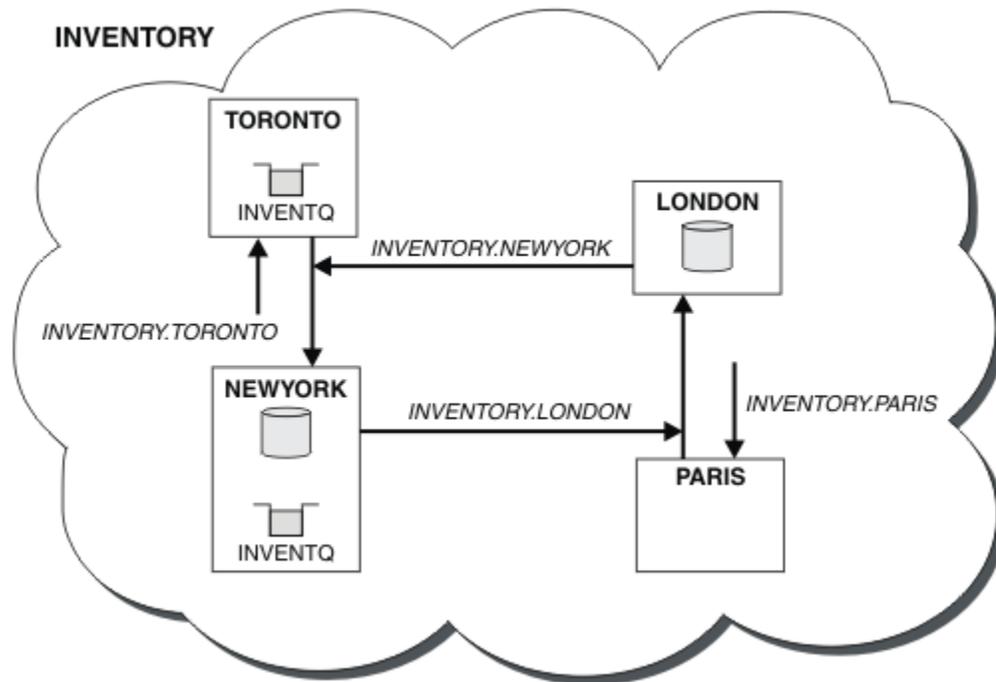


Abbildung 43. Der INVENTORY -Cluster mit vier Warteschlangenmanagern

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 Repositories 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 Accountdatensatz 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.

## 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 Repositories 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 Cluster INVENTORY wurde wie in „Neuen Cluster einrichten“ auf Seite 268 beschrieben konfiguriert. 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 Repositories 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 Repositories.

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 Cluster-senderkanaldefinition verfügt.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(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 Empfangsprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

Abbildung 44 auf Seite 292 zeigt den durch diese Task konfigurierten Cluster.

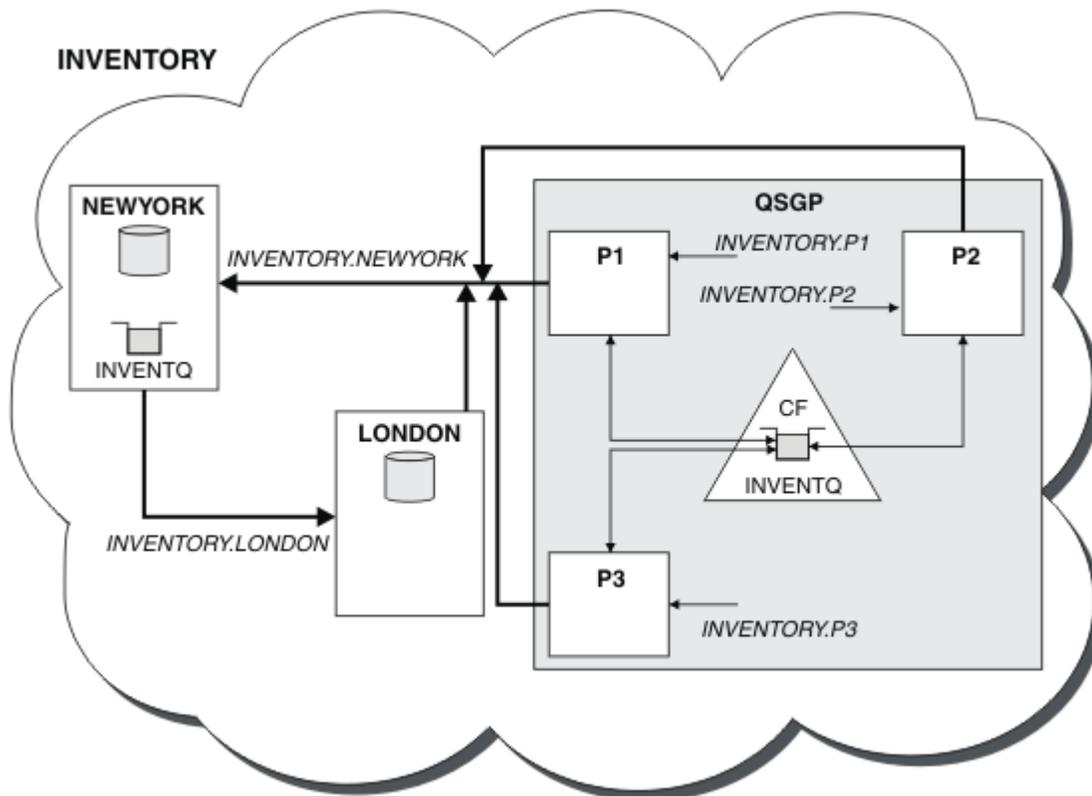


Abbildung 44. Cluster und Gruppe mit gemeinsamer Warteschlange

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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „[WS-Manager zu einem Cluster hinzufügen](#)“ auf Seite 280 beschrieben konfiguriert.
- 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. Nachdem Sie einen Kanal gelöscht haben, geben Sie immer den Befehl **start channel** aus, 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. Beispiel:

```
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: Unerwartete Veröffentlichung eines Clusterwarteschlangenobjekts empfangen
- AMQ9432: Abfrage von einem Nicht-Repository-Warteschlangenmanager empfangen

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 45](#) auf Seite 295 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

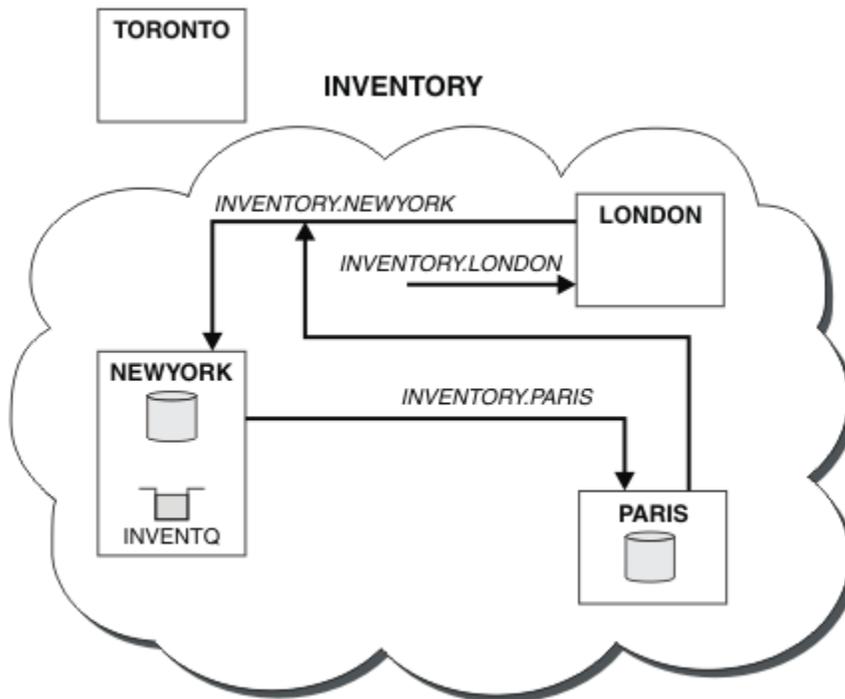


Abbildung 45. Der INVENTORY -Cluster mit dem vollständigen Repository wurde in PARIS verschoben.

## 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. Die unterstützten Protokolle sind TCP oder LU 6.2 auf einer beliebigen Plattform und 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

Für alle Cluster-WS-Manager ist ein Kanalinitiator erforderlich, 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. Auf IBM MQ for IBM i-, Windows- und UNIX and Linux-Systemen 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`.

 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.

• 

### IBM MQ for z/OS

Für jeden WS-Manager gibt es einen Kanalinitiator, der als separater Adressraum ausgeführt wird. Sie starten ihn mit dem Befehl `MQSC START CHINIT`, den Sie beim Start des Warteschlangenmanagers ausgeben.

• 

### IBM MQ for UNIX, Linux, and Windows

Wenn Sie einen WS-Manager starten, wird 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 MQ for IBM i

Wenn Sie einen WS-Manager starten, wird 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.

• 

### IBM MQ for z/OS

Verwenden Sie das von IBM MQ bereitgestellte Kanallistener-Programm. Wenn Sie einen IBM MQ-Kanallistener starten möchten, verwenden Sie den `MQSC`-Befehl `START LISTENER`, den Sie als Teil Ihres Kanalinitiatorstarts ausgeben. Beispiel:

```
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 `CONNNAME` 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

### IBM MQ for IBM i

Verwenden Sie das von IBM MQ bereitgestellte Kanallistener-Programm. Verwenden Sie den **CL**-Befehl STRMQMLSR, um einen IBM MQ-Kanallistener zu starten. Beispiel:

```
STRMQMLSR MQMNAME(QM1) PORT(1414)
```

## Windows

### IBM MQ for Windows

Verwenden Sie entweder das von IBM MQ bereitgestellte Kanallistener-Programm oder die vom Betriebssystem bereitgestellten Funktionen.

Verwenden Sie den Befehl RUNMQLSR, um den IBM MQ-Kanallistener zu starten. Beispiel:

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

## Linux

## UNIX

### IBM MQ unter UNIX and Linux

Verwenden Sie entweder das von IBM MQ bereitgestellte Kanallistener-Programm oder die vom Betriebssystem bereitgestellten Funktionen (z. B. **inetd** für die TCP-Kommunikation).

Verwenden Sie den Befehl **runmqclsr**, um den IBM MQ-Kanallistener zu starten. Beispiel:

```
runmqclsr -t tcp -p 1414 -m QM1
```

Wenn Sie **inetd** zum Starten von Kanälen verwenden möchten, konfigurieren Sie zwei Dateien:

- 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.

- 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/amqcrista amqcrista  
-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:

Unter AIX:

```
refresh -s inetd
```

Unter HP-UX:

```
inetd -c
```

Unter Solaris oder Linux:

a. Suchen Sie die Prozess-ID von **inetd** mit dem Befehl:

```
ps -ef | grep inetd
```

b. Führen Sie den entsprechenden Befehl aus:

– Für Solaris 9 und Linux:

```
kill -1 inetd processid
```

– Für Solaris 10 oder höhere Versionen:

```
inetconv
```

## Konvertieren eines vorhandenen Netzes in einen Cluster

Konvertieren Sie ein existierendes 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 268 über „Vollrepositorys in einen anderen WS-Manager verschieben“ auf Seite 293 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 Repositories 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 46 auf Seite 299](#) dargestellt.

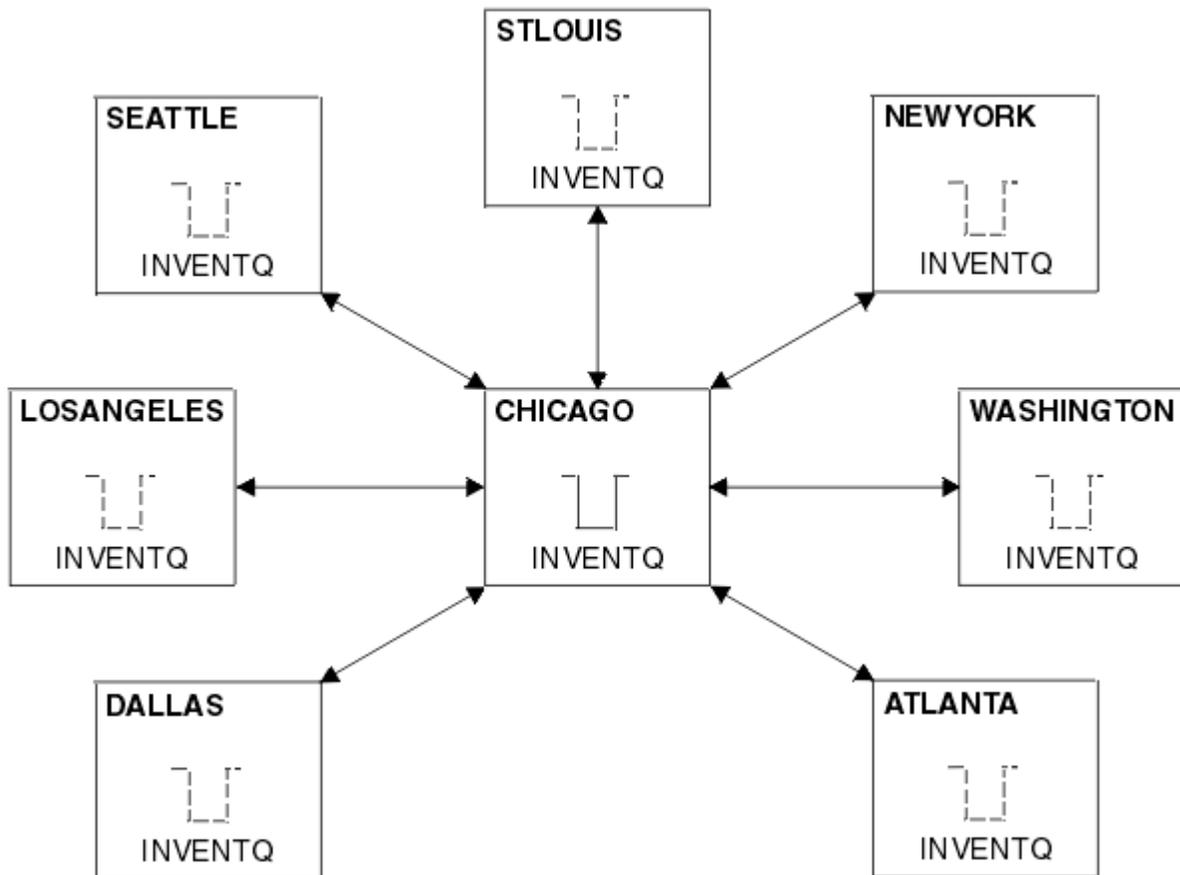


Abbildung 46. Ein Hub-und Spoke-Netz

- 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 so ausgewählt, dass die Namen von Clusterempfängerkanälen mit Namen im Format `clustername.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 379.

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 Repositories 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-Repositories 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 **CSQUITL** verwenden, um auf CHICAGO2 eine exakte Kopie von INVENTQ auf CHICAGO zu erstellen.

Wenn Sie diese Definitionen erstellen, wird eine Nachricht an die vollständigen Repositories in CHICAGO und CHICAGO2 gesendet, und die darin enthaltenen Informationen werden aktualisiert. Der Warteschlangenmanager ermittelt aus den vollständigen Repositories, wenn er eine Nachricht an den INVENTQ einreicht, 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 Repositories 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 Warteschlangennamen einrichten, bevor Sie fortfahren können.

Szenario:

- Ein IBM MQ-Cluster wurde wie in „[Konvertieren eines vorhandenen Netzes in einen Cluster](#)“ auf Seite 298 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 Repositories 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 Repositories 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 Repositories für den Cluster CHNSTORE enthalten. Ändern Sie diese Definition, um anzuzeigen, dass die WS-Manager vollständige Repositories

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 47](#) auf Seite 304 angezeigt.

Jetzt haben wir zwei überlappende Cluster. Die vollständigen Repositories 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 Warteschlangennamenskollisionen 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 einreicht, kann die Nachricht in eine der beiden Instanzen der ACCOUNTQ gestellt werden.

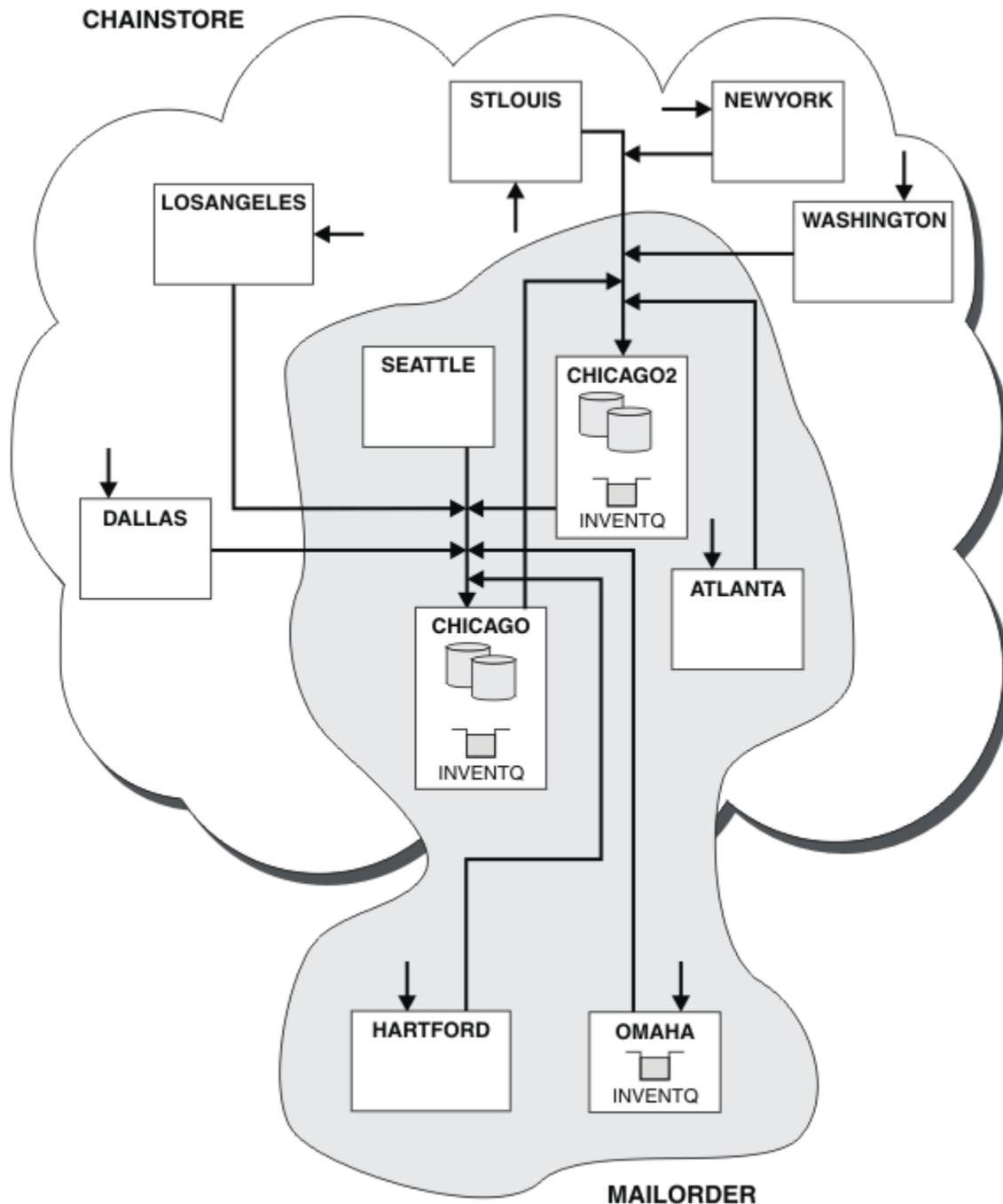


Abbildung 47. Interverbundene Cluster

### 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 Repositories enthalten:

- Ändern Sie die WS-Manager-Definitionen für CHICAGO und CHICAGO2, entfernen Sie das Attribut REPOSNL, das die Namensliste (CHAINMAIL) angibt, und ersetzen Sie es durch ein REPOS-Attribut, das den Clusternamen (CHNSTORE) angibt. 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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Ein IBM MQ-Cluster wurde wie in [„Konvertieren eines vorhandenen Netzes in einen Cluster“](#) auf Seite 298 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 MQ00\_BIND\_NOT\_FIXED verwenden, und Anwendungen, die MQ00\_BIND\_AS\_Q\_DEF verwenden, auf denen die Warteschlange mit DEFBIND(NOTFIXED) definiert wurde, schlagen bei der nächsten versuchten MQPUT -oder MQPUT1 -Anforderung 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(' ')\nALTER 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)\nSTOP 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 337.
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 48](#) auf Seite 308 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.

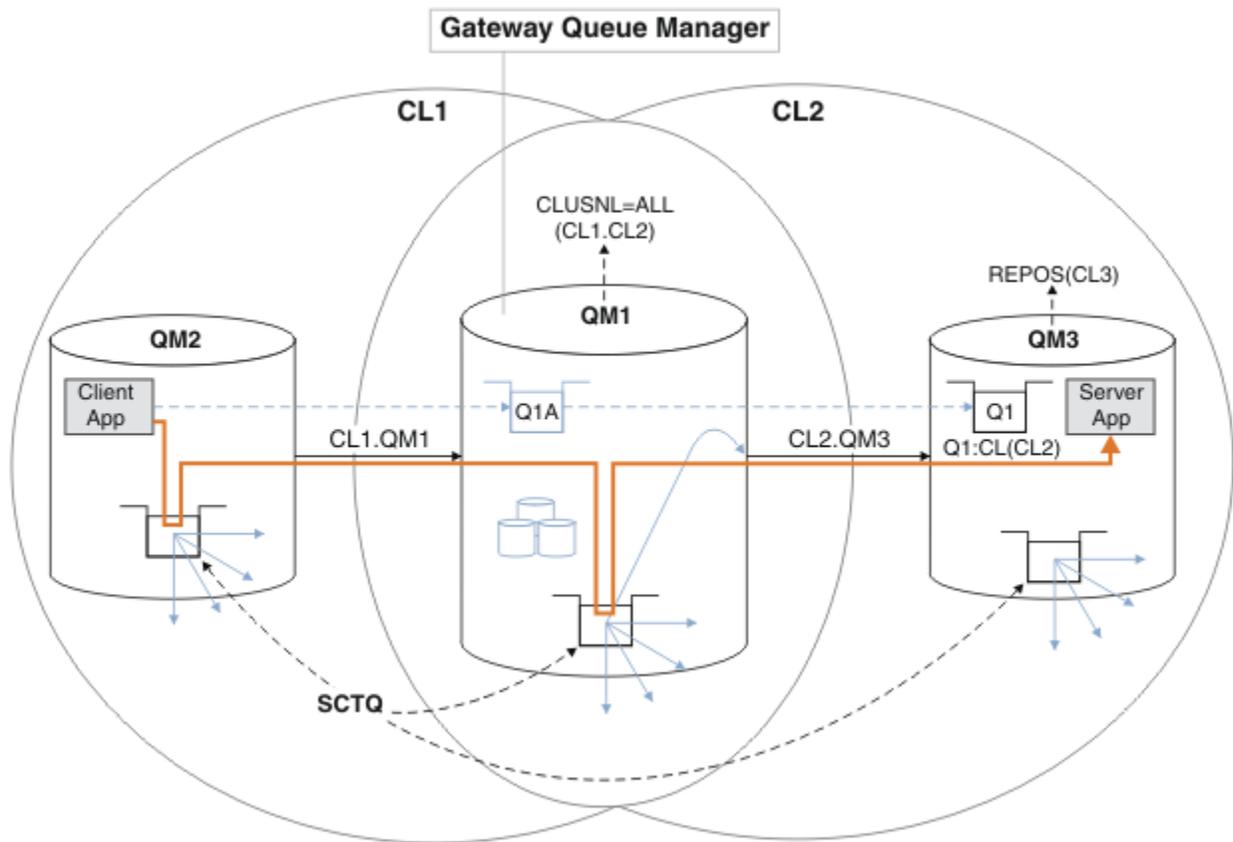


Abbildung 48. In Hub- und Spoke-Architektur mit IBM MQ-Clustern implementierte Client/Server-Anwendung

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 Repositories 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 Sicherheitsrepositories 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 in QM2, QM3, QM4 und QM5 aus, wobei *c*, *n* und *m* die in [Tabelle 24](#) auf Seite 309 für jeden Warteschlangenmanager angezeigten Werte übernehmen:

*Tabelle 24. Parameterwerte für die Erstellung von Clustern 1 und 2*

Warteschlangenmanager	Cluster <i>c</i>	Anderes Repository <i>n</i>	Dieses Repository <i>m</i>
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 der 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 in Cluster CL2 hinzu.

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. Fügen Sie den Aliasnamen des Clusterwarteschlangenmanagers Q1A zum Warteschlangenmanager des Gateways hinzu.

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

**Anmerkung:** Anwendungen, die den Warteschlangenmanager-Aliasnamen 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 OPEN gesetzt ist, werden Nachrichten an Q1@QM1 weitergeleitet. Q1@QM1 ist nicht vorhanden, so dass 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, der von der Antwortnachricht in eine temporäre dynamische Warteschlange zurückgenommen wird, die als RQ bezeichnet wird. Die Serveranwendung, die mit QM3 verbunden ist, öffnet die Antwortwarteschlange unter Verwendung des Warteschlangenmanagernamens QM2. Der Name des Warteschlangenmanagers QM2 ist in QM1 als Aliasname für einen Clusterwarteschlangenmanager definiert. QM3 leitet die Antwortnachricht an QM1 weiter. QM1 leitet die Nachricht an QM2 weiter.

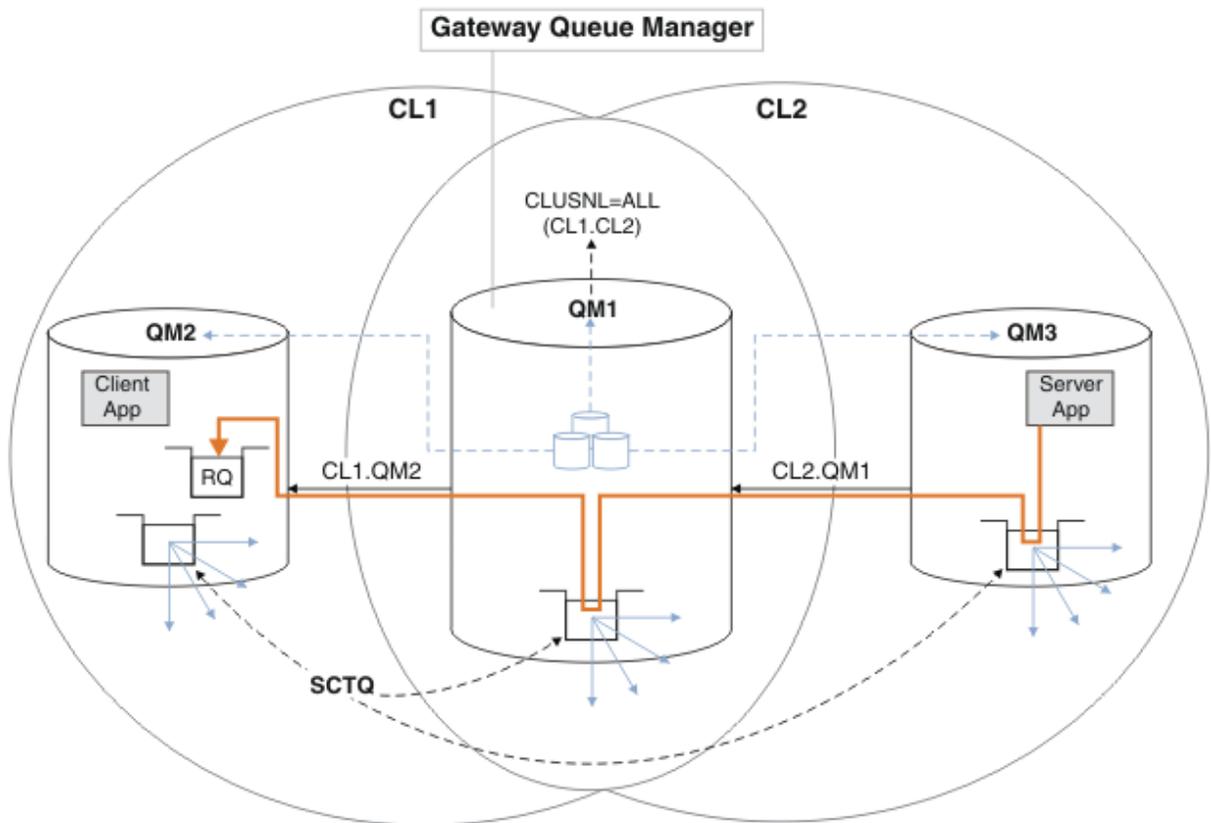


Abbildung 49. Verwenden eines Warteschlangenmanager-Aliasnamens, um die Antwortnachricht an einen anderen Cluster zurückzugeben

Die Art und Weise, wie das Routing funktioniert, ist wie folgt. Jeder Warteschlangenmanager in jedem Cluster verfügt über eine Definition des Warteschlangenmanager-Aliasnamens in 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 in beiden Clustern CL1 und CL2 in Gruppen zusammengefasst und in CL1 in den realen Warteschlangenmanager QM2 aufgelöst. Die Serveranwendung erstellt die Antwortnachricht unter Verwendung der Antwort auf den Warteschlangennamen RQ und beantwortet den Warteschlangenmanagername QM2. Die Nachricht wird an QM1 weitergeleitet, weil die Warteschlangenmanager-Aliasdefinition QM2 in QM1 in Cluster CL2 definiert ist und der Warteschlangenmanager QM2 nicht in Cluster CL2 enthalten 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 auf QM1, damit sie an QM2 übertragen werden können. QM1 leitet die Nachricht an QM2 weiter, da die Warteschlangenmanager-Aliasdefinition in QM1 für QM2 als realen Zielwarteschlangenmanager definiert ist. 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 sucht die Verbindungsinformationen für QM2 aus dem Repository für CL1 heraus und leitet die Nachricht an QM2 weiter. Damit die Nachricht von QM1 umgeleitet werden kann, muss die Serveranwendung die Antwortwarteschlange geöffnet haben, in der die Option `DEFBIND` auf `MQBND_BIND_NOT_FIXED` gesetzt ist. Wenn die Serveranwendung die Antwortwarteschlange mit der Option `MQBND_BIND_ON_OPEN` geöffnet hatte, wird die Nachricht nicht weitergeleitet und wird in einer Warteschlange für nicht zustellbare Nachrichten nicht mehr angezeigt.

- a. Erstellen Sie eine Cluster-Anforderungswarteschlange mit einem Auslöser auf QM3.

\*... On QM3

```
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE) PRO  
CESS(ECHO) REPLACE
```

- b. Erstellen Sie eine Clusterwarteschlangenaliasdefinition von 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 QM3 zu 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, 2023. 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 Tasks

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen“ auf Seite 282  
Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

### Zugehörige Informationen

[Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen](#)

[Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen](#)

[Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen](#)

## **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-Architektur mit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung](#) im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager“ auf Seite 307 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 50 auf Seite 314](#) 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 316. Außerdem müssen Sie den Switch von einer Übertragungswarteschlange in die andere verwalten.

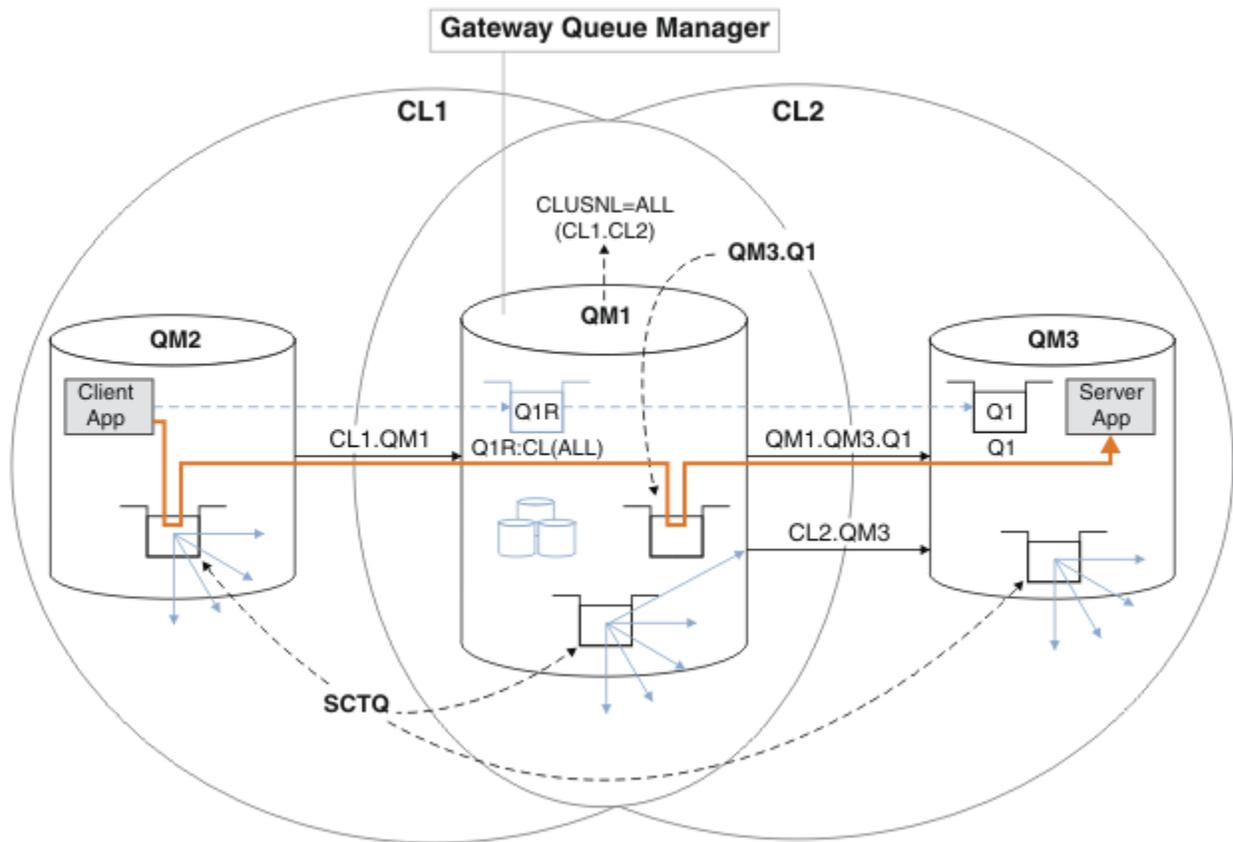


Abbildung 50. Anwendung 'Client-Server' in Hub-und Spoke-Cluster-Architektur unter Verwendung ferner Warteschlangendefinitionen implementiert

## 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 **amqspu**t in QM2 aus, um eine Nachricht zu erstellen.

```
C:\IBM\MQ>amqspu Q1R QM2
Start Beispiel AMQSPUT0
Zielwarteschlange ist Q1R
Musteranforderungsnachricht von QM2 an Q1 mit Q1R
```

Beispiel für AMQSPUT0-Ende

2. Führen Sie das Beispielprogramm **amqsge**t aus, um die Nachricht von Q1 auf QM3 abzurufen.

```
C:\IBM\MQ>amqsge Q1 QM3
Start Beispiel AMQSGET0
Nachricht &lt;Beispielanforderungsnachricht von QM2 an Q1 mit Q1R>
Keine weiteren Nachrichten
Ende Beispiel AMQSGET0
```

## 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.

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen“ auf Seite 282

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

## Zugehörige Informationen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

Zugriffsteuerung und mehrere Clusterübertragungswarteschlangen

## ***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-WS-Manager muss IBM WebSphere MQ 7.5 oder höher entsprechen.
2. Erstellen Sie die in In Hub-and-Spoke-Architektur mit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager“ auf Seite 307 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 51 auf Seite 317).

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 313 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 51 auf Seite 317 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 313 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 51 auf Seite 317 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 319.

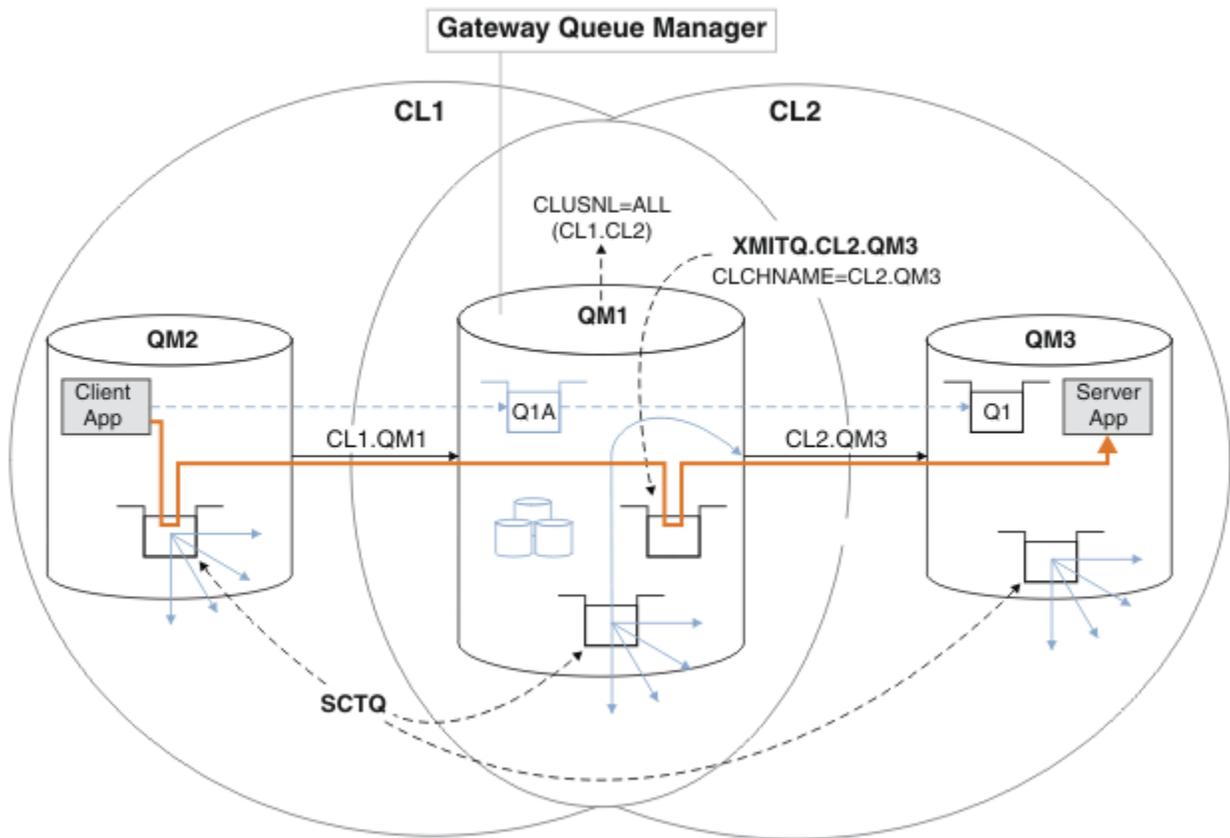


Abbildung 51. Die Client-Server-Anwendung wurde in Hub-und Spoke-Architektur unter Verwendung einer zusätzlichen Clusterübertragungswarteschlange implementiert.

## 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: Stoppen von IBM MQ-Kanal akzeptiert.

- 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: Details des Kanalstatus anzeigen.  
CHANNEL (CL2.QM3) CHLTYPE (CLUSSDR)  
CONNAME( 127.0.0.1 (1413)) CURRENT  
RQMNAME (WSM3) 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: Starten von IBM MQ-Kanal akzeptiert.
```

d) Überprüfen Sie, ob der Kanal gestartet wurde

```
*... On QM1  
DISPLAY CHSTATUS(CL2.QM3)
```

Die Antwort ist eine Zusammenfassung des Kanalstatus:

```
AMQ8417: Details des Kanalstatus anzeigen.  
CHANNEL (CL2.QM3) CHLTYPE (CLUSSDR)  
CONNAME( 127.0.0.1 (1413)) CURRENT  
RQMNAME (WSM3) 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 **amqspout** unter QM2 aus, um eine Nachricht einzureihen.

```
C:\IBM\MQ>amqspout 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

„Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen“ auf Seite 261  
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.

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen“ auf Seite 282

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

## Zugehörige Informationen

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

## ***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 51 auf Seite 317](#) dargestellte Konfiguration zu ändern.

1. Der Gateway-WS-Manager muss IBM WebSphere MQ 7.5 oder höher entsprechen.
2. Erstellen Sie die in [In Hub-and-Spoke-Architektur mit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-](#)

Warteschlangenmanager” auf Seite 307 dargestellten überlappenden Cluster, indem Sie die in dieser Task genannten Schritte ausführen.

3. Führen Sie die Schritte unter [Abbildung 51 auf Seite 317](#) in „Cluster-Übertragungswarteschlange zum Isolieren des Clusternachrichtenverkehrs hinzufügen, der von einem Gateway-Warteschlangenmanager gesendet wurde” auf Seite 316 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 316 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 316, 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 52 auf Seite 321](#) 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 Repositories 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 316 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-Clusterübertragungswarteschlange zurückzusetzen, verwaltet der Gateway-Warteschlangenmanager den Switch automatisch für Sie.

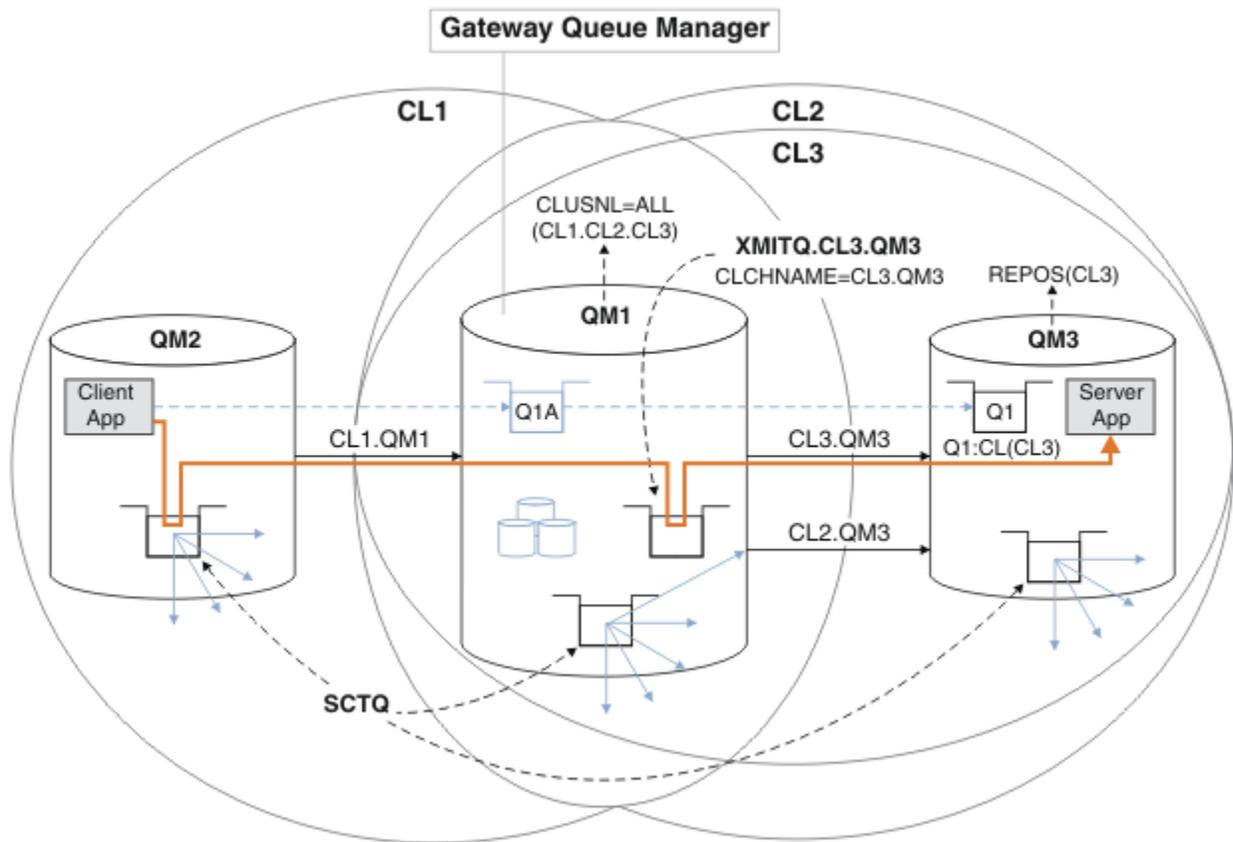


Abbildung 52. Verwenden eines zusätzlichen Clusters zum Trennen des Nachrichtenverkehrs im Gateway-Warteschlangenmanager, der zu einer Anzahl von Clusterwarteschlangen auf demselben Warteschlangenmanager wechselt

## 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 NAMLIST(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 QM3 zum Gateway-Warteschlangenmanager QM1 hinzu.

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)
CONNNAME(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)  
CONNNAME(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 **amqspout** unter QM2 aus, um eine Nachricht einzureihen.

```
C:\IBM\MQ>amqspout 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

„Arbeiten mit Clusterübertragungswarteschlangen und Clustersenderkanälen“ auf Seite 261

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 Clus-

ternachrichten 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.

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen“ auf Seite 282

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

### **Zugehörige Informationen**

Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen

Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen

Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen

### **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-WS-Manager muss IBM WebSphere MQ 7.5 oder höher entsprechen.
2. Erstellen Sie die in In Hub-and-Spoke-Architektur mit Hilfe von IBM MQ-Clustern implementierte Client/Server-Anwendung im Abschnitt „Erstellen von zwei überlappenden Clustern mit einem Gateway-Warteschlangenmanager“ auf Seite 307 dargestellten überlappenden Cluster, indem Sie die in dieser Task genannten Schritte ausführen.

### **Informationen zu diesem Vorgang**

Wenn Sie die Architektur mit mehreren Clusterwarteschlangen implementieren wollen, muss Ihr Gateway-Warteschlangenmanager IBM WebSphere MQ 7.5 oder höher entsprechen. Alle für die Verwendung mehrerer Clusterübertragungswarteschlangen verwendeten Warteschlangen müssen den Standardwarteschlangentyp der Clusterübertragung im Gateway-Warteschlangenmanager ändern. Ändern Sie den Wert des Warteschlangenmanagerattributs **DEFCLXQ** in QM1 von SCTQ in CHANNEL ; weitere Informationen finden Sie im Abschnitt Abbildung 53 auf Seite 326. 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 282.

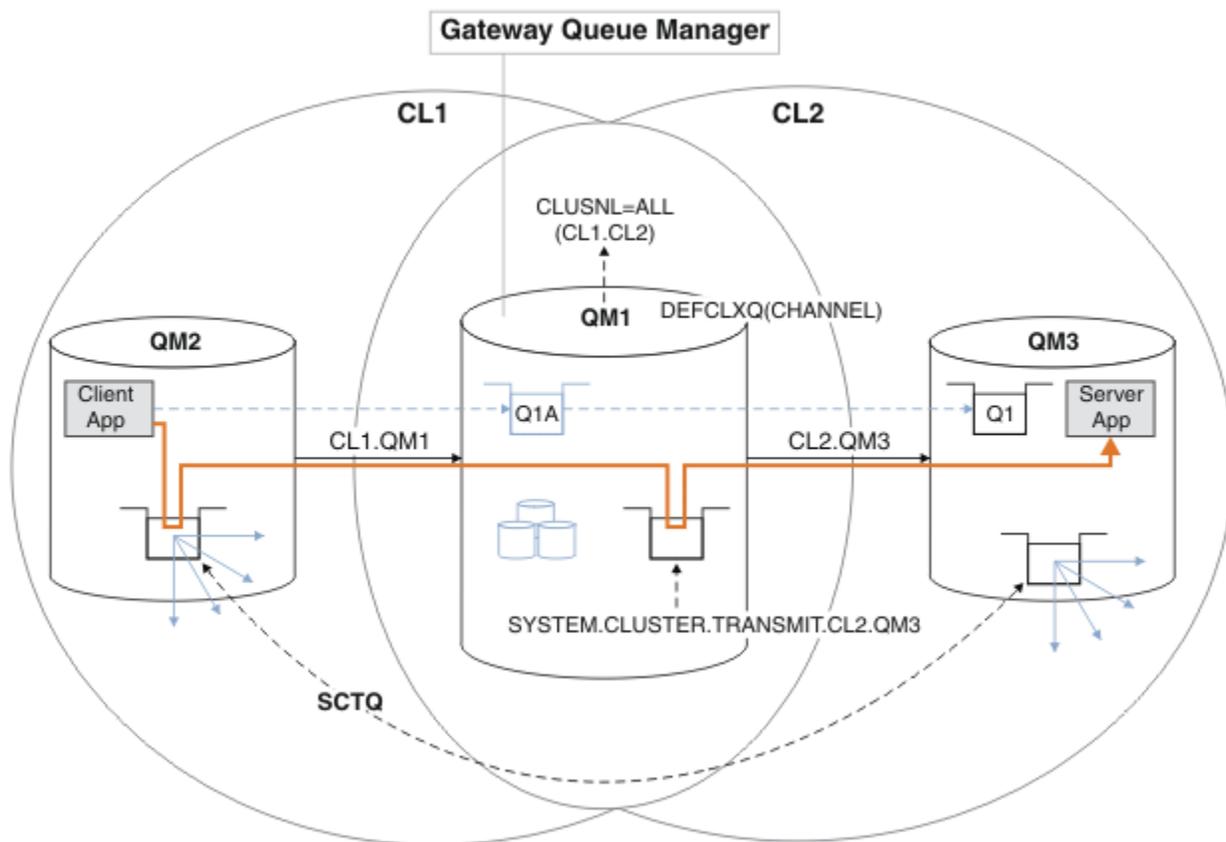


Abbildung 53. Die Client-Server-Anwendung, die in Hub-und Spoke-Architektur implementiert ist, mit separaten Clusterübertragungswarteschlangen auf dem Gateway-Warteschlangenmanager.

## 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: Details des Kanalstatus anzeigen.
CHANNEL (CL1.QM2) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1412)) CURRENT
RQMNAME (WSM2) STATUS (RUNNING)
```

```

SUBSTATE (MQGET) XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL2.QM3) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1413)) CURRENT
RQMNAME (WSM3) STATUS (RUNNING)
SUBSTATE (MQGET) XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL2.QM5) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1415)) CURRENT
RQMNAME (WSM5) STATUS (RUNNING)
SUBSTATE (MQGET) XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL1.QM4) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1414)) CURRENT
RQMNAME (WSM4) 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: Stoppen von IBM MQ-Kanal akzeptiert.

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: Details des Kanalstatus anzeigen.
CHANNEL (CL1.QM2) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1412)) CURRENT
RQMNAME (WSM2) STATUS (STOPPED)
SUBSTATE ( ) XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL2.QM3) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1413)) CURRENT
RQMNAME (WSM3) STATUS (STOPPED)
SUBSTATE ( ) XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL2.QM5) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1415)) CURRENT
RQMNAME (QM5) STATUS (STOPPED)
SUBSTATE ( ) XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL1.QM4) CHLTYPE (CLUSSDR)
CONNNAME( 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, 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 CHANNEL (CL2.QM5)
```

Die Antwort ist, dass der Befehl akzeptiert wird:

AMQ8018: Starten von IBM MQ-Kanal akzeptiert.

- 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: Kanalstatus nicht gefunden.
AMQ8409: Warteschlangendetails anzeigen.
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: Details des Kanalstatus anzeigen.
CHANNEL (CL1.QM2) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1412)) CURRENT
RQMNAME (WSM2) STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL2.QM3) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1413)) CURRENT
RQMNAME (WSM3) STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Details des Kanalstatus anzeigen.
CHANNEL (CL2.QM5) CHLTYPE (CLUSSDR)
CONNNAME( 127.0.0.1 (1415)) CURRENT
RQMNAME (WSM5) STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
```

```
AMQ8417: Details des Kanalstatus anzeigen.  
CHANNEL (CL1.QM4) CHLTYPE (CLUSSDR)  
CONNNAME ( 127.0.0.1 (1414)) CURRENT  
RQMNAME (WSM4) 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 **amqspu**t unter QM2 aus, um eine Nachricht einzureihen.

```
C:\IBM\MQ>amqspu 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 **amqsge**t aus, um die Nachricht von Q1 unter QM3 abzurufen

```
C:\IBM\MQ>amqsge 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 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 Clusternachrichten zu verwenden. Die Lösung verwendet einen zusätzlichen Cluster, um die Nachrichten in einer bestimmten Clusterwarteschlange zu isolieren.

„WS-Manager zu einem Cluster hinzufügen: separate Übertragungswarteschlangen“ auf Seite 282

Befolgen Sie diese Anweisungen, um dem erstellten Cluster einen WS-Manager hinzuzufügen. Nachrichten zu Clusterwarteschlangen und Themen werden unter Verwendung mehrerer Clusterübertragungswarteschlangen übertragen.

### Zugehörige Informationen

[Zugriffssteuerung und mehrere Clusterübertragungswarteschlangen](#)

[Clustering: Anwendungsisolation mit mehreren Clusterübertragungswarteschlangen](#)

[Clustering: Planung der Konfiguration von Clusterübertragungswarteschlangen](#)

## 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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „[Hinzufügen eines Warteschlangenmanagers, der eine Warteschlange enthält](#)“ auf Seite 288 beschrieben konfiguriert. Sie enthält vier WS-Manager. LONDON und NEWYORK enthalten vollständige Repositories. PARIS und TORONTO enthalten Teilrepositories. 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 in TORONTO so, dass sie vom 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 MQ00\_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 und geben Sie dabei die Attribute IPPROCS, OPROCS und CURDEPTH an, bzw. verwenden Sie unter IBM i den Befehl **WRKMQMSTS**. Wenn die Anzahl der Ein- und Ausgabeprozesse und die aktuelle Tiefe der Warteschlangenmanager alle null sind, ist die Warteschlange leer.

5. Überwachen Sie den Kanal, um sicherzustellen, dass keine unbestätigten Nachrichten vorhanden sind.

Um sicherzustellen, dass keine unbestätigten 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 Inventuranwendung unter TORONTO zugestellt werden sollen, können Sie die Warteschlange löschen:

```
DELETE QLOCAL(INVENTQ)
```

7. Sie können die Inventuranwendung 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 einreicht, 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 Cluster-namen 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 334.

### 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 280 beschrieben konfiguriert und wie in [„Clusterwarteschlange aus einem WS-Manager entfernen“](#) auf Seite 330 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 beitrifft, 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-Mitgliedern 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-Mitglieder 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 293 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 330 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 390 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)
```

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 280 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 337 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 Repositories 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 gängigste Methode: [„Warteschlangenmanager aus einem Cluster entfernen: Best Practice“](#) auf Seite 332.

## 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 280 beschrieben konfiguriert und wie in [„Clusterwarteschlange aus einem WS-Manager entfernen“](#) auf Seite 330 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 beitrifft, 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-Mitgliedern 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-Mitglieder 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 293 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 330 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 [„Clusterthemen-Definition in einen anderen WS-Manager verschieben“](#) auf Seite 390 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 280 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 337 beschrieben.

### Zugehörige Verweise

[DELETE CHANNEL](#)

[ANZEIGEN CHANNEL](#)

[ANZEIGEN CHSTATUS](#)

[DISPLAY CLUSQMGR](#)

[STOP CHANNEL](#)

[RESET CLUSTER](#)

## 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. Beispiel:

- 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 332 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)
```

- Überwachen Sie den Warteschlangenmanager, bis alle Clusterressourcen nicht mehr verfügbar sind:

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)  
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)  
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

### Zugehörige Informationen

[Cluster](#)

[Vergleich von Clustering und verteilter Steuerung von Warteschlangen](#)

[Clusterkomponenten](#)

## 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

- Setzen Sie den Befehl `SUSPEND QMGR runmqsc` aus, um einen WS-Manager zu sperren:

```
SUSPEND QMGR CLUSTER(SALES)
```

Der Befehl `SUSPEND runmqsc` benachrichtigt die WS-Manager im SALES -Cluster, 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.

- 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)` .

- Führen Sie die erforderlichen Wartungsaufgaben aus.
- Führen Sie den Befehl `RESUME QMGR runmqsc` aus, um den WS-Manager wieder zu aufnehmen:

## Ergebnisse

Der Befehl RESUME **runmqsc** benachrichtigt die vollständigen Repositories, die der WS-Manager wieder verfügbar ist. Die vollständigen WS-Manager-Repositories 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.  Setzen Sie unter z/OS den Wert für INDXTYPE einer Clusterübertragungswarteschlange auf CORRELID.

## Vorbereitende Schritte

- Stellen Sie sicher, dass die Clusterübertragungswarteschlange nicht voll ist.
- Stellen Sie sicher, dass Sie keinen ALTER **runmqsc** -Befehl ausgeben, um ihn entweder inaktiviert zu setzen oder versehentlich inaktiviert zu werden.
- Stellen Sie sicher, dass das Medium, in dem die Clusterübertragungswarteschlange auf  gespeichert ist (z. B. z/OS-Seitengruppen), nicht voll wird.

## Informationen zu diesem Vorgang



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. Beispiel: Bei großen Clustern kann der Befehl **REFRESH CLUSTER** den Cluster während seiner Ausführung unterbrechen, 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

Setzen Sie den REFRESH CLUSTER **MQSC** -Befehl von einem Warteschlangenmanager ab, um automatisch definierten Cluster-WS-Manager 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

Der Befehl `CLUSTER AKTUALISIEREN` ermöglicht es, dass ein Warteschlangenmanager im Hinblick auf seinen vollständigen Repository-Inhalt kalt gestartet wird. IBM MQ stellt sicher, dass keine Daten aus Ihren Warteschlangen verloren gehen.

### Zugehörige Informationen

[Clustering: Best Practices für REFRESH CLUSTER verwenden](#)

## Cluster-WS-Manager wiederherstellen

Führen Sie mit dem Befehl `REFRESH CLUSTER runmqsc` die Clusterinformationen zu einem Warteschlangenmanager auf dem neuesten Stand aus. Führen Sie diese Prozedur aus, nachdem Sie einen Warteschlangenmanager aus einer Zeitpunktsicherung wiederhergestellt haben.

### Vorbereitende Schritte

Sie haben einen Clusterwarteschlangenmanager aus einer zeitpunktbasierter Sicherung wiederhergestellt.

### Informationen zu diesem Vorgang

Wenn Sie einen Warteschlangenmanager in einem Cluster wiederherstellen möchten, müssen Sie den Warteschlangenmanager wiederherstellen und anschließend die Clusterinformationen mit dem Befehl `REFRESH CLUSTER runmqsc` auf den neuesten Stand bringen.

**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 Informationen

[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 KAINTE 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.

#### **Unter IBM MQ for z/OS**

Steuern Sie das Problem der Clusterempfängerseite des Kanals, der aktiv bleibt, mithilfe der Parameter ADOPTMCA und ADOPTCHK unter ALTER QMGR.

#### **Unter Multiplatforms**

Steuern Sie das Problem der Clusterempfängerseite des verbleibenden aktiven Kanals mit den Attributen AdoptNewMCA, AdoptNewMCATimeout und AdoptNewMCACheck in der Datei qm.ini oder in der Windows NT -Registry.

## 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

[„WS-Manager-Aliasnamen und -Cluster“ auf Seite 355](#)

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 359](#)

Verwenden Sie Warteschlangenaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffsteuerungen zu übernehmen.

[„Aliasnamen für Antwortwarteschlangen und Cluster“ auf Seite 359](#)

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 255

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 268

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 Informationen

[Cluster](#)

[Vergleich von Clustering und verteilter Steuerung von Warteschlangen](#)

[Komponenten eines Clusters](#)

## **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 54 auf Seite 343 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)
```

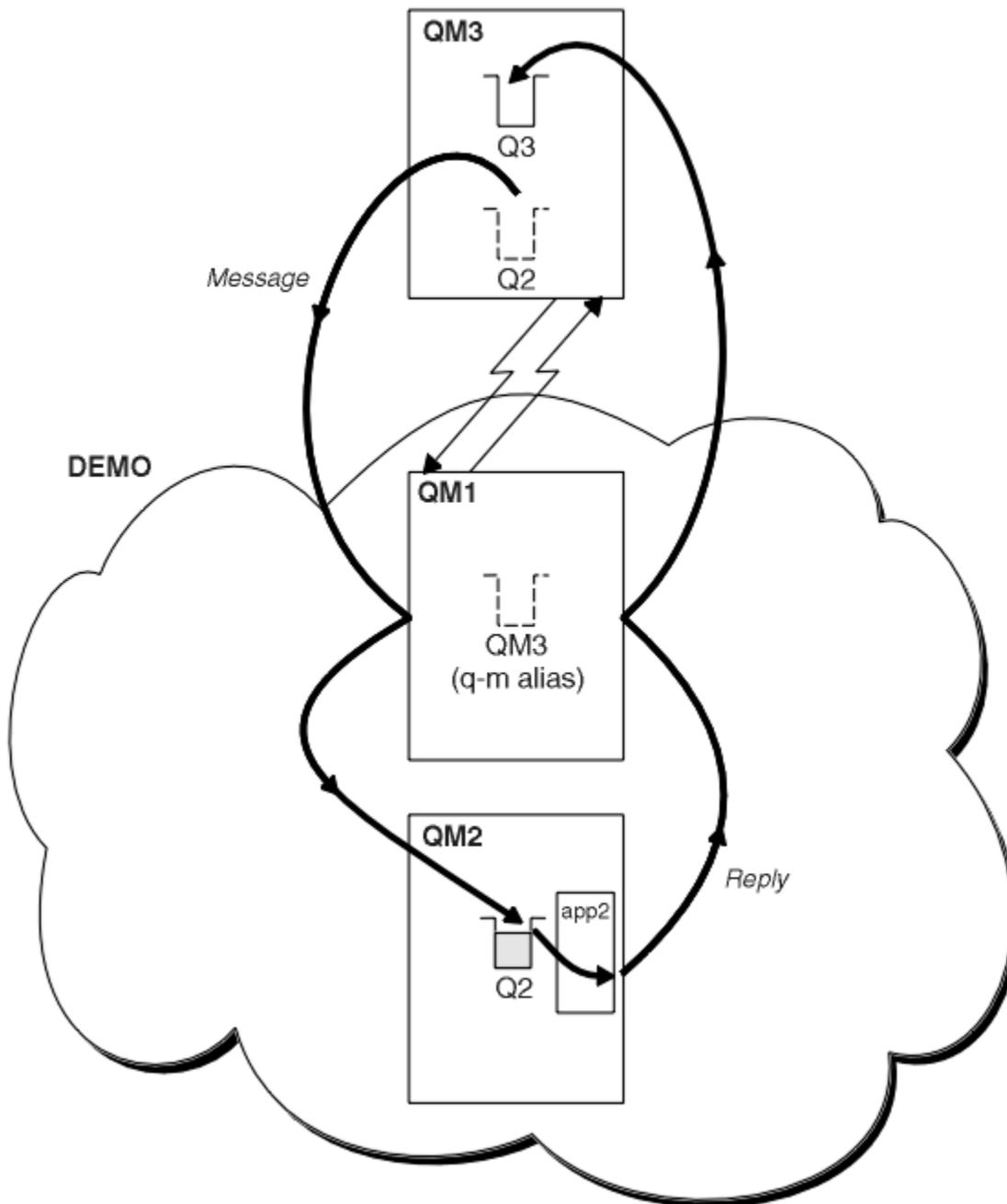


Abbildung 54. Aus einem WS-Manager außerhalb des Clusters einschalten

### 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 einreicht.

- 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 54 auf Seite 343](#) nicht explizit angezeigt.

In dem Beispiel gibt eine Anwendung in QM3 einen MQPUT -Aufruf aus, um eine Nachricht in Q2 zu stellen. 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. Der Name der Übertragungswarteschlange ist QM3, die in der Definition von QREMOTE durch das Auslassen des Attributs XMITQ implizit enthalten ist. QM3 ist jedoch derselbe Name, den wir voraussichtlich für den Rest des Clusters mit dem Warteschlangenmanager-Aliasnamen bewerben möchten. IBM MQ lässt Sie nicht zu, dass sowohl die Übertragungswarteschlange als auch der Warteschlangenmanager mit dem gleichen Namen angegeben werden. 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 adressieren. Die Lösung, die QM1 benötigt, um QM3 als WS-Manager-Aliasnamen zu bewerben, um Antwortnachrichten an QM1 zurückzugeben und zu verhindern, dass QM1 als Name der Übertragungswarteschlange verwendet wird.

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, der durch MQPUT in der MQMD erstellt wurde. 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 Antwortwarteschlangentalias ( 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 Antwortwarteschlangen 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 anderen Parameter angeben, der den Aliasnamen der Empfangswarteschlange für Antworten enthält, um den Aliasnamen des Antwortwarteschlangenmanagers für MQPUT zu 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.

#### Warteschlangentaliasnamen und -cluster

Verwenden Sie Warteschlangentaliasnamen, um den Namen einer Clusterwarteschlange zu verdecken, eine Warteschlange zu einem Cluster zu machen, andere Attribute zu übernehmen oder andere Zugriffsteuerungen 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 346

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 346 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 342, um den Namen des Zielwarteschlangenmanagers innerhalb des Clusters zu verdecken.

## Vorgehensweise

Definieren Sie im Beispiel Abbildung 55 auf Seite 347 einen WS-Manager-Aliasnamen auf dem Gateway-WS-Manager QM1 mit dem Namen DEMO:

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

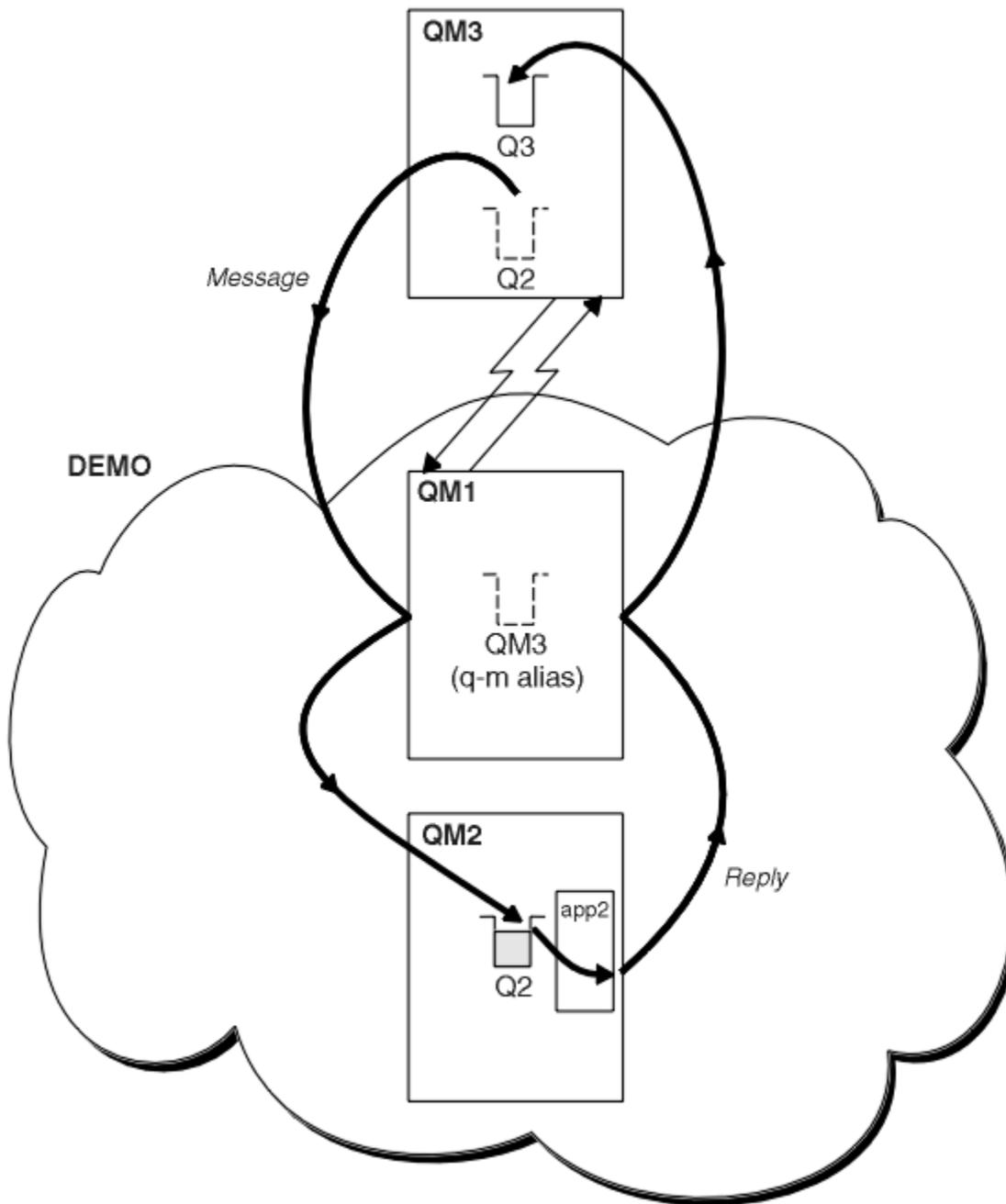


Abbildung 55. Aus einem WS-Manager außerhalb des Clusters einschalten

Die QREMOTE -Definition in QM1 macht den WS-Manager-Aliasnamen DEMO dem Gateway-Warteschlangenmanager bekannt. QM3, der Warteschlangenmanager außerhalb des Clusters, kann den WS-Manager-Aliasnamen DEMO verwenden, um Nachrichten an Clusterwarteschlangen unter DEMO zu senden, anstatt einen tatsächlichen WS-Managernamen 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 56 auf Seite 349 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.

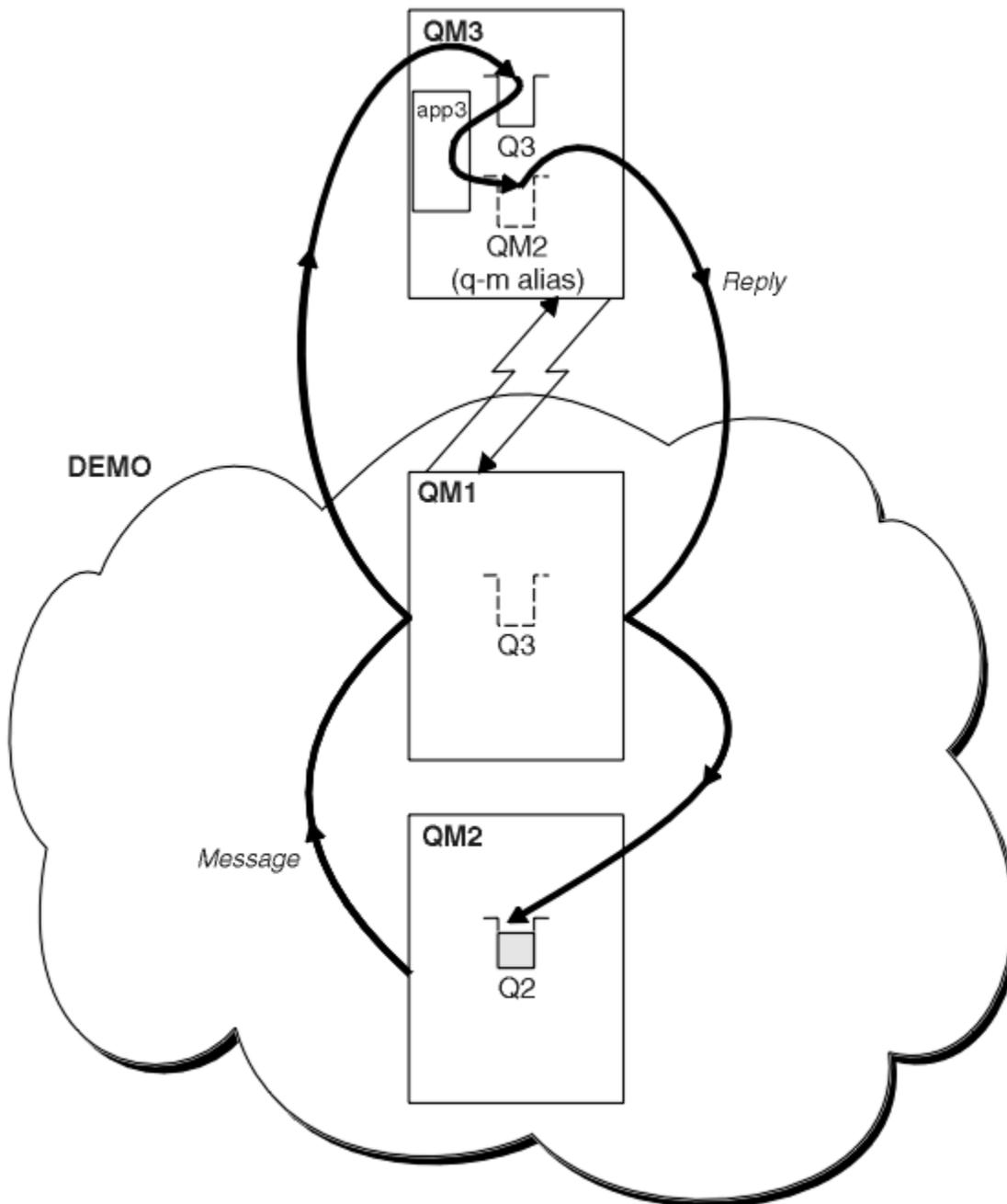


Abbildung 56. In einen WS-Manager außerhalb des Clusters einschalten

### 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 einreicht, 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 56 auf Seite 349](#) nicht angezeigt.

Eine Anwendung in QM2 gibt einen MQPUT -Aufruf 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** in QM3 kann dann Antworten an QM2 senden, indem sie einen MQPUT -Aufruf ausgibt und den Namen der Warteschlange, Q2 und den Namen des Warteschlangenmanagers angibt, QM2.

## 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 Zugriffsteuerungen 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 54 auf Seite 343](#) in „Anforderung/Antwort in einem Cluster konfigurieren“ auf [Seite 342](#) dargestellt.

### **Informationen zu diesem Vorgang**

In diesem Szenario sendet der WS-Manager außerhalb des Clusters QM3 in [Abbildung 57 auf Seite 352](#) Anforderungen 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 57 auf Seite 352](#) nicht explizit angezeigt.

Die Prozedur erweitert das Beispiel in [Abbildung 54 auf Seite 343](#) in „Anforderung/Antwort in einem Cluster konfigurieren“ auf [Seite 342](#).

### **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 einreicht.

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

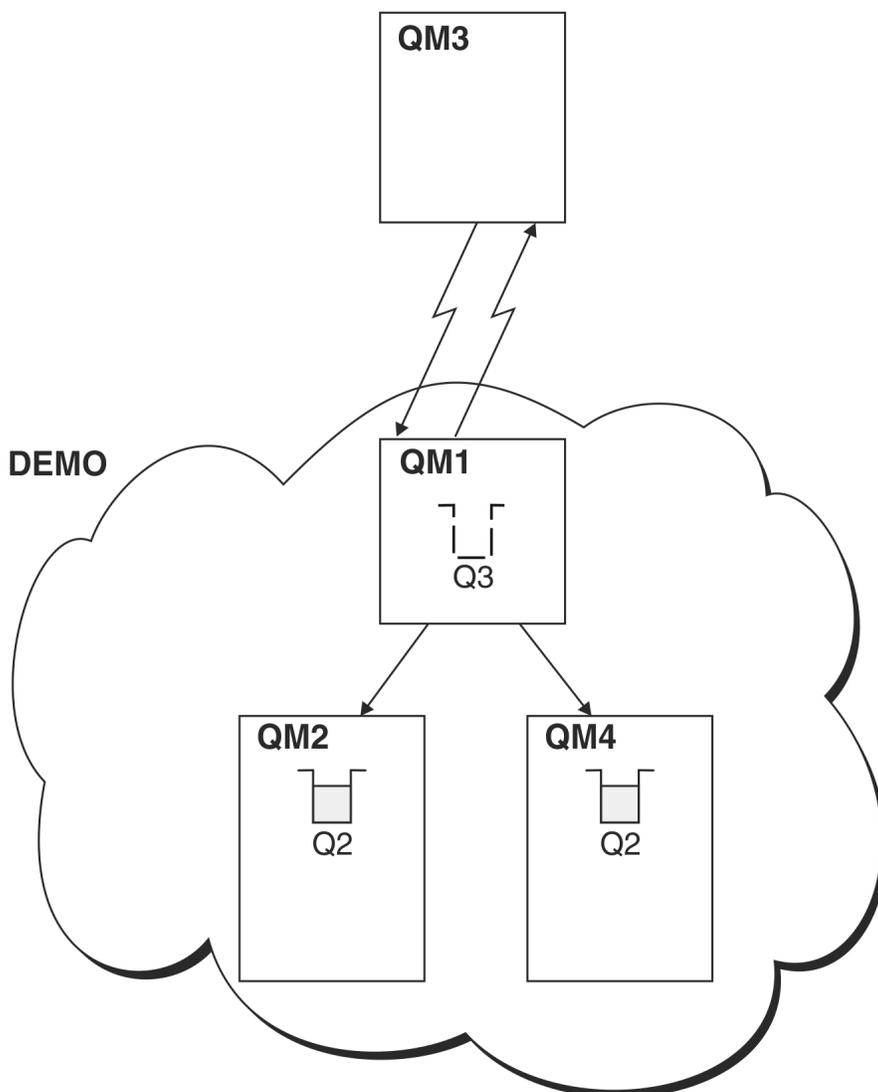


Abbildung 57. Aus einem WS-Manager außerhalb des Clusters einschalten

Wenn eine Anwendung in QM3 einen MQPUT -Aufruf ausgibt, um eine Nachricht an Q2 zu senden, bewirkt die QREMOTE -Definition unter QM3 , dass die Nachricht über den Gateway-Warteschlangenmanager QM1 weitergeleitet 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 Zugriffsteuerungen 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.

#### 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 Informationen**

#### Auflösung des Warteschlangennamens

Namensauflösung

### ***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.

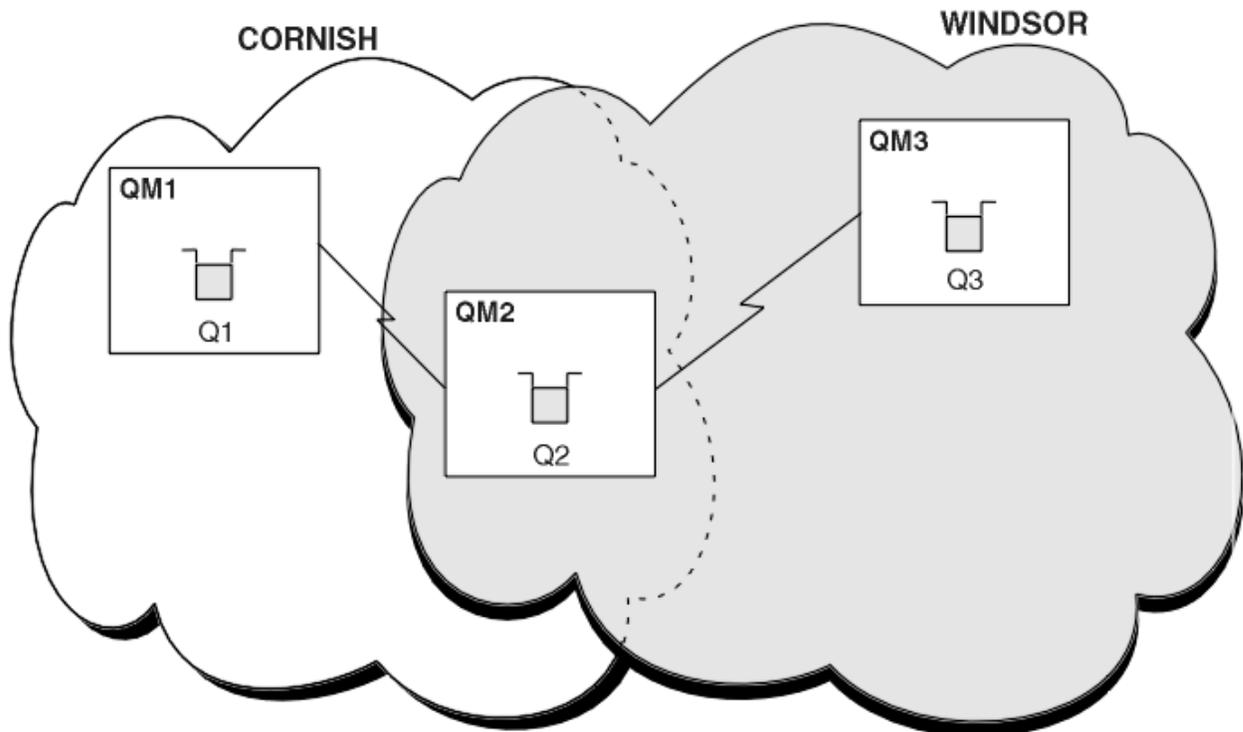


Abbildung 58. Clusterübergreifende Überbrückung

Abbildung 58 auf Seite 354 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 NAMLIST(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 58 auf Seite 354](#) unter QM2 benötigen Sie beispielsweise Folgendes:

```
DEFINE QALIAS(MYQ3) TARGQ(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 Zugriffsteuerungen 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 WS-Manager-Aliasname kann verwendet werden, um den in einem MQOPEN -Aufruf angegebenen WS-Managernamen erneut zu einem anderen Warteschlangenmanager 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 307, von der aus der in Abbildung 1 dargestellte Antwortnachrichtenfluss ausgeführt wird.

Das Diagramm zeigt den Pfad, der von der Antwortnachricht in eine temporäre dynamische Warteschlange zurückgenommen wird, die als RQ bezeichnet wird. Die Serveranwendung, die mit QM3 verbunden ist, öffnet die Antwortwarteschlange unter Verwendung des Warteschlangenmanagernamens QM2. Der Name des Warteschlangenmanagers QM2 ist in QM1 als Aliasname für einen Clusterwarteschlangenmanager definiert. QM3 leitet die Antwortnachricht an QM1 weiter. QM1 leitet die Nachricht an QM2 weiter.

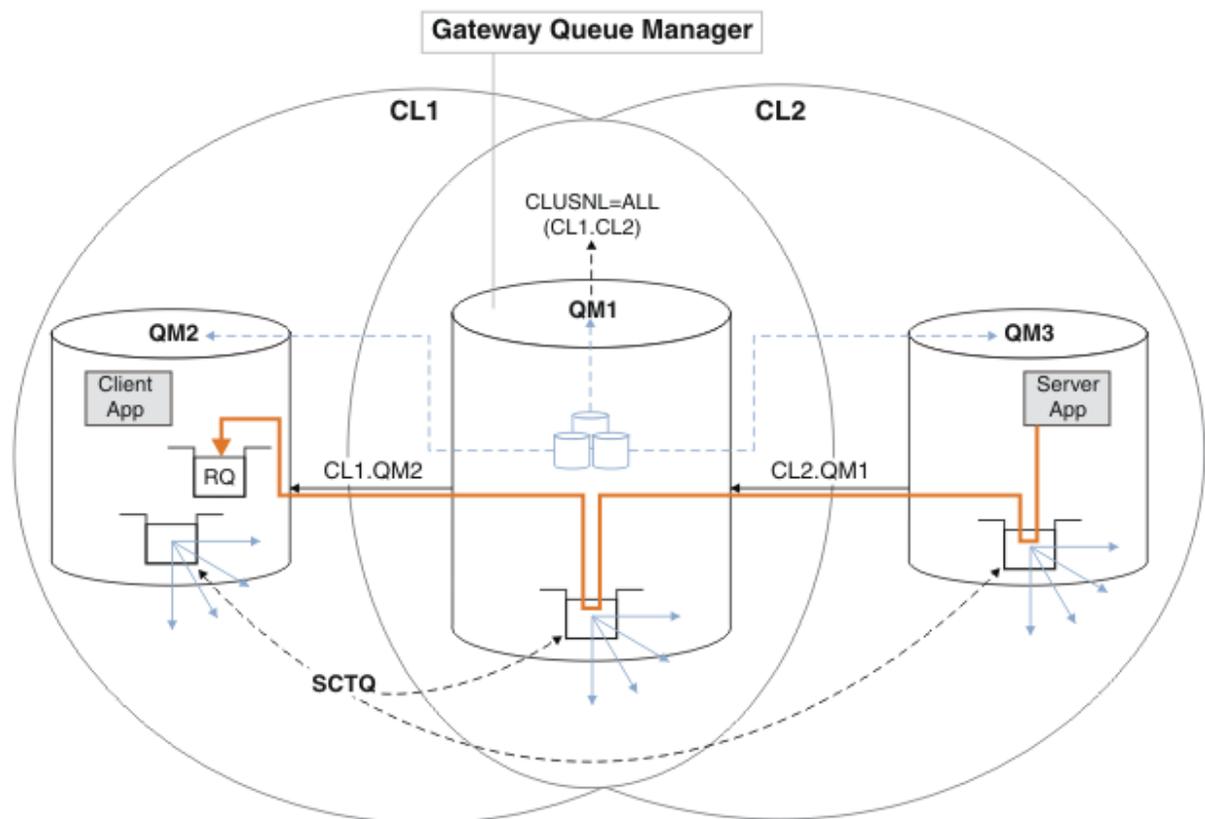


Abbildung 59. Verwenden eines Warteschlangenmanager-Aliasnamens, um die Antwortnachricht an einen anderen Cluster zurückzugeben

Die Art und Weise, wie das Routing funktioniert, ist wie folgt. Jeder Warteschlangenmanager in jedem Cluster verfügt über eine Definition des Warteschlangenmanager-Aliasnamens in 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 in beiden Clustern CL1 und CL2 in Gruppen zusammengefasst und in CL1 in den realen Warteschlangenmanager QM2 aufgelöst. Die Serveranwendung erstellt die Antwortnachricht unter Verwendung der Antwort auf den Warteschlangennamen RQ und beantwortet den Warteschlangenmanagername QM2. Die Nachricht wird an QM1 weitergeleitet, weil die Warteschlangenmanager-Aliasdefinition QM2 in QM1 in Cluster CL2 definiert ist und der Warteschlangenmanager QM2 nicht in Cluster CL2 enthalten 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 auf QM1, damit sie an QM2 übertragen werden können. QM1 leitet die Nachricht an QM2 weiter, da die Warteschlangenmanager-Aliasdefinition in QM1 für 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 sucht die Verbindungsinformationen für QM2 aus dem Repository für CL1 heraus und leitet die Nachricht an QM2 weiter. Damit die Nachricht von QM1 umgeleitet werden kann, muss die Serveranwendung die Antwortwarteschlange geöffnet haben, in der die Option DEFBIND auf MQBND\_BIND\_NOT\_FIXED gesetzt ist. Wenn die Serveranwendung die Antwortwarteschlange mit der Option MQBND\_BIND\_ON\_OPEN geöffnet hatte, wird die Nachricht nicht weitergeleitet und wird in einer Warteschlange für nicht zustellbare Nachrichten nicht mehr angezeigt.

### **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 351.

### **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 Zugriffsteuerungen 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.

Beispiel:

- Eine Anwendung auf WS-Manager VENICE sendet eine Nachricht mit dem Aufruf MQPUT an den Warteschlangenmanager PISA. Die Anwendung stellt die folgenden Antworten auf die Antwortwarteschlange im Nachrichtendeskriptor bereit:

```
ReplyToQ=' QUEUE '  
ReplyToQMgt=' '
```

- 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 Zugriffsteuerungen 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 Zugriffsteuerungen 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) TARGQ(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 mit dem Namen LOKAL aufgelöst 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) TARGQ(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 klonen einander sind. Jeder Warteschlangenmanager ist in der Lage, dieselben Anwendungen auszuführen und lokale Definitionen derselben Warteschlangen zu verwenden.  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](#).

 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

[„Nachrichten an und von Clustern weiterleiten“](#) auf Seite 341

Verwenden Sie Warteschlangenaliasnamen, WS-Manager-Aliasnamen und Definitionen ferner Warteschlangen, um Cluster mit externen Warteschlangenmanagern und anderen Clustern zu verbinden.

### Zugehörige Tasks

[„WS-Manager-Cluster konfigurieren“](#) auf Seite 255

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 268

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 Übertragungswarte-

schlange 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 351

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 Informationen**

[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.](#)

[Das Beispielprogramm 'Clusterwarteschlangenüberwachung' \(AMQSCLM\)](#)

[Exits für Clusterauslastung schreiben und kompilieren](#)

### **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 60 auf Seite 363 zeigt einen Cluster, in dem mehr als eine Definition für die Warteschlange Q3 vorhanden ist. Wenn eine Anwendung von QM1 eine Nachricht in Q3 einreicht, 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.

Mit der Option MQOPEN steuert DefBind, 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.

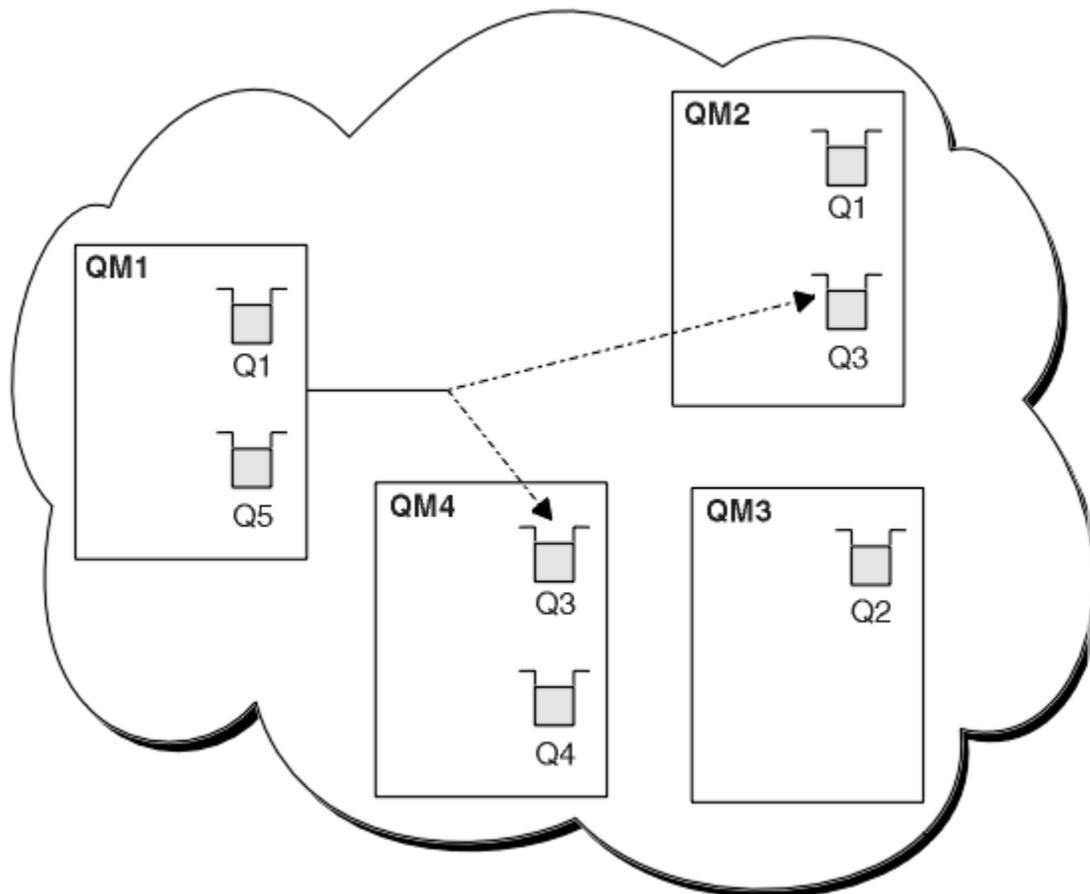


Abbildung 60. Ein Cluster mit mehreren Instanzen derselben Warteschlange

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 wieder in einen anderen Kanal umverteilt. Beachten Sie auch, dass die Nachrichtenzuordnung nur dann stattfindet, wenn ein Clusterkanal tatsächlich fehlschlägt. Die Neuordnung 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 für z/OS wird der Kanal erst vollständig gestoppt, wenn der Nachrichten-neuzuordnungsprozess 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 aus 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 verwendeter 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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie im Abschnitt Neuen WS-Manager zu einem Cluster hinzufügen beschrieben konfiguriert. Es enthält drei Warteschlangenmanager; LONDON und NEWYORK enthalten beide vollständige Repositories, 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)
```

- 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.

- Installieren Sie die Inventaranwendung auf dem System in Paris.
- 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 61 auf Seite 365](#) wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

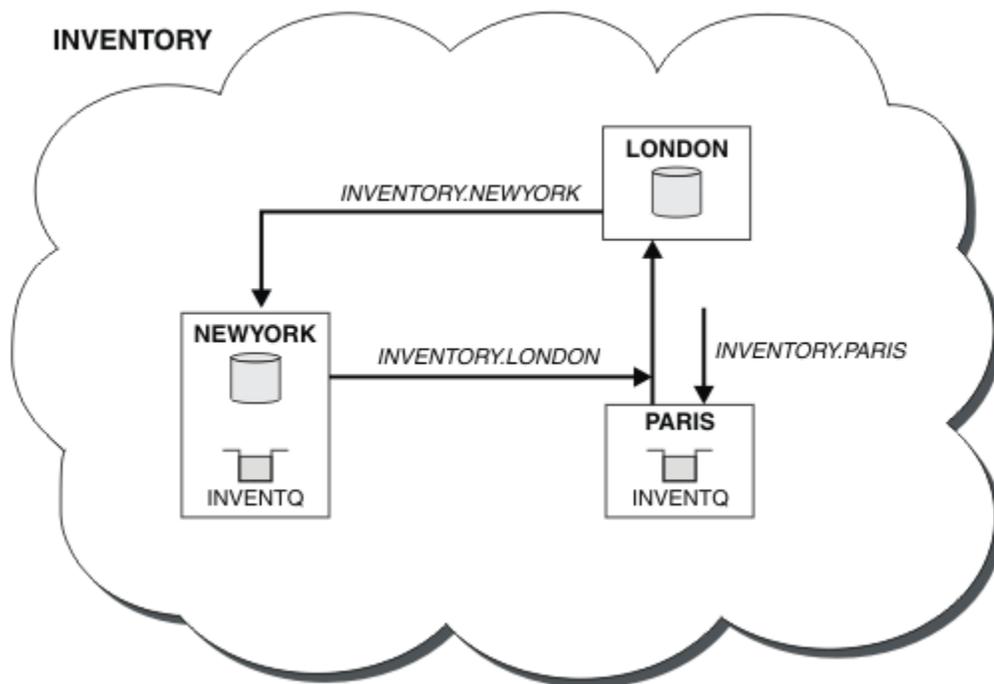


Abbildung 61. Der INVENTORY -Cluster mit drei Warteschlangenmanagern

Die Änderungen an diesem Cluster wurden ohne Änderung der Warteschlangenmanager NEWYORK oder LONDON vorgenommen. Die vollständigen Repositories 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 verwendeter 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 Repositories 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 Repositories, 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 Repositories 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 Repositories.

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('Clusterempfängerkanal bei Netz B für TOKYO')
```

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Clusterempfängerkanal bei Netz A für 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('Clustersenderkanal von TOKYO an Repository auf 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 Empfangsprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In [Abbildung 62 auf Seite 368](#) wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

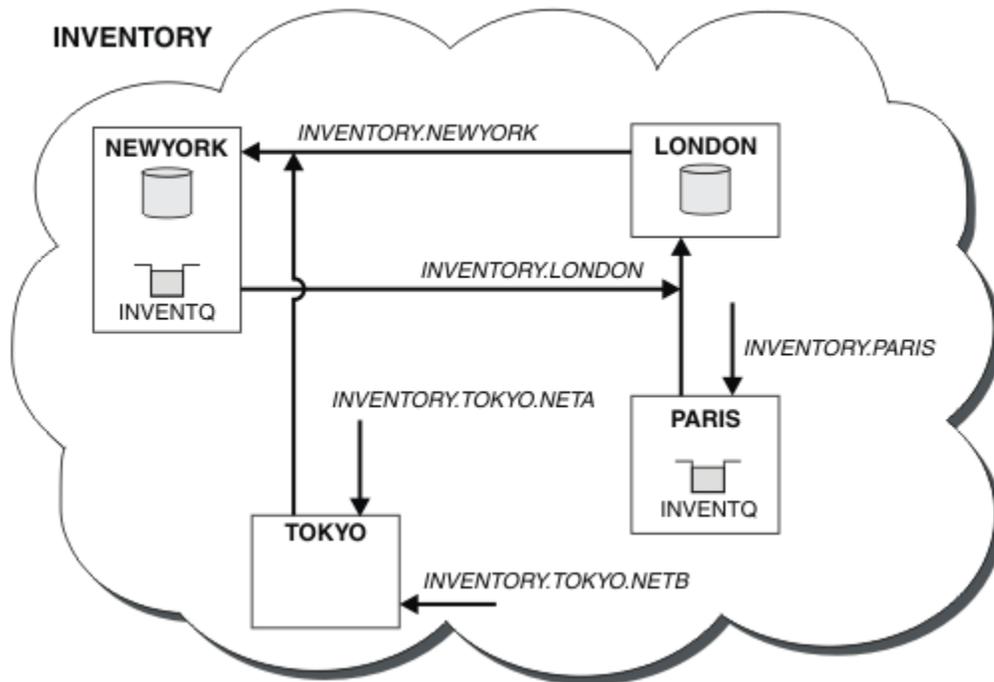


Abbildung 62. Der INVENTORY -Cluster mit vier Warteschlangenmanagern

Wenn Sie nur drei Definitionen vornehmen, haben wir den WS-Manager TOKYO 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 verwendeter 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 280

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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „Zwei Netze in einem Cluster verwenden“ auf Seite 366 beschrieben konfiguriert. Sie enthält vier WS-Manager; LONDON und NEWYORK enthalten vollständige Repositories; PARIS und TOKYO enthalten Teilrepositories. 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 verwendeter 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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen“ auf Seite 280 beschrieben konfiguriert. Es enthält drei Warteschlangenmanager; LONDON und NEWYORK enthalten beide vollständige Repositories, 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 Repositories 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 Repositories.

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) CONNAME(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(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender channel from CHICAGO 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 Empfangsprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In [Abbildung 63 auf Seite 372](#) wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

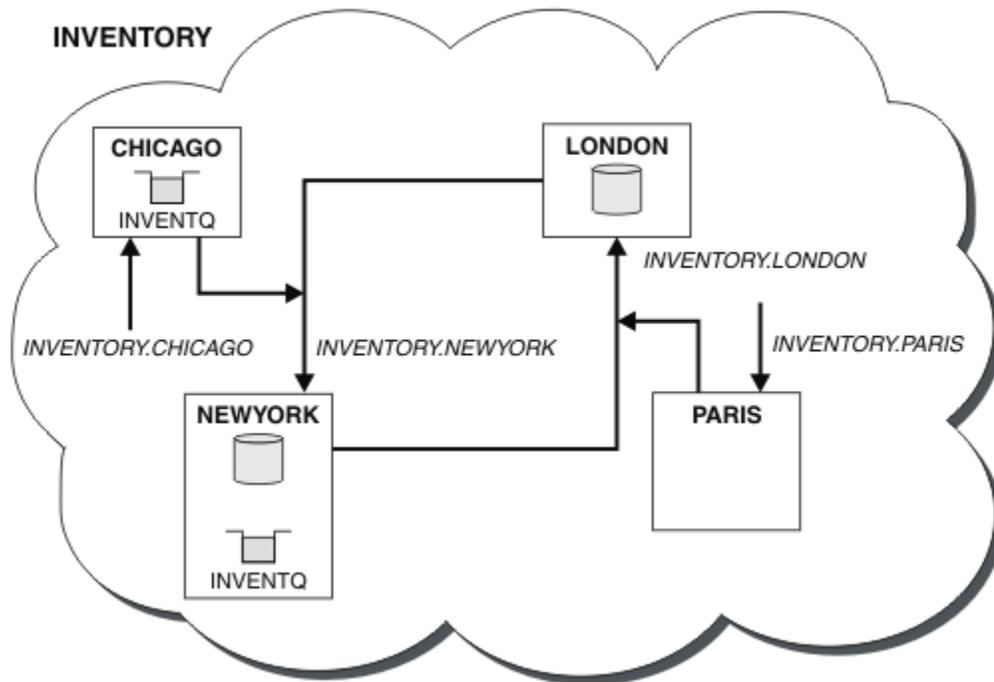


Abbildung 63. Cluster BESTANDSFÜHRUNG mit vier Warteschlangenmanagern

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 ihr Status mehrmals, wobei einige Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig 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 verwendeter 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 verwendeter 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 Repositories 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 Repositories 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 Repositories aus.

Wählen Sie zwei Warteschlangenmanager aus, die die vollständigen Repositories 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) CONNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Clusterempfängerkanal')
```

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-Repositories zu verbinden.

```
DEFINE CHANNEL (PRICECHECK.REPOS1) CHLTYPE (CLUSSDR) TRPTYPE (TCP) CONNAME (REPOS1.COM) CLUSTER (PRICECHECK) DESCR ('Cluster senderkanal an Repository-Warteschlangenmanager')
```

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 Empfangsprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Clusterempfängerkanal, wenn er benötigt wird.

## Ergebnisse

In [Abbildung 64 auf Seite 374](#) wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

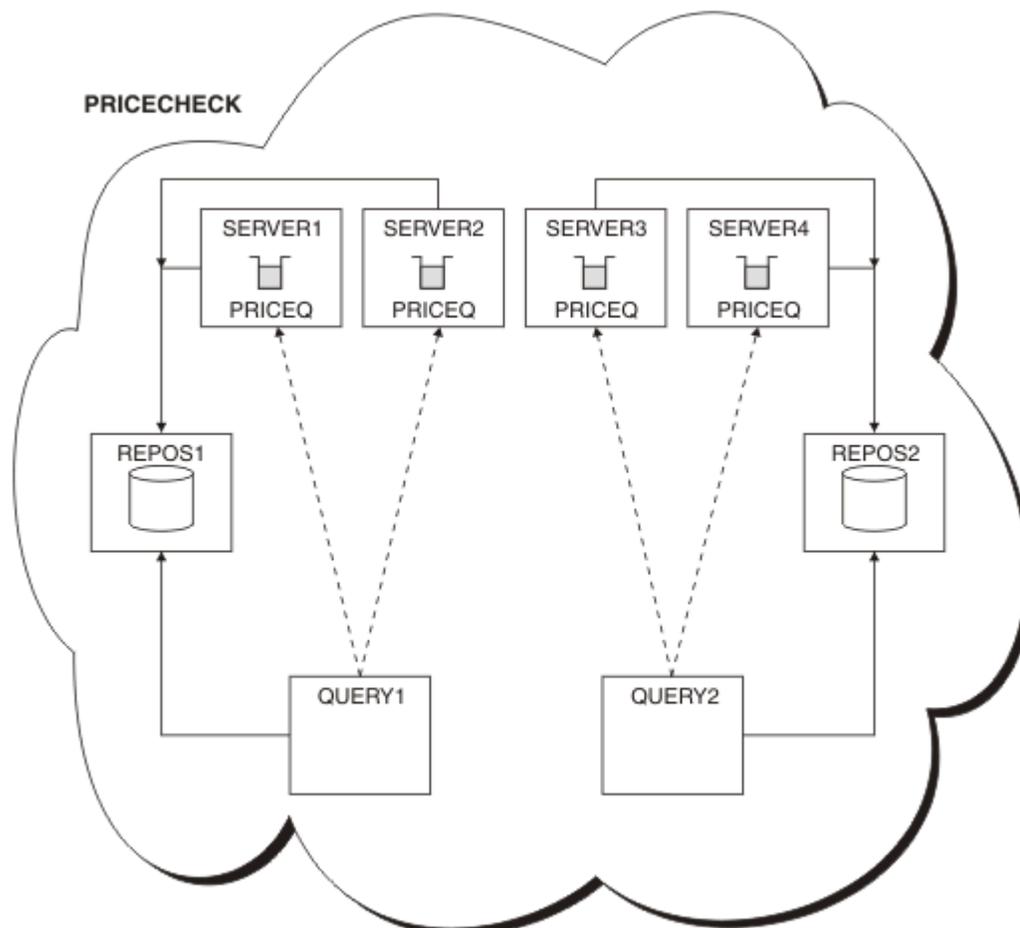


Abbildung 64. Der PRICECHECK -Cluster mit vier Server-WS-Managern, zwei Repositories und zwei Abfragewarteschlangenmanagern

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 Repositories verfügbar sind, bevor Sie diese Task starten.

Szenario:

- Der INVENTORY -Cluster wurde wie in „WS-Manager zu einem Cluster hinzufügen“ auf Seite 280 beschrieben konfiguriert. Es enthält drei WS-Manager: LONDON und NEWYORK enthalten vollständige Repositories, 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 Repositories 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 Repositories.

```
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 375 wird der Cluster angezeigt, der von dieser Task eingerichtet wurde.

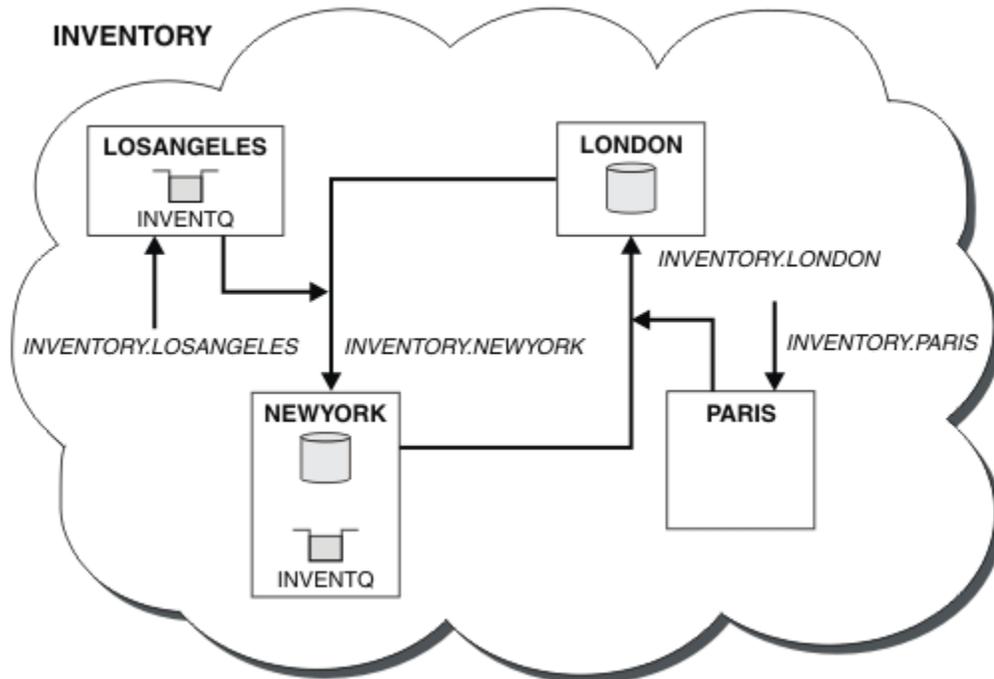


Abbildung 65. Der INVENTORY -Cluster mit vier Warteschlangenmanagern

Diese Änderung am Cluster wurde ausgeführt, ohne dass die Warteschlangenmanager LONDON und PARIS geändert werden müssen. Die Repositories 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 verwendeter 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 zu stellen. Anwendungen können eine einzelne Nachricht in eine Warteschlange stellen, die noch nicht geöffnet ist, indem Sie den Aufruf MQPUT1 verwenden.

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 Datenbankanwendung, 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 verwendeter 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.

### **Benennen Sie ein bestimmtes Ziel im Aufruf MQOPEN .**

Geben Sie den Namen der fernen Warteschlange und den Namen des Warteschlangenmanagers in jedem MQOPEN -Aufruf an, und alle Nachrichten, die über diese Objektkennung in die Warteschlange gestellt werden, wechseln zu demselben Warteschlangenmanager, der möglicherweise der lokale WS-Manager ist.

Die Angabe des Namens der fernen Warteschlange und des Warteschlangenmanagers 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.**

Erzwingt, dass alle Nachrichten, die mit der Option MQOO\_BIND\_ON\_OPEN im Aufruf MQOPEN an dieselbe Zieladresse gestellt werden sollen, in dieselbe Zieladresse versetzt werden. 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 `MQ00_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.

`MQ00_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 `MQ00_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 `MQ00_BIND_ON_OPEN` im Aufruf `MQOPEN` besteht darin, Ihre Warteschlangendefinitionen zu ändern. Geben Sie in Ihren Warteschlangendefinitionen `DEFBIND(OPEN)` an und lassen Sie die Option `DefBind` im Aufruf `MQOPEN` die Standardeinstellung `MQ00_BIND_AS_Q_DEF` zu.

### **Legen Sie die Option `MQ00_BIND_ON_GROUP` im Aufruf `MQOPEN` fest.**

Erzwingt alle Nachrichten in einer Gruppe, die mit der Option `MQ00_BIND_ON_GROUP` im Aufruf `MQOPEN` an dieselbe Zieladresse gestellt werden sollen. Bei Verwendung von Nachrichtengruppen mit Clustern muss entweder `MQ00_BIND_ON_OPEN` oder `MQ00_BIND_ON_GROUP` angegeben werden, um sicherzustellen, dass alle Nachrichten in der Gruppe an demselben Ziel verarbeitet werden.

Wenn Sie eine Warteschlange öffnen und `MQ00_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. `MQ00_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 `MQ00_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.

`MQ00_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 `MQ00_BIND_ON_GROUP` angeben, wird eine Route ausgewählt, wenn die Warteschlange geöffnet wird.

Damit `MQ00_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 `MQ00_BIND_ON_GROUP` angegeben wird, die Nachrichten jedoch nicht gruppiert sind, entspricht das Verhalten dem Wert `MQ00_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 MQ00\_BIND\_ON\_GROUP im Aufruf MQOPEN besteht darin, Ihre Warteschlangendefinitionen zu ändern. Geben Sie in Ihren Warteschlangendefinitionen DEFBIND(GROUP) an und lassen Sie die Option DefBind im Aufruf MQOPEN die Standardeinstellung MQ00\_BIND\_AS\_Q\_DEF zu.

## 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.

## Publish/Subscribe-Messaging konfigurieren

Sie können den Status des eingereichten 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 eingereichtem Publish/Subscribe finden Sie in den folgenden Unterabschnitten:
  - „Publish/Subscribe-Nachrichtenattribute in der Warteschlange festlegen“ auf Seite 382
  - „In Warteschlange eingereichtes Publish/Subscribe starten“ auf Seite 384
  - „In Warteschlange eingereichtes Publish/Subscribe stoppen“ auf Seite 384
  - „Datenstrom hinzufügen“ auf Seite 385
  - „Datenstrom löschen“ auf Seite 386
  - „Einen Subskriptionspunkt hinzufügen“ auf Seite 386
  - „Kombinieren von Topic-Bereichen in Publish/Subscribe-Netzen“ auf Seite 395

## 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](#) .

<i>Tabelle 25. Publish/Subscribe-Konfigurationsparameter</i>	
Beschreibung	MQSC-Parametername
Wiederholungszähler für Befehlsnachricht	<b>PSRTCNT</b>
Unzustellbare Befehlseingabenachricht verwerfen	<b>PSNPMSG</b>
Verhalten nach unzustellbarer Befehlsantwortnachricht	<b>PSNPRES</b>
Verarbeiten von Befehlsnachrichten unter Synchronisationspunkt	<b>PSSYNCPT</b>

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 **PSYNCP** auf `Yes` gesetzt ist.

Dies kann sich auch auf die Leistung der Verarbeitung nicht persistenter Veröffentlichungen auswirken, da die **MQGET** -Verarbeitung aus der Warteschlange `STREAM` nun unter Synchronisationspunkt erfolgt.

- `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 eingereichtes 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
- ENABLED

### 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 eingereichte Publish/Subscribe-Befehle und -MQI-Aufrufe (Publish/Subscribe Message Queue Interface).

## In Warteschlange eingereichtes Publish/Subscribe stoppen

Sie können die eingereichte 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 WebSphere MQ 6 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 dem gleichen Namen wie der Datenstrom der IBM WebSphere MQ 6.
2. Definieren Sie ein lokales Thema mit dem gleichen Namen wie der Datenstrom der IBM WebSphere MQ 6.
3. Fügen Sie den Namen der Warteschlange der Namensliste SYSTEM.QPUBSUB.QUEUE.NAMELIST hinzu.
4. Wiederholen Sie diese Schritte für alle Warteschlangenmanager der IBM WebSphere MQ 7.1 oder höher, die sich in der Publish/Subscribe-Hierarchie befinden.

### 'Sport' hinzufügen

Im Beispiel für die gemeinsame Nutzung des Datenstroms 'Sport' arbeiten IBM WebSphere MQ 6 - und 7.1 -Warteschlangenmanager in derselben Publish/Subscribe-Hierarchie. Die Warteschlangenmanager der IBM WebSphere MQ 6 verwenden einen Datenstrom mit der Bezeichnung 'Sport'. Das Beispiel zeigt, wie eine Warteschlange und ein Thema auf IBM WebSphere MQ 7.1-Warteschlangenmanagern mit dem Namen 'Sport' mit einer Themenzeichenfolge 'Sport' erstellt werden, die gemeinsam mit dem IBM WebSphere MQ 6-Datenstrom 'Sport' verwendet wird.

Eine IBM WebSphere MQ 7.1 Veröffentlichungsanwendung, Veröffentlichung im Thema 'Sport' mit der Themenzeichenfolge 'Soccer/Results', erstellt die resultierende Themenzeichenfolge 'Sport/Soccer/Results'. Auf IBM WebSphere MQ 7.1 -Warteschlangenmanagern empfangen Subskribenten des Themas 'Sport' mit der Themenzeichenfolge 'Soccer/Results' die Veröffentlichung.

Auf IBM WebSphere MQ 6 -Warteschlangenmanagern empfangen Subskribenten des Datenstroms 'Sport' mit der Themenzeichenfolge 'Soccer/Results' die Veröffentlichung.

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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.BROKER.ADMIN.STREAM')
  3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM', 'SYSTEM.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.

Wenn ein Broker Version 6 hat, verwalten Sie ihn als IBM WebSphere MQ 6 -Broker. Dies bedeutet, dass Sie die Möglichkeit haben, die Datenstromwarteschlange manuell zu erstellen oder den Broker dynamisch zu erstellen, wenn er benötigt wird. Die Warteschlange basiert auf der Modellwarteschlangendefinition `SYSTEM.BROKER.MODEL.STREAM`.

Wenn ein Broker die Version 7.1 aufweist, müssen Sie jeden IBM WebSphere MQ 7.1 -Warteschlangenmanager in der Hierarchie manuell konfigurieren.

## Datenstrom löschen

Sie können einen Datenstrom aus einem Warteschlangenmanager in IBM WebSphere MQ 7.1 oder höher 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 für alle weiteren angeschlossenen Warteschlangenmanager in IBM WebSphere MQ 7.1 oder höher, 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

Anwendungen von IBM WebSphere MQ 7.1 oder höher verwenden keine Subskriptionspunkte, können jedoch mit vorhandenen Anwendungen, die mit dem Migrationsmechanismus für Subskriptionspunkte arbeiten, eingesetzt werden.

**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.

Subskriptionspunkte funktionieren nicht mit eingereichten Publish/Subscribe-Programmen, die MQRFH1-Header verwenden, die von IBM WebSphere MQ 6 oder früher migriert wurden.

Es ist nicht erforderlich, Subskriptionspunkte hinzuzufügen, damit integrierte Publish/Subscribe-Anwendungen verwendet werden können, die für IBM WebSphere MQ 7.1 oder höher geschrieben 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 vorhandene 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, USD und GBP, hinzufügt.

```
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 Informationen

[Verteilte Publish/Subscribe-Netzwerke](#)

[Veröffentlichungsumfang](#)

[Subskriptionsumfang](#)

[Themenbereiche](#)

[Clusterthemen definieren](#)

## Publish/Subscribe-Cluster konfigurieren

Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLROUTE** fest.

### 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 Repositories 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 280 beschrieben konfiguriert. Es enthält drei Warteschlangenmanager; LONDON und NEWYORK enthalten beide vollständige Repositories, 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 Cluster-Topics auf einem WS-Manager im Cluster stellen Sie das Topic 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 gewährt. Bei der Durchführung von Publikations- und Subskriptionsoperationen kann sich jeder Warteschlangenmanager direkt mit jedem anderen Warteschlangenmanager im Cluster verbinden. Siehe [Direct routed Publish/Subscribe-Cluster](#).

Ab IBM MQ 8.0 können 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). Bei der Durchführung von Veröffentlichungs- und Subskriptionsoperationen stellen Warteschlangenmanager im Cluster nur mit den TOPICHOST-Warteschlangenmanagern eine Verbindung her, nicht direkt miteinander. Die Topic-Host-Warteschlangenmanager sind für das Routing von Publikationen aus Warteschlangenmanagern verantwortlich, in denen Publikationen für Warteschlangenmanager mit übereinstimmenden Subskriptionen veröffentlicht werden. Siehe [Topic host routed Publish/Subscribe clusters](#).

**Anmerkung:** Nachdem ein Themenobjekt in Gruppen zusammengefasst wurde (über die Eigenschaft **CLUSTER**), können Sie den Wert der Eigenschaft **CLROUTE** nicht mehr ändern. Sie müssen erst die Konfiguration des Objekts als Clusterthema rückgängig machen (indem **CLUSTER** auf ' ' gesetzt wird), damit dieser Wert geändert werden kann. Durch Aufhebung der Clusterzuordnungen eines Themas wird die Themendefinition zu einem lokalen Thema, wodurch ein Zeitraum entsteht, in dem keine Veröffentlichungen an ferne Warteschlangenmanager geliefert werden; dies ist zu berücksichtigen, wenn diese Änderung vorgenommen wird. 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. Wenn Sie beispielsweise ein Cluster mit direktem Routing-Cluster auf dem LONDON -Warteschlangenmanager erstellen möchten, müssen Sie das Thema wie folgt erstellen:

```
DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
```

Durch die Definition eines Cluster-Topics auf einem WS-Manager im Cluster stellen Sie das Topic für den gesamten Cluster zur Verfügung.

Weitere Informationen zur Verwendung von **CLROUTE** finden Sie im Abschnitt [DEFINE TOPIC \(CLROUTE\)](#) 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. Weitere Informationen finden Sie in [„Weitere Topic-Hosts zu einem Topic-Host-Routing-Cluster hinzufügen“](#) auf Seite 392.

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. Weitere Informationen finden Sie in [„Kombinieren der Topic-Bereiche mehrerer Cluster“](#) auf Seite 399. Sie können auch steuern, welche Veröffentlichungen von einem Cluster in einen anderen fließen. Weitere Informationen finden Sie in [„Topic-Bereiche in mehreren Clustern kombinieren und isolieren“](#) auf Seite 401.

## 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.

## Zugehörige Informationen

[Publish/Subscribe-Cluster entwerfen](#)

[Verteilte Publish/Subscribe-Fehlerbehebung](#)

[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 wei-

tergeleiteten 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.

Für das Thema Host-Routing definieren Sie zunächst den neuen Themenhost 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 prüfen Sie, dass eine Definition 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 sind. Eine Möglichkeit, die Proxy-Subskriptionen auf einem WS-Manager anzuzeigen, besteht darin, den folgenden Befehl **runmqsc** auszugeben:

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

3. Für Topic-Host-Routing definieren Sie den Themenhost auf QM2 als PUB(ENABLED) erneut und definieren Sie dann den Themenhost auf QM1 erneut als PUB(DISABLED).

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. Mit **RESET CLUSTER** werden automatisch alle Themenobjekte gelöscht, 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, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLROUTE** fest.

#### 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 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.

## 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 388.

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 zum Anzeigen der Proxy-Subskriptionen 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, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLROUTE** fest.

**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.

### Zugehörige Tasks

Publish/Subscribe-Cluster konfigurieren

Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLROUTE** fest.

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.

### Zugehörige Informationen

Verteilte Publish/Subscribe-Netzwerke

Veröffentlichungsumfang

Subskriptionsumfang

## 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.

### **Zugehörige Tasks**

Publish/Subscribe-Cluster konfigurieren

Definieren Sie ein Thema in einem Warteschlangenmanager. Um das Thema zu einem Clusterthema zu machen, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLROUTE** fest.

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.

### **Zugehörige Informationen**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

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 Tasks

Warteschlangenmanager der Version 7 oder höher zu vorhandenen Topic-Bereichen der IBM WebSphere MQ 6 hinzufügen

Erweitern Sie ein vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6, um mit einem Warteschlangenmanager der Version 7 oder höher zu interagieren und dabei dieselben Topic-Bereiche gemeinsam zu nutzen.

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 subscribieren 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 subscribieren

Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subscribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

### Zugehörige Informationen

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

Clusterthemen definieren

## ***Warteschlangenmanager der Version 7 oder höher zu vorhandenen Topic-Bereichen der IBM WebSphere MQ 6 hinzufügen***

Erweitern Sie ein vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6, um mit einem Warteschlangenmanager der Version 7 oder höher zu interagieren und dabei dieselben Topic-Bereiche gemeinsam zu nutzen.

## Vorbereitende Schritte

Sie verfügen über ein vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6.

Sie haben IBM WebSphere MQ 7 oder höher auf einem neuen Server installiert und einen Warteschlangenmanager konfiguriert.

## Informationen zu diesem Vorgang

Sie möchten Ihr vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6 erweitern, um mit Warteschlangenmanagern der Version 7 oder höher zu arbeiten.

Sie haben beschlossen, die Entwicklung des Publish/Subscribe-Systems von IBM WebSphere MQ 6 zu stabilisieren, das die Publish/Subscribe-Schnittstelle in der Warteschlange verwendet. Sie beabsichtigen, mit IBM WebSphere MQ 7 oder einer späteren MQI-Version Erweiterungen zum System hinzuzufügen. Sie haben jetzt keine Pläne zum Umschreiben der in die Warteschlange gestellten Publish/Subscribe-Anwendungen.

Sie beabsichtigen, in der Zukunft ein Upgrade der Warteschlangenmanager von IBM WebSphere MQ 6 auf IBM WebSphere MQ 7 oder höher vorzunehmen. Im Moment führen Sie die vorhandenen Publish/Subscribe-Anwendungen in der Warteschlange auf den Warteschlangenmanagern von IBM WebSphere MQ 7 oder höher aus.

## Vorgehensweise

1. Erstellen Sie eine Gruppe von Sender-Empfänger-Kanälen, um den Warteschlangenmanager von IBM WebSphere MQ 7 oder höher in beiden Richtungen mit einem der Warteschlangenmanager von IBM WebSphere MQ 6 zu verbinden.
2. Erstellen Sie zwei 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. Wenn das Publish/Subscribe-System von IBM WebSphere MQ 6 Datenströme verwendet, fügen Sie die Datenströme dem Warteschlangenmanager von IBM WebSphere MQ 7 oder höher hinzu (siehe [Datenstrom hinzufügen](#)).
5. Überprüfen Sie, ob der IBM WebSphere MQ 7 oder der spätere Warteschlangenmanager **PSMODE** auf **ENABLE** gesetzt ist.
6. Ändern Sie das Attribut **PARENT** so, dass es auf einen der Warteschlangenmanager von IBM WebSphere MQ 6 verweist.
7. Überprüfen Sie den Status der Elternbeziehung zwischen den Warteschlangenmanagern in beide Richtungen.

## Nächste Schritte

Nachdem Sie die Task abgeschlossen haben, nutzen sowohl der Warteschlangenmanager für IBM WebSphere MQ 6 als auch der Warteschlangenmanager für Version 7 oder höher dieselben Topicbereiche. Sie können zum Beispiel alle folgenden Tasks ausführen.

- Tauschen Sie Veröffentlichungen und Subskriptionen zwischen Warteschlangenmanagern unter IBM WebSphere MQ 6 und 7 oder höher aus.
- Ausführung Ihrer vorhandenen Publish/Subscribe-Programme der IBM WebSphere MQ 6 auf dem Warteschlangenmanager von IBM WebSphere MQ 7 oder höher
- Sie können den Topic-Bereich auf dem Warteschlangenmanager IBM WebSphere MQ 6 oder 7 oder höher anzeigen und ändern.
- Schreiben von Publish/Subscribe-Anwendungen von IBM WebSphere MQ 7 oder höher und deren Ausführung auf dem Warteschlangenmanager von IBM WebSphere MQ 7 oder höher
- Erstellen neuer Veröffentlichungen und Subskriptionen mit den Anwendungen von IBM WebSphere MQ 7 oder höher und deren Austausch mit Anwendungen der IBM WebSphere MQ 6

## 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 abonnieren 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 abonnieren

Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und abonnieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

## **Zugehörige Informationen**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

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 abonnieren Sie es in einem anderen Cluster.

## **Vorbereitende Schritte**

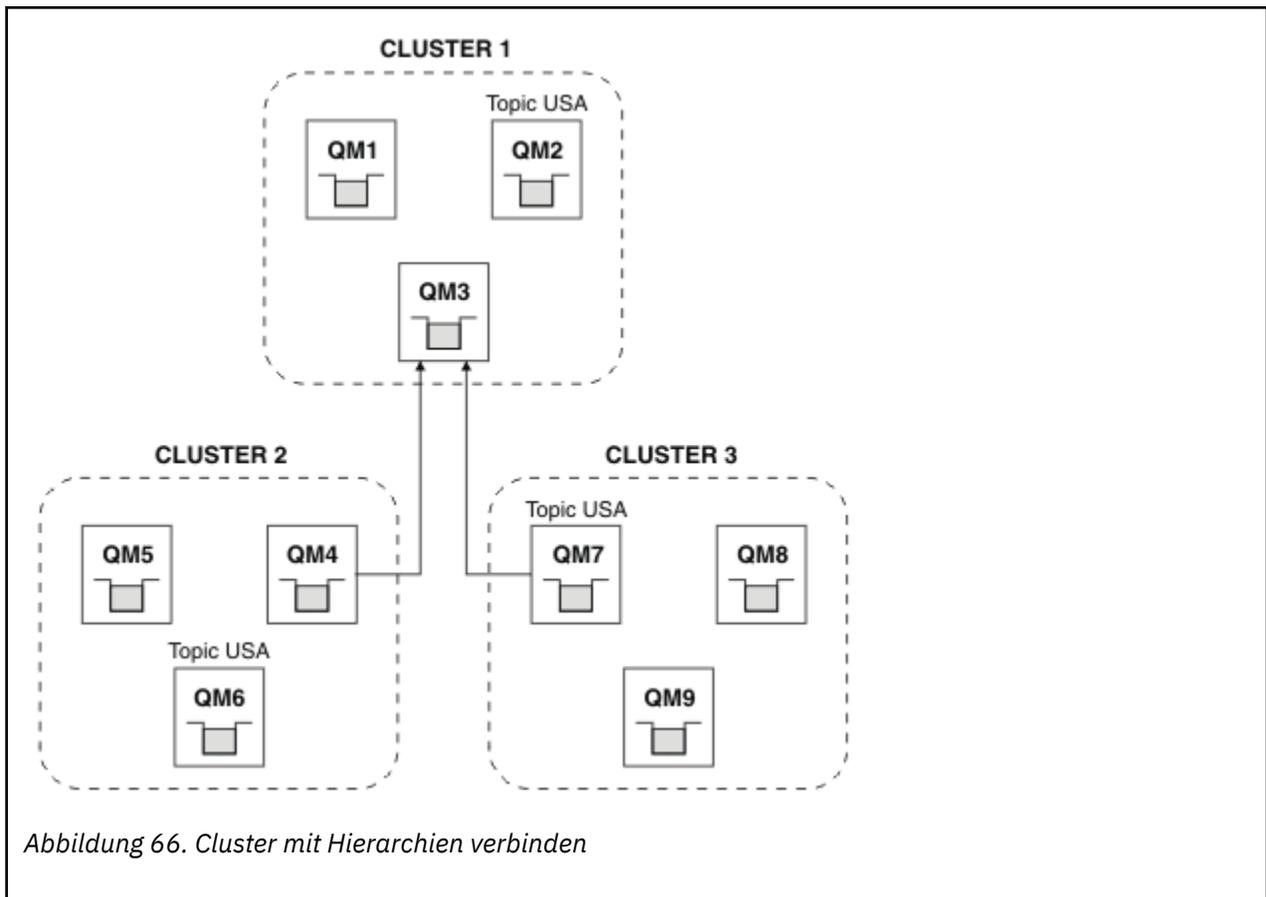
Bei dieser Task wird davon ausgegangen, dass Sie vorhandene Publish/Subscribe-Cluster direkt weitergeleitet haben und einige Clusterthemen in allen Clustern weitergeben möchten.

**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 66 auf Seite 400). 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.



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 66 auf Seite 400](#) veröffentlichen 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 401.

### **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.](#)

[Warteschlangenmanager der Version 7 oder höher zu vorhandenen Topic-Bereichen der IBM WebSphere MQ 6 hinzufügen](#)

[Erweitern Sie ein vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6, um mit einem Warteschlangenmanager der Version 7 oder höher zu interagieren und dabei dieselben Topic-Bereiche gemeinsam zu nutzen.](#)

[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 subscribieren](#)

[Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subscribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.](#)

### **Zugehörige Informationen**

[Verteilte Publish/Subscribe-Netzwerke](#)

[Themenbereiche](#)

[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 399. 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 66 auf Seite 400 im Abschnitt „Kombinieren der Topic-Bereiche mehrerer Cluster“ auf Seite 399 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 \(siehe Abbildung 67 auf Seite 402\). Verwenden Sie den Brückenwarteschlangenmanager, um zu filtern, welche Veröffentlichungen und Subskriptionen von einem Cluster in einen anderen fließen können.](#)

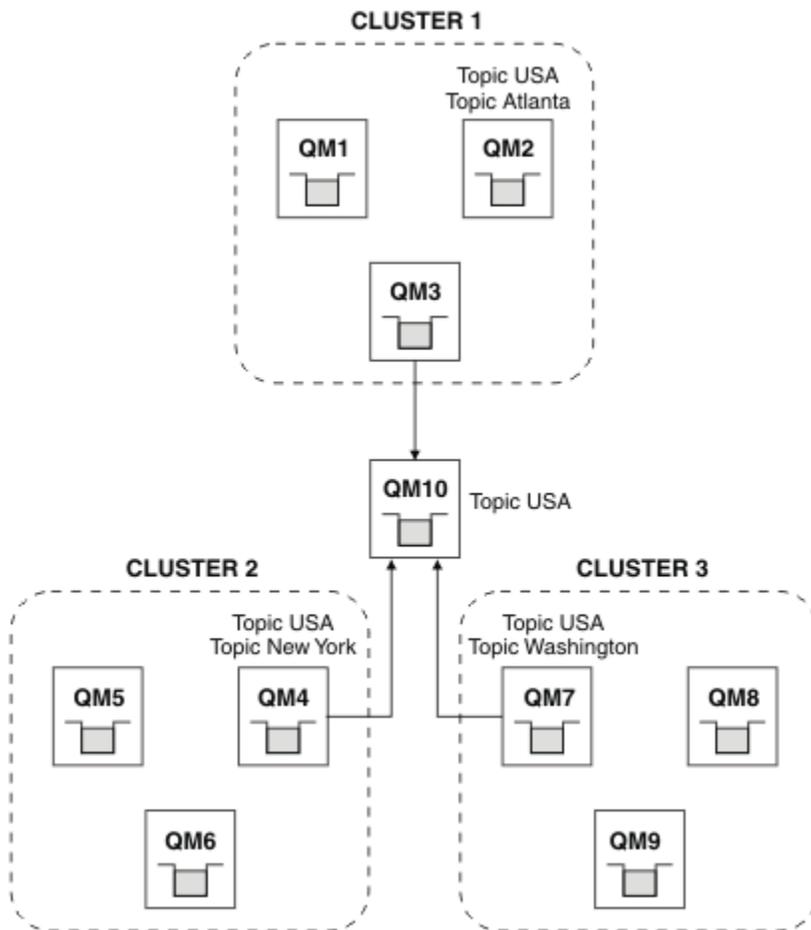


Abbildung 67. Überbrückte Cluster

Verwenden Sie die Brücke, um Clusterthemen zu isolieren, die nicht über die Brücke auf den anderen Clustern zugänglich gemacht werden sollen. In [Abbildung 67](#) auf Seite 402 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 67 auf Seite 402](#) 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 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.

[Warteschlangenmanager der Version 7 oder höher zu vorhandenen Topic-Bereichen der IBM WebSphere MQ 6 hinzufügen](#)

Erweitern Sie ein vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6, um mit einem Warteschlangenmanager der Version 7 oder höher zu interagieren und dabei dieselben Topic-Bereiche gemeinsam zu nutzen.

[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 subscribieren Sie es in einem anderen Cluster.

[Themenbereiche in mehreren Clustern veröffentlichen und subscribieren](#)

Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subscribieren. Sie können diese Technik verwenden, solange sich die Topic-Bereiche in den Clustern nicht überschneiden.

### Zugehörige Informationen

[Verteilte Publish/Subscribe-Netzwerke](#)

[Themenbereiche](#)

[Clusterthemen definieren](#)

[Veröffentlichungsumfang](#)

[Subskriptionsumfang](#)

### ***Themenbereiche in mehreren Clustern veröffentlichen und subscribieren***

Sie können Themen in mehreren Clustern mit überlappenden Clustern veröffentlichen und subscribieren. 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 68](#) auf Seite 404. 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 subscribieren.

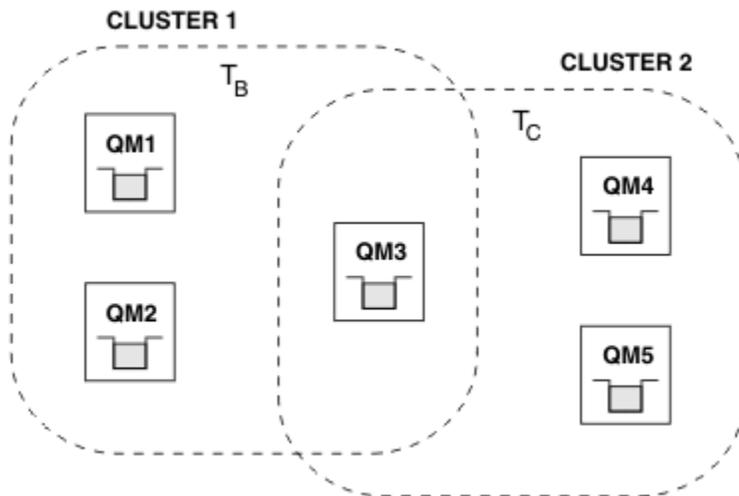


Abbildung 68. Überschneidung von Clustern, nicht überlappenden Themenbereichen

## 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 68](#) auf Seite 404 können Publisher und Subskribenten, die mit QM3 verbunden sind,  $T_B$  oder  $T_C$  veröffentlichen oder subscribieren.

## 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 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.

Warteschlangenmanager der Version 7 oder höher zu vorhandenen Topic-Bereichen der IBM WebSphere MQ 6 hinzufügen

Erweitern Sie ein vorhandenes Publish/Subscribe-System der IBM WebSphere MQ 6, um mit einem Warteschlangenmanager der Version 7 oder höher zu interagieren und dabei dieselben Topic-Bereiche gemeinsam zu nutzen.

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 abonnieren 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.

## **Zugehörige Informationen**

Verteilte Publish/Subscribe-Netzwerke

Themenbereiche

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 eingereihte Publish/Subscribe starten .
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 .
  - 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 zum Erstellen einer Punkt-zu-Punkt-Kanalverbindung finden Sie unter IBM MQ - Verfahren zur verteilten Steuerung von Warteschlangen .

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

Im ersten Beispiel wird gezeigt, wie der Warteschlangenmanager QM2 als untergeordnetes Element von QM1 zugeordnet wird, und anschließend QM2 abfragen, um zu bestätigen, dass es erfolgreich zu einem untergeordneten Element mit einer **STATUS** von **ACTIVE** geworden ist:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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, 2023. 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, überprüfen Sie, ob die Kanäle zwischen dem untergeordneten Element 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. Sie können verwaltete Themen so konfigurieren, dass sie die Ebene der gemeinsamen Nutzung durch die Verwendung der Themeneigenschaften **SUBSCOPE** und **PUBSCOPE** steuern. Siehe [Verteilte Publish/Subscribe-Netze konfigurieren](#).

### 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, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLROUTE** fest.

#### 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 Informationen

#### 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, 2023. 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, legen Sie die Eigenschaft **CLUSTER** fest. Wenn Sie das Routing auswählen möchten, das für Veröffentlichungen und Subskriptionen für dieses Thema verwendet werden soll, legen Sie die Eigenschaft **CLRROUTE** fest.

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.

ULW

## 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 UNIX, Linux, and Windows.

### Prozedur

- Verwenden Sie die Informationen in den folgenden Links, um Ihre -Installationen zu konfigurieren:
  - [„Primäre Installation ändern“ auf Seite 419](#)
  - [„WS-Manager einer Installation zuordnen“ auf Seite 421](#)
  - [„Anwendungen in einer Umgebung mit mehreren Installationen verbinden“ auf Seite 409](#)

## Anwendungen in einer Umgebung mit mehreren Installationen verbinden

Wenn auf UNIX, Linux, and Windows-Systemen Bibliotheken von IBM WebSphere MQ 7.1 oder höher geladen werden, verwendet IBM MQ automatisch die richtigen Bibliotheken, ohne dass weitere Maßnahmen des Benutzers 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 411.

### 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.

Werden jedoch Bibliotheken von IBM WebSphere MQ 7.1 oder höher gefunden und geladen, lädt und verwendet IBM MQ die entsprechenden Bibliotheken, ohne dass weitere Maßnahmen des Benutzers erforderlich sind. Wenn die Anwendung eine Verbindung zu einem Warteschlangenmanager herstellt, lädt IBM MQ Bibliotheken aus der Installation, welcher der Warteschlangenmanager zugeordnet ist.

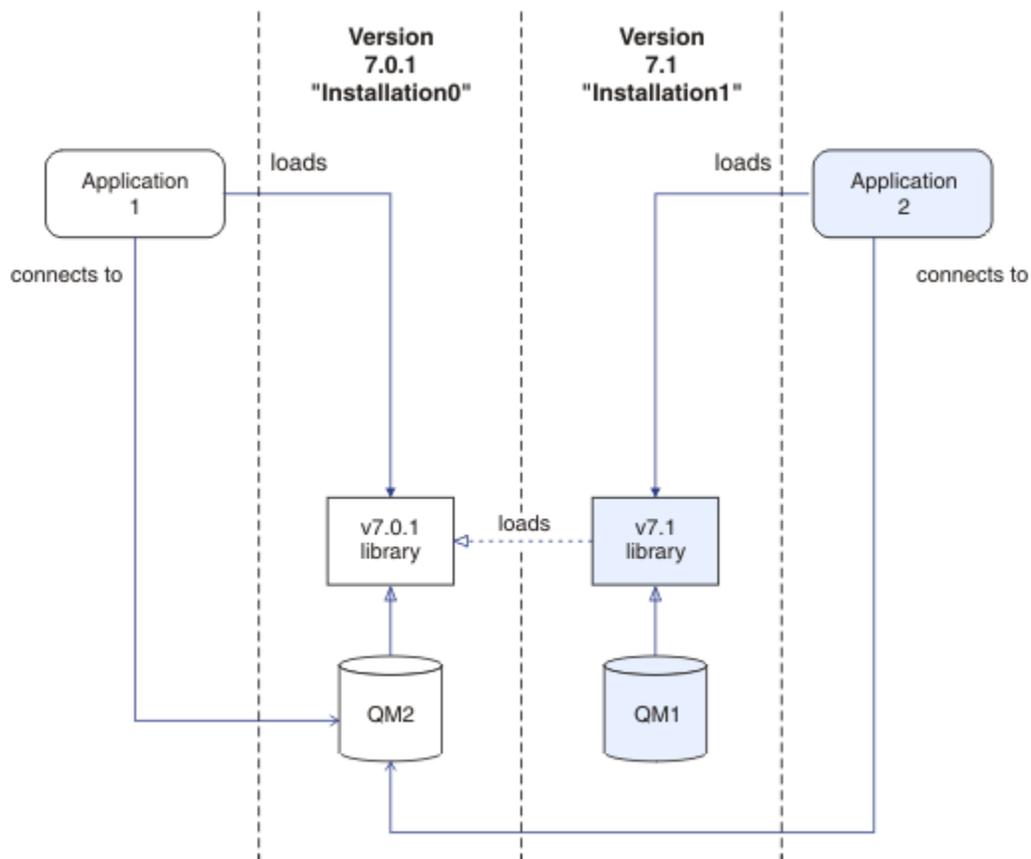


Abbildung 69. Anwendungen in einer Umgebung mit mehreren Installationen verbinden

Abbildung 69 auf Seite 410 zeigt beispielsweise eine Umgebung mit mehreren Installationen mit einer IBM WebSphere MQ 7.0.1 -Installation ( Installation0) und einer IBM WebSphere MQ 7.1 -Installation ( Installation1). Zwei Anwendungen sind mit diesen Installationen verbunden, aber sie laden unterschiedliche Bibliotheksversionen.

Application 1 lädt eine IBM WebSphere MQ 7.0.1 -Bibliothek direkt. Wenn application 1 eine Verbindung zu QM2herstellt, werden die IBM WebSphere MQ 7.0.1 -Bibliotheken verwendet. Wenn application 1 versucht, eine Verbindung zu QM1herzustellen, bzw. wenn QM2 Installation1zugeordnet ist, schlägt application 1 mit dem Fehler 2059 (080B) (RC2059) fehl: MQRC\_Q\_MGR\_NOT\_AVAILABLE Die Anwendung schlägt fehl, da die Bibliothek von IBM WebSphere MQ 7.0.1 nicht in der Lage ist, andere Bibliotheksversionen zu laden. Das heißt, wenn Bibliotheken von IBM WebSphere MQ 7.0.1 direkt geladen werden, können Sie keinen Warteschlangenmanager verwenden, der einer Installation einer späteren Version von IBM MQ zugeordnet ist.

Application 2 lädt eine IBM WebSphere MQ 7.1 -Bibliothek direkt. Wenn application 2 eine Verbindung zu QM2herstellt, lädt und verwendet die Bibliothek IBM WebSphere MQ 7.1 die Bibliothek IBM WebSphere MQ 7.0.1 . Wenn application 2 eine Verbindung zu QM1herstellt oder wenn QM2 Installation1zugeordnet ist, wird die IBM WebSphere MQ 7.1 -Bibliothek geladen und die Anwendung funktioniert wie erwartet.

Die Migrationsszenarios sowie die Verbindung von Anwendungen mit mehreren Installationen werden im Abschnitt Koexistenz mehrerer Warteschlangenmanager unterschiedlicher Installationen unter UNIX, Linux, and Windows ausführlicher behandelt.

Weitere Informationen zum Laden von Bibliotheken von IBM WebSphere MQ 7.1 finden Sie im Abschnitt „IBM MQ-Bibliotheken laden“ auf Seite 411.

## Unterstützung und Einschränkungen

Wenn eine der folgenden Bibliotheken von IBM WebSphere MQ 7.1 oder höher gefunden und geladen werden, kann IBM MQ 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 in [„Einschränkungen für Anwendungen mit mehreren Installationen“](#) auf Seite 416.

### Zugehörige Konzepte

[„Einschränkungen für Anwendungen mit mehreren Installationen“](#) auf Seite 416

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 411

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 ändern“](#) auf Seite 419

Sie können den Befehl **setmqinst** verwenden, um eine Installation als primäre Installation festzulegen oder zu dekonfigurieren.

[„WS-Manager einer Installation zuordnen“](#) auf Seite 421

Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den Befehl **crtmqm** ausgegeben hat. Unter UNIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

### Zugehörige Informationen

[Primäre Installation auswählen](#)

## 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.

Diese Informationen beziehen sich auf IBM WebSphere MQ 7.1 oder spätere Versionen von Bibliotheken.

Wie IBM MQ-Bibliotheken lokalisiert und geladen werden, hängt von der Installationsumgebung ab:

- Wenn auf UNIX and Linux-Systemen eine Kopie von IBM WebSphere MQ 7.1 oder eine aktuellere Version in der Standardposition installiert ist, funktionieren die vorhandenen Anwendungen weiterhin auf dieselbe Weise wie frühere Versionen. Wenn die Anwendungen jedoch symbolische Links in `/usr/lib` benötigen, müssen Sie entweder eine IBM WebSphere MQ 7.1-Installation oder eine aktuellere Version als primäre Installation auswählen oder die symbolischen Links manuell erstellen.

- Wenn IBM WebSphere MQ 7.1 oder eine aktuellere Version in einer Nicht-Standardposition installiert ist, die der Fall ist, wenn IBM WebSphere MQ 7.0.1 ebenfalls installiert ist, müssen Sie möglicherweise Ihre vorhandenen Anwendungen so ändern, dass 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 414.

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.

<i>Tabelle 26. Vorteile und Nachteile der Optionen zum Laden von Bibliotheken</i>			
<b>Plattform</b>	<b>Option</b>	<b>Leistungen</b>	<b>Rückzugsschragen</b>
  UNIX and Linux-Systeme	<p>Legen Sie den eingebetteten Laufzeitsuchpfad (RPath) der Anwendung fest oder ändern Sie diesen.</p> <p>Mit dieser Option müssen Sie die Anwendung erneut kompilieren und verknüpfen. Weitere Informationen zum Kompilieren und Verknüpfen von Anwendungen finden Sie unter <a href="#">Erstellen einer prozeduralen Anwendung</a>.</p>	<ul style="list-style-type: none"> <li>• Der Umfang der Änderung ist klar.</li> </ul>	<ul style="list-style-type: none"> <li>• Sie müssen in der Lage sein, die Anwendung erneut zu kompilieren und zu verlinken.</li> <li>• Wenn sich die Speicherposition von IBM MQ ändert, müssen Sie den Laufzeit-Suchpfad ändern.</li> </ul>

Tabelle 26. Vorteile und Nachteile der Optionen zum Laden von Bibliotheken (Forts.)

Plattform	Option	Leistungen	Rückzugsschrägen
UNIX and Linux-Systeme	<p>Setzen Sie die Umgebungsvariable <code>LD_LIBRARY_PATH</code> unter Verwendung von <code>setmqenv</code> oder <code>crtmqenv</code> mit der Option <b>-k</b> oder <b>-l</b>. (</p> <p><b>AIX</b> Unter AIX ist diese Umgebungsvariable <code>LIBPATH</code></p>	<ul style="list-style-type: none"> <li>• Es sind keine Änderungen an vorhandenen Anwendungen erforderlich.</li> <li>• Überschreibt eingebettete RPaths in einer Anwendung.</li> <li>• Die Variable lässt sich leicht ändern, wenn sich die Speicherposition von IBM MQ ändert.</li> </ul>	<ul style="list-style-type: none"> <li>• <code>setuid</code> -und <code>setgid</code> -Anwendungen oder Anwendungen, die auf andere Weise erstellt wurden, können <code>LD_LIBRARY_PATH</code> aus Sicherheitsgründen ignorieren.</li> <li>• Umgebungsspezifisch, daher muss in jeder Umgebung, in der die Anwendung ausgeführt wird, festgelegt werden.</li> <li>• Mögliche Auswirkungen auf andere Anwendungen, die auf <code>LD_LIBRARY_PATH</code> basieren.</li> <li>• <b>HP-UX</b> HP-UX: Optionen, die beim Kompilieren der Anwendungen verwendet wurden, könnten die Verwendung von <code>LD_LIBRARY_PATH</code> inaktivieren. Weitere Informationen finden Sie im Abschnitt <a href="#">Überlegungen zur Laufzeitverknüpfung für HP-UX</a>.</li> <li>• <b>Linux</b> Linux: Der zum Erstellen der Anwendung verwendete Compiler könnte die Verwendung von <code>LD_LIBRARY_PATH</code> inaktivieren. Weitere Informationen finden Sie im Abschnitt <a href="#">Überlegungen zur Laufzeitverknüpfung für Linux</a>.</li> </ul>
<b>Windows</b> Windows-Systeme	<p>Legen Sie die Variable <code>PATH</code> mit <code>setmqenv</code> oder <code>crtmqenv</code> fest.</p>	<ul style="list-style-type: none"> <li>• Für vorhandene Anwendungen sind keine Änderungen erforderlich.</li> <li>• Die Variable lässt sich leicht ändern, wenn sich die Speicherposition von IBM MQ ändert.</li> </ul>	<ul style="list-style-type: none"> <li>• Umgebungsspezifisch, daher muss in jeder Umgebung, in der die Anwendung ausgeführt wird, festgelegt werden.</li> <li>• Mögliche Auswirkungen auf andere Anwendungen.</li> </ul>

Tabelle 26. Vorteile und Nachteile der Optionen zum Laden von Bibliotheken (Forts.)

Plattform	Option	Leistungen	Rückzugsschrauben
 UNIX, Linux, and Windows-Systeme	Setzen Sie die primäre Installation auf IBM WebSphere MQ 7.1 oder eine höhere Installation fest. Siehe „Primäre Installation ändern“ auf Seite 419.  Weitere Informationen zur primären Installation finden Sie im Abschnitt <a href="#">Primäre Installation auswählen</a> .	<ul style="list-style-type: none"> <li>Für vorhandene Anwendungen sind keine Änderungen erforderlich.</li> <li>Die primäre Installation lässt sich leicht ändern, wenn sich die Speicherposition von IBM MQ ändert.</li> <li>Ähnliches Verhalten für vorherige Versionen von IBM MQ.</li> </ul>	<ul style="list-style-type: none"> <li>Wenn IBM WebSphere MQ 7.0.1 installiert ist, können Sie die primäre Installation nicht auf IBM WebSphere MQ 7.1 oder höher setzen.</li> <li>   UNIX and Linux: Funktioniert nicht, wenn <code>/usr/lib</code> nicht der Standardsuchpfad ist.                     </li> </ul>

## Überlegungen zum Laden von Bibliotheken in HP-UX

### 

Die Beispielp compilierungsbefehle in der Produktdokumentation für Vorgängerversionen von IBM MQ enthielten die Linkoption `-W1, +noenvvar` für 64-Bit-Anwendungen. Mit dieser Option wird die Verwendung von `LD_LIBRARY_PATH` zum Laden von gemeinsam genutzten Bibliotheken inaktiviert. Wenn Ihre Anwendungen IBM MQ-Bibliotheken von einer anderen als der im Laufzeitsuchpfad angegebenen Position laden sollen, müssen Sie die Anwendungen aktualisieren. Sie können die Anwendungen aktualisieren, indem Sie ohne die Linkoption `-W1, +noenvvar` erneut kompilieren und verknüpfen oder den Befehl `chatx` verwenden.

Wenn Sie wissen möchten, wie Ihre Anwendungen derzeit Bibliotheken laden, lesen Sie den Abschnitt „Lademechanismen für die Betriebssystembibliothek“ auf Seite 414.

## Überlegungen zum Laden von Bibliotheken in 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.

## Überlegungen zum Laden von Bibliotheken in Solaris

### 

Die Beispielp compilierungsbefehle in der Produktdokumentation für Vorgängerversionen von IBM MQ enthalten die `-lmqmc -lmqzse`-Linkoptionen. Die entsprechenden Versionen dieser Bibliotheken werden nun automatisch von IBM MQ geladen. Wenn IBM MQ nicht im Standardverzeichnis installiert ist oder auf dem System mehrere Installationen vorhanden sind, müssen Sie Ihre bestehenden Anwendungen aktualisieren. Sie können die Anwendungen aktualisieren, indem Sie sie erneut kompilieren und verknüpfen, ohne dass die `-lmqmc -lmqzse`-Linkoptionen angegeben sind.

## 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** **UNIX** Auf UNIX and Linux-Systemen können verschiedene Methoden zur Lokalisierung der zu ladenden Bibliotheken verwendet worden sein:

- Mit Hilfe der Umgebungsvariable *LD\_LIBRARY\_PATH* (alternativ *LIBPATH* unter AIX und *SHLIB\_PATH* unter HP-UX). 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 WebSphere MQ 7.1 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. Beispiel:

- **AIX** AIX: **dump**
- **HP-UX** HP-UX: **chatr**
- **Linux** Linux: **readelf**
- **Solaris** Solaris: **elfdump**

### Zugehörige Konzepte

„Einschränkungen für Anwendungen mit mehreren Installationen“ auf Seite 416

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 409

Wenn auf UNIX, Linux, and Windows-Systemen Bibliotheken von IBM WebSphere MQ 7.1 oder höher geladen werden, verwendet IBM MQ automatisch die richtigen Bibliotheken, ohne dass weitere Maßnahmen des Benutzers 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 ändern“ auf Seite 419

Sie können den Befehl **setmqinst** verwenden, um eine Installation als primäre Installation festzulegen oder zu dekonfigurieren.

„WS-Manager einer Installation zuordnen“ auf Seite 421

Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den Befehl **crtmqm** ausgegeben hat. Unter UNIX, Linux, and Windows können Sie die Installation ändern, die einem Warteschlangenmanager zugeordnet ist, indem Sie den Befehl **setmqm** verwenden.

### Zugehörige Informationen

Primäre Installation auswählen

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](#).

### Nachrichten Kennungen

Nachrichten Kennungen, die den Sonderwert MQHC\_UNASSOCIATED\_HCONN verwenden, sind begrenzt auf die Verwendung mit der ersten Installation, die in einem Prozess geladen wird. Wenn die Nachrichten Kennung nicht von einer bestimmten Installation verwendet werden kann, wird der Ursachencode MQRC\_HMSG\_NOT\_AVAILABLE zurückgegeben.

Diese Einschränkung betrifft Nachrichteneigenschaften. Sie können keine Nachrichten Kennungen verwenden, um Nachrichteneigenschaften von einem Warteschlangenmanager in einer Installation abzurufen und sie in einen Warteschlangenmanager in einer anderen Installation zu versetzen. Weitere Informationen zu Nachrichten Kennungen finden Sie in [MQCRTMH-Create message handle](#).

### Exits

In einer Umgebung mit Mehrfachinstallationen müssen bereits vorhandene Exits für die Verwendung von IBM WebSphere MQ 7.1-Installationen oder höher entsprechend 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 Verbindung zu den IBM MQ-Bibliotheken herstellen. Weitere Informationen finden Sie im Abschnitt [Exits und installierbare Services in UNIX, Linux, and Windows schreiben](#).

### Direktaufruf

Auf einem Server mit Mehrfachinstallationen müssen Anwendungen, die eine Direktaufrufverbindung mit IBM WebSphere MQ 7.1 oder höher verwenden, folgende 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. Der Versuch, eine Verbindung herzustellen, führt zu einem Fehler und dem 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. Die Direktaufrufverbindung zu einem Warteschlangenmanager, der mit IBM WebSphere MQ 7.1 oder einer höheren Installation verknüpft ist, hat keinen Einfluss darauf, ob eine Anwendung Direktaufrufverbindungen herstellen kann.

4. Es ist nicht möglich, eine Verbindung zu einem Warteschlangenmanager zu kombinieren, der einer IBM WebSphere MQ 7.0.1 -Installation zugeordnet ist, und eine Verbindung zu einem Warteschlangenmanager herzustellen, der einer Installation von IBM WebSphere MQ 7.1 oder höher zugeordnet ist.

Wenn `AMQ_SINGLE_INSTALLATION` festgelegt ist, kann jede Verbindung zu einem Warteschlangenmanager als Direktaufrufverbindung hergestellt werden. Anderenfalls gelten nahezu dieselben Einschränkungen:

- Die Installation muss dieselbe sein, von der aus die Laufzeitbibliotheken von IBM MQ geladen wurden.
- Jede Verbindung in demselben Prozess muss zu derselben Installation hergestellt werden. Wenn Sie versuchen, eine Verbindung zu einem Warteschlangenmanager herzustellen, der mit einer anderen Installation verknüpft ist, schlägt die Verbindung mit dem Ursachencode `MQRC_INSTALLATION_MISMATCH` fehl. Wenn `AMQ_SINGLE_INSTALLATION` festgelegt ist, gilt diese Einschränkung nicht nur für Direktaufrufverbindungen, sondern für alle Verbindungen.
- Stellen Sie Direktaufrufverbindungen nur zu einem einzigen Warteschlangenmanager her.

### Zugehörige Informationen

[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](#)

## .NET-Anwendungen in einer Umgebung mit mehreren Installationen verbinden

Standardmäßig verwenden Anwendungen die .NET-Assemblies von der primären Installation. Wenn keine primäre Installation vorhanden ist oder Sie die primären Installationsassemblies nicht verwenden möchten, müssen Sie die Anwendungskonfigurationsdatei oder die `DEVPATH` -Umgebungsvariable aktualisieren.

Wenn auf dem System eine primäre Installation vorhanden ist, werden die .NET-Assemblies und die Richtliniendateien dieser Installation im globalen Assemblycache (Global Assembly Cache, GAC) registriert. Die .NET-Assemblies für alle anderen Installationen befinden sich im Installationspfad jeder Installation, aber die Assemblies sind nicht in der GAC registriert. Daher werden Anwendungen standardmäßig mit den .NET-Assemblies 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 Installationsassemblies 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 417.

Sie müssen die Umgebungsvariable `DEVPATH` aktualisieren, wenn der folgende Fall wahr ist:

- Sie möchten, dass Ihre Anwendung die Assemblies 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 419.

### .NET-Anwendungen mit der Anwendungskonfigurationsdatei verbinden

Innerhalb der Anwendungskonfigurationsdatei müssen Sie verschiedene Tags für die Umleitung von Anwendungen festlegen, um Assemblies 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 27. 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: <ul style="list-style-type: none"> <li>• Verwenden Sie den Tag <i>bindingRedirect</i>, um die Verwendung der früheren Version der Assemblys anzugeben, die sich in der GAC befinden.</li> </ul>
So führen Sie eine Anwendung mit einer späteren Version von IBM MQ nicht der primären Installation aus. (spätere Versionsbaugruppen im Installationsordner):	In der Anwendungskonfigurationsdatei: <ul style="list-style-type: none"> <li>• Verwenden Sie den Tag <i>codebase</i>, um auf die Position der späteren Assemblys zu verweisen.</li> <li>• Verwenden Sie den Tag <i>bindingRedirect</i>, um die Verwendung der späteren Assemblys anzugeben.</li> </ul>	In der Anwendungskonfigurationsdatei: <ul style="list-style-type: none"> <li>• Verwenden Sie den Tag <i>codebase</i>, um auf die Position der späteren Assemblys zu verweisen.</li> </ul>
So führen Sie eine Anwendung mit einer früheren Version von IBM MQ nicht der primären Installation aus. (ältere Versionsbaugruppen im Installationsordner):	In der Anwendungskonfigurationsdatei: <ul style="list-style-type: none"> <li>• Verwenden Sie den Tag <i>codebase</i>, um auf die Position der früheren Versionsbaugruppen zu verweisen.</li> <li>• Tag <i>publisherpolicy Apply=no</i> einschließen</li> </ul>	In der Anwendungskonfigurationsdatei: <ul style="list-style-type: none"> <li>• Verwenden Sie den Tag <i>codebase</i>, um auf die Position der früheren Versionsbaugruppen zu verweisen.</li> <li>• Verwenden Sie den Tag <i>bindingRedirect</i>, um die Verwendung der früheren Versionsbaugruppen anzugeben.</li> <li>• Tag <i>publisherpolicy Apply=no</i> einschließen</li> </ul>

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 409

Wenn auf UNIX, Linux, and Windows-Systemen Bibliotheken von IBM WebSphere MQ 7.1 oder höher geladen werden, verwendet IBM MQ automatisch die richtigen Bibliotheken, ohne dass weitere Maßnahmen des Benutzers erforderlich sind. IBM MQ verwendet Bibliotheken aus der Installation, die dem Warteschlangenmanager zugeordnet ist, zu dem die Anwendung eine Verbindung herstellt.

### Zugehörige Informationen

[Primäre Installation auswählen](#)

[.NET verwenden](#)

[Mehrere Installationen](#)

ULW

## Primäre Installation ändern

Sie können den Befehl **setmqinst** verwenden, um eine Installation als primäre Installation festzulegen oder zu dekonfigurieren.

### Informationen zu diesem Vorgang

Diese Aufgabe bezieht sich auf UNIX, 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](#).

Wenn eine Installation von IBM WebSphere MQ 7.1 oder höher neben einer Installation von IBM WebSphere MQ 7.0.1 existiert, muss die IBM WebSphere MQ 7.0.1-Installation die primäre Installation sein. Sie wird als primäre Installation markiert, wenn IBM WebSphere MQ 7.1 oder eine höhere Version installiert ist und IBM WebSphere MQ 7.1 oder höher nicht als primäre Installation ausgewählt werden kann.

**Windows** Während des Installationsprozesses unter Windows können Sie angeben, dass die Installation die primäre Installation sein soll.

**Linux** **UNIX** Auf UNIX and Linux-Systemen müssen Sie nach der Installation den Befehl **setmqinst** ausgeben, um die Installation als die primäre Installation festzulegen.

„Primäre Installation festlegen“ auf Seite 420.

„Deinstallieren der primären Installation“ auf Seite 420.

## Primäre Installation festlegen

### Vorgehensweise

Gehen Sie wie folgt vor, um eine Installation als primäre Installation festzulegen:

1. Überprüfen Sie, ob eine Installation bereits die primäre Installation ist, indem Sie den folgenden Befehl eingeben:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

Dabei steht *MQ\_INSTALLATION\_PATH* für den Installationspfad einer Installation von IBM WebSphere MQ 7.1 oder höher.

2. Wenn eine bestehende Installation von IBM WebSphere MQ 7.1 oder höher als die primäre Installation festgelegt ist, heben Sie diese Festlegung mithilfe der Anweisungen im Abschnitt „[Deinstallieren der primären Installation](#)“ auf Seite 420 auf. Wenn IBM WebSphere MQ 7.0.1 auf dem System installiert ist, kann die primäre Installation nicht geändert werden.
3. Stellen Sie sicher, dass Sie mit der entsprechenden Berechtigung angemeldet sind:

-  Als Root in UNIX and Linux.
-  Als Mitglied der Gruppe "Administratoren" auf Windows-Systemen.

4. 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
```

5.  Führen Sie auf Windows-Systemen einen Neustart des Systems durch.

## Deinstallieren der primären Installation

### Vorgehensweise

Gehen Sie wie folgt vor, um eine Installation als primäre Installation zu dekonfigurieren:

1. Überprüfen Sie, welche Installation die primäre Installation ist, indem Sie den folgenden Befehl eingeben:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

Dabei steht *MQ\_INSTALLATION\_PATH* für den Installationspfad einer Installation von IBM WebSphere MQ 7.1 oder höher.

Wenn IBM WebSphere MQ 7.0.1 die primäre Installation ist, kann die Festlegung der primären Installation nicht aufgehoben werden.

2. Stellen Sie sicher, dass Sie mit der entsprechenden Berechtigung angemeldet sind:

-  Als Root in UNIX and Linux.
-  Als Mitglied der Gruppe "Administratoren" auf Windows-Systemen.

3. 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 Informationen

[Komponenten, die nur mit der primären Installation unter Windows verwendet werden können](#)

[Verknüpfungen von externen Speicherarchiven und Steuerbefehlen zur primären Installation von UNIX and Linux](#)

[Deinstallieren, Durchführen eines Upgrades und Wartung der primären Installation](#)

[Auswählen eines Installationsnamens](#)

[setmqinst](#)

**ULW**

## WS-Manager einer Installation zuordnen

Wenn Sie einen WS-Manager erstellen, wird er automatisch der Installation zugeordnet, die den Befehl **crtmqm** ausgegeben hat. Unter UNIX, 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 Installation, die dem WS-Manager zugeordnet ist. 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 Installationsname, der durch den Befehl **setmqm** angegeben wird, muss mit der Installation übereinstimmen, von der der Befehl abgesetzt wird.
- **strmqm** muss in der Regel von der Installation aus abgesetzt werden, die dem Warteschlangenmanager zugeordnet ist. Wenn jedoch ein Warteschlangenmanager von IBM WebSphere MQ 7.0.1 oder früher auf einer Installation von IBM WebSphere MQ 7.1 oder höher zum ersten Mal gestartet wird, kann **strmqm** verwendet werden. In diesem Fall startet **strmqm** den Warteschlangenmanager und ordnet ihn der Installation zu, von der aus der Befehl abgesetzt wird.
- **dspmq** zeigt Informationen zu allen Warteschlangenmanagern auf einem System an, nicht nur die WS-Manager, die derselben Installation wie der Befehl **dspmq** zugeordnet sind. Der Befehl **dspmq -o installation** zeigt Informationen zu den Warteschlangenmanagern an, denen die Installationen zugeordnet sind.

Bei Hochverfügbarkeitsumgebungen ordnet der Befehl **addmqinf** den Warteschlangenmanager automatisch der Installation zu, von der der Befehl **addmqinf** ausgegeben wird. Wenn der Befehl **strmqm** dann von derselben Installation wie der Befehl **addmqinf** ausgegeben wird, ist keine weitere Konfiguration erforderlich. Um den Warteschlangenmanager unter Verwendung einer anderen Installation zu starten, müssen Sie zuerst die zugeordnete Installation mit dem Befehl **setmqm** ändern.

Wenn Sie einem WS-Manager eine 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 Warteschlangenmanager mit dem Befehl **endmqm** von der Installation, die derzeit dem Warteschlangenmanager zugeordnet ist.
2. Ordnen Sie den WS-Manager mit dem Befehl **setmqm** einer anderen Installation von dieser 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** von der Installation aus, 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 wenn die Statusinformationen des Warteschlangenmanagers nicht verfügbar sind, kann der Warteschlangenmanager mit dem Befehl **setmqm** keine andere Installation zuordnen. Nehmen Sie in dieser Situation die folgenden Aktionen vor:

1. Verwenden Sie den Befehl **dspmqinst**, um die anderen Installationen auf Ihrem System anzuzeigen.
2. Ändern Sie das `InstallationName`-Feld der Zeilengruppe `QueueManager` in `mqs.ini` manuell, um eine andere Installation anzugeben.
3. Verwenden Sie den Befehl **dltmqm** von dieser Installation aus, um den Warteschlangenmanager zu löschen.

## Zugehörige Konzepte

[„Installationen von IBM MQ auf einem System finden“](#) auf Seite 422

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 96

Die IBM MQ-Konfigurationsdatei `mqs.ini` enthält Informationen, die für alle WS-Manager auf dem Knoten relevant sind. Sie wird automatisch während der Installation erstellt.

## Zugehörige Informationen

[Primäre Installation auswählen](#)

[addmqinf](#)

[dspmq](#)

[dspmqinst](#)

[endmqm](#)

[setmqm](#)

[strmqm](#)

ULW

## 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:

- Verwenden Sie den Befehl **dspmqr**. Der Befehl liefert keine Details zu allen Installationen auf einem System, wenn er in einer IBM WebSphere MQ 7.0.1-Installation ausgegeben wird.
- Fragen Sie mithilfe der Plattforminstallationstools ab, wo IBM MQ installiert wurde. Verwenden Sie dann den Befehl **dspmqr** in einer IBM WebSphere MQ 7.1-Installation oder höher. Mit den folgenden Beispielbefehlen können Sie abfragen, wo IBM MQ installiert ist:

- Auf AIX-Systemen können Sie den Befehl **lslpp** verwenden:

```
lslpp -R ALL -l mqm.base.runtime
```

- Auf HP-UX-Systemen können Sie den Befehl **swlist** verwenden:

```
swlist -a location -a revision -l product MQSERIES
```

- Auf Linux-Systemen können Sie den Befehl **rpm** verwenden:

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

- Unter Solaris verwenden Sie die Befehle **pkginfo** und **pkgparam**:

1. Listen Sie die installierten Pakete auf, indem Sie den folgenden Befehl eingeben:

```
pkginfo | grep -w mqm
```

2. Geben Sie für jedes aufgelistete Paket den folgenden Befehl ein:

```
pkgparam pkgname BASEDIR
```

- 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, Version, InstallLocation
```

- Geben Sie auf UNIX and Linux-Systemen den folgenden Befehl aus, um zu ermitteln, wo IBM MQ installiert wurde:

```
cat /etc/opt/mqm/mqinst.ini
```

Verwenden Sie dann den Befehl **dspmqr** in einer IBM WebSphere MQ 7.1-Installation oder höher.

- 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
```

- 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
```

**Hinweis:** Der Befehl **reg.exe** zeigt nur Informationen für Installationen von IBM WebSphere MQ 7.1 oder höher an.

### Zugehörige Informationen

[dspmqr](#)

[dspmqrinst](#)

## 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

 Unter z/OS wird die Hochverfügbarkeit in die Plattform integriert. Sie können die Verfügbarkeit der Serveranwendung auch durch Verwendung von Gruppen mit gemeinsamer Warteschlange verbessern. Weitere Informationen finden Sie im Abschnitt [Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange](#).

 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
-  HA-Cluster unter IBM i
-   PowerHA für AIX (früher HACMP unter AIX) und andere UNIX and Linux-Clustering-Lösungen

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 Aktivitäten 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.

## 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 28 auf Seite 426](#) 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 29 auf Seite 426](#) sind weitere Voraussetzungen für die Aktivierung der automatischen Clientwiederverbindung aufgeführt.

Mit dem Programmzugriff auf Verbindungswiederholungsoptionen kann eine Clientanwendung Verbindungsoptionen festlegen. Mit Ausnahme von JMS- und XMS-Clients kann, wenn eine Clientanwendung Zugriff auf Verbindungswiederholungsoptionen hat, auch ein Ereignishandler erstellt werden, um Verbindungswiederherstellungsergebnisse zu verarbeiten.

Eine vorhandene Clientanwendung kann ohne erneute Kompilierung und Verlinkung von der Unterstützung für die erneute Verbindung profitieren können:

- Geben Sie für einen Nicht-JMS-Client die Umgebungsvariable `DefRecon` der Datei `mqclient.ini` an, um die Verbindungswiederholungsoptionen 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.
- Geben Sie für einen JMS-Client die Verbindungswiederholungsoptionen in der Konfiguration der Verbindungsfactory an. Bei der Ausführung im EJB-Container eines Java EE-Servers können MDBs die Verbindung zu IBM MQ wiederherstellen, indem sie den Mechanismus für die Verbindungswiederherstellung verwenden, der von den Aktivierungsspezifikationen des IBM MQ -Ressourcenadapters bereitgestellt wird (oder Listener-Ports, wenn sie in WebSphere Application Serverausgeführt werden). 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 bietet diese Möglichkeit zur Wiederherstellung der Verbindung für die Zustellung von Nachrichten an Message-driven Beans, aber andere Java EE-Elemente wie Servlets müssen ihre eigene Verbindungswiederholung implementieren.

**Anmerkung:** Die automatische Clientverbindungswiederholung wird von IBM MQ classes for Java nicht unterstützt.

Tabelle 28. Unterstützte Clients

Clientschnittstelle	Client	Programmzugriff auf Verbindungswiederschlussoptionen	Reconnection-Unterstützung
Messaging-APIs	C, C++, COBOL, Unmanaged Visual Basic, XMS (Unmanaged XMS unter Windows)	7.0.1	7.0.1
	JMS (JSE und 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 <sup>1</sup> )	Nicht unterstützt	7.0.1
	Windows Communication Foundation (verwaltet <sup>1</sup> )	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

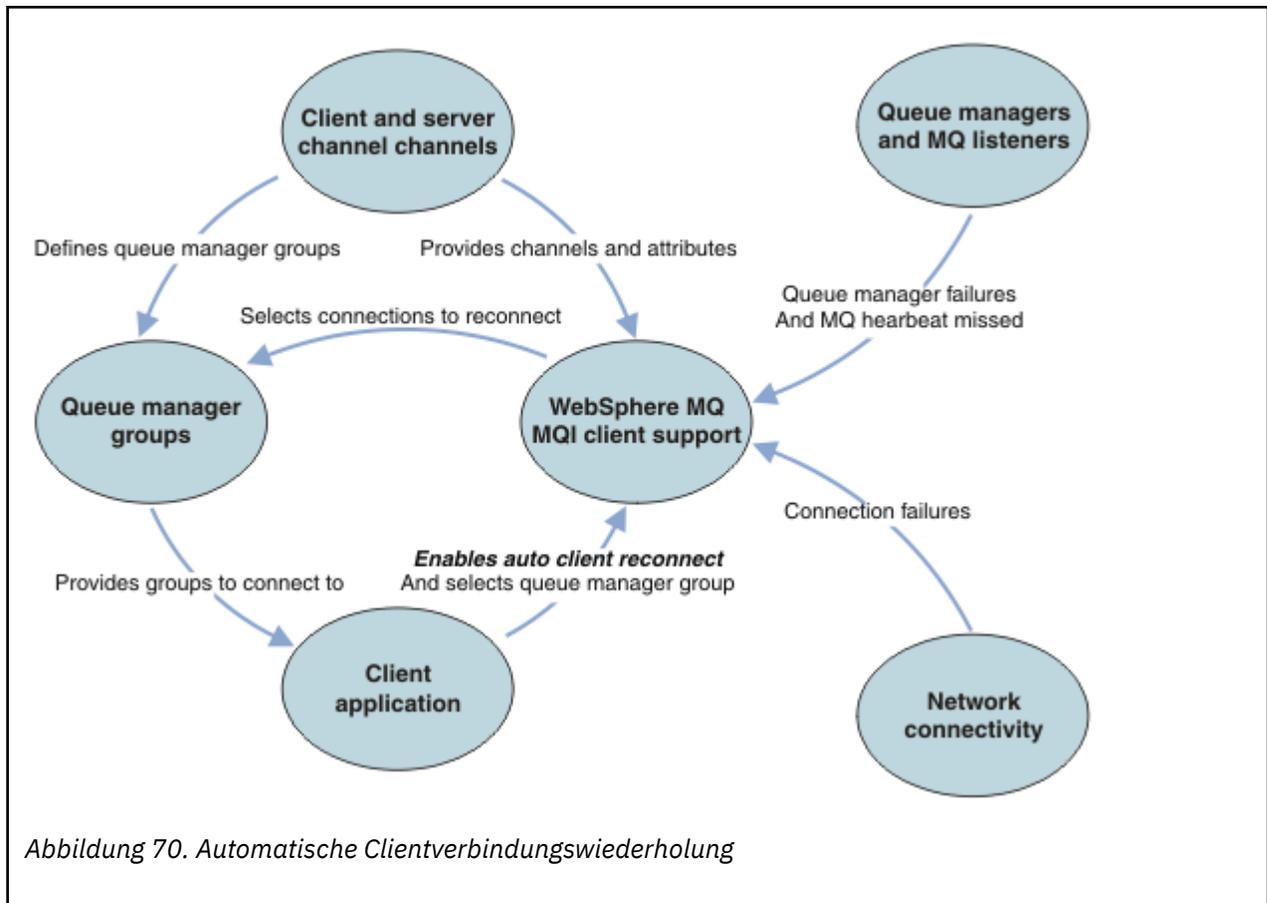
1. Definieren Sie den verwalteten oder nicht verwalteten Modus in der WCF-Bindungskonfiguration.

Die automatische Verbindungswiederverbindung hat die folgenden Konfigurationsanforderungen:

Tabelle 29. Konfigurationsanforderungen für die automatische Verbindungswiederverbindung

Komponente	Voraussetzung	Auswirkung der Anforderung nicht erfüllen
IBM MQ MQI client-Installation	Im Abschnitt <a href="#">Tabelle 28 auf Seite 426</a>	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
MQI	Einer von: <ul style="list-style-type: none"> <li>MQCONNX, wobei MQCNO Optionen auf MQCNO_RECONNECT oder MQCNO_RECONNECT_Q_MGR gesetzt ist.</li> <li>Defrecon=YES   QMGR in mqclient.ini</li> <li>Geben Sie in JMS die Eigenschaft CLIENTRECONNECTOPTIONS für die Verbindungsfactory an.</li> </ul>	MQCC_FAILED, wenn eine Verbindung unterbrochen wird oder der Warteschlangenmanager beendet wird oder fehlschlägt.

Abbildung 70 auf Seite 427 zeigt die Hauptinteraktionen zwischen Komponenten, die an der Clientwieder-  
 verbinding beteiligt sind.



## Clientanwendung

Die Clientanwendung ist ein IBM MQ MQI client.

- Standardmäßig werden die Clients nicht automatisch erneut verbunden. Aktivieren Sie die automatische Clientverbindungs-wiederholung, indem Sie die Option `MQCONN MQCNO MQCNO_RECONNECT` oder `MQCNO_RECONNECT_Q_MGR` festlegen.
- Viele Anwendungen sind so geschrieben, dass sie in der Lage sind, die automatische Verbindungs-wiederherstellung ohne zusätzliche Codierung zu nutzen. Aktivieren Sie die automatische Verbindungs-wiederherstellung für vorhandene Programme, ohne Codierungsänderungen vorzunehmen, indem Sie das Attribut `DefRecon` in der Zeilengruppe für die Kanäle der `mqclient.ini`-Konfigurationsdatei 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.
  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 `MQCONN 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 521.

- 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 `MQCNO MQCNO_RECONNECT_Q_MGR`, um eine Verbindung zu einer Instanz desselben Warteschlangenmanagers herzustellen.

- Ein Client kann einen Ereignishandler registrieren, so dass er den Status der Verbindungswiederverbindung erhalten kann. Der `MQHCONN`, der an den Ereignishandler übergeben wurde, 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 Verbindungswiederverbindung durchgeführt 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 erneut verbindbare Verbindung, die `MQCNO_RECONNECT_Q_MGR` angegeben wurde, und die Verbindung hat versucht, eine Verbindung zu einem anderen WS-Manager herzustellen.

#### **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 wiederverbindbarer Client ist in der Lage, die Verbindung automatisch nur *nach* der Verbindung wieder herzustellen. Das heißt, der `MQCONN`-Aufruf selbst wird nicht wiederholt, wenn er fehlschlägt. Wenn Sie beispielsweise den Rückkehrcode 2543 - `MQRC_STANDBY_Q_MGR` von `MQCONN` empfangen, 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 `MQPUT`) 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.

## **Warteschlangenmanager mit mehreren Instanzen**

Vereinfachtes Neustarten von IBM MQ MQI client-Anwendungen, nachdem ein Multi-Instanz-Warteschlangenmanager seine Standby-Instanz aktiviert hat, indem er die automatische Clientverbindungswiederholung verwendet.

Die Standby-Instanz eines Warteschlangenmanagers mit mehreren Instanzen befindet sich in der Regel an einer anderen Netzadresse für die aktive Instanz. Geben Sie die Netzadressen der Instanzen in der Definitionstabelle für die Clientverbindung an (`CCDT`). Geben Sie entweder eine Liste der Netzadressen für den Parameter **CONNNAME** an, oder definieren Sie mehrere Zeilen für den Warteschlangenmanager in der `CCDT`.

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 verhindern, dass ein Client die Verbindung zu einem anderen WS-Manager erneut herstellen kann. Legen Sie die Option `MQCNO` (`MQCNO_RECONNECT_Q_MGR`)

fest. Der IBM MQ MQI client schlägt fehl, wenn er die Verbindung zu einem anderen WS-Manager wiederherstellt. Wenn Sie die Option MQCNO festlegen, schließen MQCNO\_RECONNECT\_Q\_MGR keine anderen Warteschlangenmanager in derselben 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.

## WS-Manager-Gruppen

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 *name* des Warteschlangenmanagers, QMNAME , 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 .

## Gruppen mit gemeinsamer Warteschlange

 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.

 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) KAIN

Keepalive-Intervall. Legen Sie das Keepalive-Intervall auf dem Clientverbindungskanal fest.

 Beachten Sie, dass KAIN nur für z/OS gültig ist.

KAIN 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 KAIN 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.

## Netz- konnektivität

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.
- Das Beenden eines Warteschlangenmanagers mit dem Standardbefehl **endmqm** löst keine automatische Clientverbindungswiederholung aus. 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:
  - Warteschlangenmanager-Fehler
  - Beenden eines Warteschlangenmanagers und Angeben der Option **-r** (Verbindung wiederherstellen) 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. Weitere Informationen finden Sie in „[Wiederherstellung eines automatisch verbundenen Clients](#)“ auf Seite 523.
- Wenn Sie ein Kanalexitprogramm für die Verbindung geschrieben haben, empfängt der Exit diese zusätzlichen MQI-Aufrufe.
- Sie können einen Umverbindungseignishandler 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.

## 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 THE NEW EXTENT OF PAGE SET psid HAS 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 ACCESSING PAGE 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 Log RBA has reached rba. 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 Log RBA has reached rba. 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 Log RBA has reached rba. 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, THE LOG RBA RANGE MUST BE RESET. 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 - APPROACHING END OF THE LOG RBA RANGE OF 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 COPYn 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 PROTOKOLLDATEN**

- 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: ERROR ON ARCHIVE DATA SET, OFFLOAD CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING GENERATED**

- Überprüfen Sie das Jobprotokoll des Warteschlangenmanagers auf weitere Fehlermeldungen.
- 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 Fehlermeldungen.

### **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 ret-code 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 FAILED FOR PAGE SET psid. 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 PAGE SET psid HAS 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: Unavailable page sets can cause problems - take action to correct this situation**

- Führen Sie die Systemprogrammiereraktion aus.

**CSQQ008I: nn units of recovery are still in doubt in queue manager qqqq**

- 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 This message cannot be processed**

- Überprüfen Sie die CSQOUTX-Datei, um die Ursache für den CSQINPX-Fehler zu ermitteln.
- Einige Befehle werden möglicherweise nicht verarbeitet.

**CSQX035I: csect-name Connection to queue manager qmgr-name stopping or broken, 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, -Beendigungs-codes und -Ursachencodes dokumentiert.

**CSQX032I: csect-name Initialization command handler terminated**

- Überprüfen Sie den MQRC, um die Ursache des Fehlers zu ermitteln.
- Diese Codes sind in IBM MQ for z/OS-Nachrichten, -Beendigungs-codes und -Ursachencodes dokumentiert.

**CSQX048I: csect-name Unable to convert message for name, 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 stopped, 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 Cluster queue q-name definitions inconsistent**

- Mehrere Clusterwarteschlangen innerhalb des Clusters haben inkonsistente Werte. Untersuchen und beheben Sie die Unterschiede.

### **CSQX411I: csect-name Repository manager stopped**

- Wenn der Repository-Manager aufgrund eines Fehlers gestoppt wurde, überprüfen Sie das Jobprotokoll auf Nachrichten.

### **CSQX417I: csect-name Cluster-senders remain for removed queue manager qmgr-name**

- Führen Sie die Systemprogrammiereraktion aus.

### **CSQX418I: csect-name Only one repository for cluster cluster\_name**

- Zur Erhöhung der Hochverfügbarkeit sollten Cluster mit zwei vollständigen Repositories konfiguriert werden.

### **CSQX419I: csect-name No cluster-receivers for cluster cluster\_name**

- Führen Sie die Systemprogrammiereraktion aus.

### **CSQX420I: csect-name No repositories for cluster cluster\_name**

- Führen Sie die Systemprogrammiereraktion aus.

### **CSQX448E: csect-name Repository manager stopping because of errors. 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.

## **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. **V 9.0.4** Auf Linux-Systemen können Sie auch replizierte Datenwarteschlangenmanager (RDQMs) implementieren, die eine Quorum-basierte Gruppe verwenden, um eine hohe Verfügbarkeit bereitzustellen.

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 z. B. Platten und Netze, die miteinander verbunden und so konfiguriert sind, dass bei einem Ausfall ein High Availability Manager, wie z. B. HACMP (UNIX) oder MSCS (Windows), eine *Funktionsübernahme* ausführt. 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 437 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.

#### **V 9.0.4 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.

#### **V 9.0.5 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.

## **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](#).

## Linux UNIX HA-Cluster unter UNIX and Linux

Sie können IBM MQ mit einem Hochverfügbarkeitscluster auf UNIX 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.

Vor IBM WebSphere MQ 7.0.1 wurde SupportPac MC91 bereitgestellt, um die Konfiguration von HA-Clustern zu unterstützen. IBM WebSphere MQ 7.0.1 bietet mehr Kontrolle als frühere Versionen darüber, wo Warteschlangenmanager ihre Daten speichern. Dies vereinfacht die Konfiguration von Warteschlangenmanagern in einem HA-Cluster. Die meisten der mit SupportPac MC91 bereitgestellten Scripts sind nicht mehr erforderlich, und das SupportPac wird zurückgezogen.

Dieser Abschnitt enthält eine Einführung in „HA-Clusterkonfigurationen“ auf Seite 437, die Beziehung von HA-Clustern zu Warteschlangenmanagerclustern, „IBM MQ-Clients“ auf Seite 438 und „IBM MQ in einem HA-Cluster“ auf Seite 438 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 WebSphere MQ 7 führte Features in die Verarbeitung der Clientkanaldefinitionstabelle (CCDT) ein, die Sie bei der Verfügbarkeit von Verbindungen 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 Plat-

ten 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.

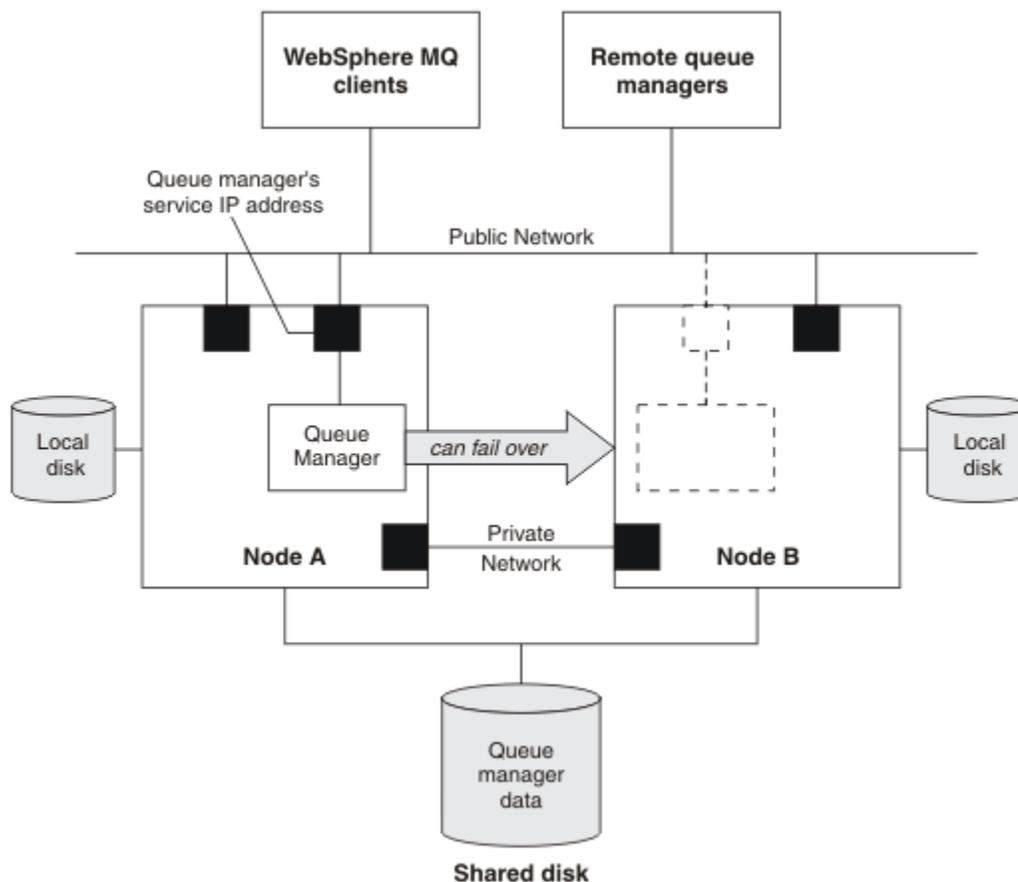


Abbildung 71. HV-Cluster

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 → UNIX **Gemeinsam genutzte Platten auf UNIX 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 Datei mqs.ini enthält eine Zeilengruppe für den Warteschlangenmanager QM1. Die Zeilengruppe 'Log' in der Datei qm.ini hat einen Wert für 'LogPath'.

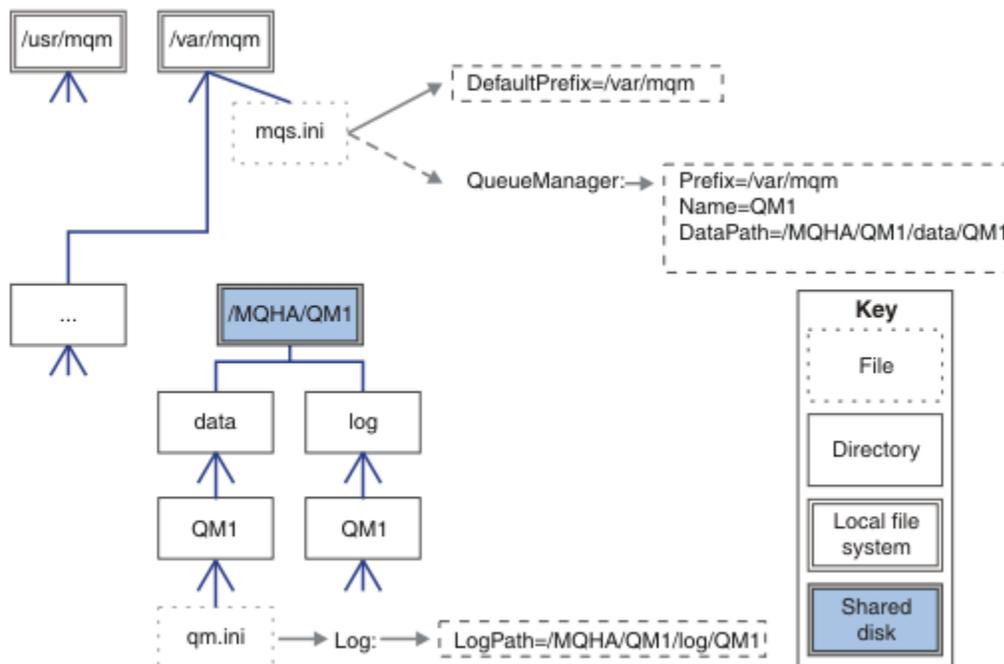


Abbildung 72. Gemeinsam genutzte data- und log-Verzeichnisse

### Vorgehensweise

1. Entscheiden Sie die Namen der Mountpunkte für die Dateisysteme des Warteschlangenmanagers. Beispiel: `/MQHA/qmgrname/data` für die Datendateien des Warteschlangenmanagers 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 qmqm-Benutzer muss Eigner der Mountpunkte sein.

Linux

UNIX

## **HA-Cluster-WS-Manager unter UNIX 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** .

Beispiel:

```
crtmqm -md /MQHA/qmqmname/data -ld /MQHA/qmqmname/log qmqmname
```

3. Starten Sie den Warteschlangenmanager manuell, indem Sie den Befehl **strmqm** verwenden.
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 Befehl qmqmname
```

Dabei steht qmqmname für den Namen des Warteschlangenmanagers.

Weitere Informationen zur Verwendung des Befehls **addmqinf** finden Sie im Abschnitt [„WS-Manager-Konfiguration zu anderen HA-Clusterknoten unter UNIX and Linux hinzufügen“](#) auf Seite 441.

Der Befehl **addmqinf** wird in ähnlicher Weise wie im folgenden Beispiel angezeigt:

```
addmqinf -sQueueManager -vName=qmqmname -vDirectory=qmqmname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmqmname/data/qmqmname
```

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 UNIX and Linux hinzufügen“](#) auf Seite 441 beschriebenen Schritte auszuführen.

Linux

UNIX

## **WS-Manager-Konfiguration zu anderen HA-Clusterknoten unter UNIX 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 UNIX and Linux erstellen](#)“ auf Seite 441 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 UNIX and Linux erstellen](#)“ auf Seite 441 beschrieben.

## 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 direktes Bearbeiten von `/var/mqm/mqs.ini`.
- Durch Absetzen des Befehls **addmqinf**, der durch den Befehl **dspmqinf** in Schritt 6 in „[HA-Cluster-WS-Manager unter UNIX and Linux erstellen](#)“ auf Seite 441 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 dem Warteschlangenmanager, der dem Warteschlangenmanager derzeit zugeordnet ist, müssen Sie die Installation, die dem Warteschlangenmanager zugeordnet ist, mit dem Befehl **setmqm** festlegen. Weitere Informationen finden Sie in [setmqm](#).

4. Entpacken Sie die WS-Manager-Dateisysteme.

Linux

UNIX

### **HA-Cluster-WS-Manager unter UNIX and Linux starten**

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 amqzma0 amqfcxba amqfpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqslr amqcrsta amqirmfa amqimppa \
               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 → UNIX **HA-Cluster-WS-Managers unter UNIX and Linux stoppen**

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 für das sofortige Stoppen, ohne dass Annahmen über den aktuellen Status des Warteschlangenmanagers gemacht werden. 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 „HA-Cluster-WS-Manager unter UNIX and Linux starten“ auf Seite 442 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 amqzma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
amqzlsa0 runmqchi runmqslr amqcrsta amqirmfa amqimppa \
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 *.*$"
cnt=`ps -ef | tr "\t" " " | grep amqzma0 | grep -v grep | \
egrep "$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 amqzma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
amqzlsa0 runmqchi runmqslr amqcrsta amqirmfa amqimppa \
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

UNIX

## HA-Cluster-Warteschlangenmanager unter UNIX 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.

Ab IBM WebSphere MQ 7.1 ist es möglich, dass mehrere Installationen von IBM MQ auf einem System koexistieren. Weitere Informationen zu mehreren Installationen finden Sie unter [Mehrere Installationen](#). Wenn Sie beabsichtigen, das Überwachungsscript über mehrere Installationen hinweg zu verwenden, einschließlich Installationen unter IBM WebSphere MQ 7.1 oder höher, müssen Sie möglicherweise einige zusätzliche Schritte ausführen. Wenn Sie über eine primäre Installation verfügen oder wenn Sie das Script mit früheren Versionen als IBM WebSphere MQ 7.1 verwenden, müssen Sie den `MQ_INSTALLATION_PATH` nicht angeben, um das Script zu verwenden. Andernfalls kann mit den folgenden Schritten sichergestellt werden, dass der `MQ_INSTALLATION_PATH` korrekt angegeben wird:

1. Verwenden Sie den Befehl `crtmqenv` von einer IBM WebSphere MQ 7.1-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 ist. In der Regel behandeln HA-Cluster ein blocktes 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 „HA-Cluster-WS-Manager unter UNIX and Linux starten“ auf Seite 442 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 < 7.1.0.0
    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
```

## **Warteschlangenmanager unter UNIX 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 → UNIX **HA-Cluster-Warteschlangenmanager unter UNIX 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 Befehl `rmvmqinf`, um die Konfigurationsinformationen des Warteschlangenmanagers zu entfernen:  
`rmvmqinf qmgrname`
3. Optional: Verwenden Sie den Befehl `dltmqm`, um den WS-Manager vollständig zu löschen.

**Wichtig:** Beachten Sie, dass das Löschen des Warteschlangenmanagers mit dem Befehl `dltmqm` die Daten- und Protokolldateien des Warteschlangenmanagers vollständig löscht.

Wenn Sie den WS-Manager gelöscht haben, können Sie den Befehl `rmvmqinf` verwenden, um die übrigen Konfigurationsdaten von den anderen Knoten zu entfernen.

## Windows **Unterstützung des Microsoft-Cluster-Service (MSCS)**

Einführung und Konfiguration von MSCS für die Unterstützung der Funktionsübernahme von virtuellen Servern.

Diese Informationen beziehen sich nur auf IBM MQ for Windows.

Mit dem Microsoft-Cluster-Service (MSCS) können Sie Server zu einem *Cluster* verbinden, wodurch eine höhere Verfügbarkeit der Daten und Anwendungen und eine einfachere Verwaltung des Systems möglich werden. MSCS kann Server- oder Anwendungsfehler automatisch erkennen und wiederherstellen.

MSCS unterstützt die *Funktionsübernahme von virtuellen Servern*, die Anwendungen, Websites, Druckwarteschlangen oder Dateifreigaben entsprechen (z. B. ihre Plattenspindeln, Dateien und IP-Adressen).

*Failover* ist der Prozess, mit dem MSCS einen Fehler in einer Anwendung auf einem Computer im Cluster feststellt und die unterbrechungsgesteuerte Anwendung ordnungsgemäß herunterfährt, die Statusdaten an den anderen Computer überträgt und die Anwendung dort erneut initialisiert.

In diesem Abschnitt werden MSCS-Cluster eingeführt und die Konfiguration der MSCS-Unterstützung in den folgenden Abschnitten beschrieben:

- „Einführung in MSCS-Cluster“ auf Seite 448
- „IBM MQ für MSCS-Clustering einrichten“ auf Seite 449

In den folgenden Abschnitten erfahren Sie außerdem, wie Sie IBM MQ für MSCS-Clustering konfigurieren:

- „Warteschlangenmanager für die Verwendung mit MSCS erstellen“ auf Seite 451
- „Warteschlangenmanager in MSCS-Speicher versetzen“ auf Seite 452
- „WS-Manager unter MSCS-Steuerung einschalten“ auf Seite 454
- „Entfernen eines Warteschlangenmanagers aus MSCS-Steuerung“ auf Seite 461

In den folgenden Abschnitten erhalten Sie darüber hinaus hilfreiche Hinweise zur Verwendung von MSCS mit IBM MQ und Erläuterungen zu den IBM MQ-Dienstprogrammen für die MSCS-Unterstützung:

- „Hinweise und Tipps zur Verwendung von MSCS“ auf Seite 462
- „Unterstützung für MSCS-Dienstprogramme“ auf Seite 465

### **Windows Einführung in MSCS-Cluster**

MSCS-Cluster sind Gruppen aus 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 ihre Operation dort erneut einleitet.

„Hochverfügbarkeitskonfigurationen“ auf Seite 435 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 73 auf Seite 449 zeigt eine typische Konfiguration für einen Cluster mit zwei Computern.

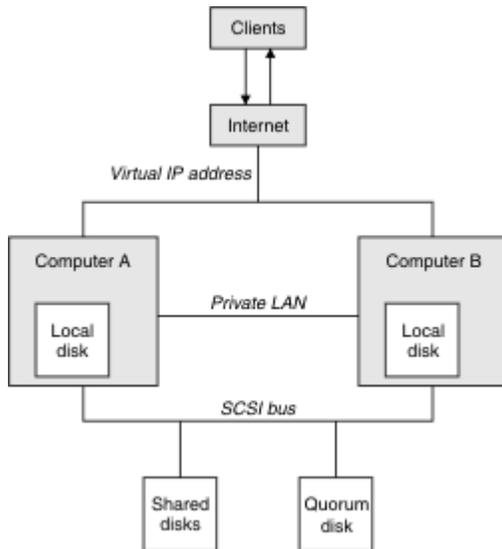


Abbildung 73. MSCS-Cluster mit zwei Computern

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.

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 MQ bereitgestellt wird. Weitere Informationen hierzu finden Sie im Abschnitt [„Warteschlangenmanager in MSCS-Speicher versetzen“](#) auf Seite 452. 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 451.

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 454.

Wenn Sie einen Warteschlangenmanager aus der MSCS-Steuerung entfernen möchten, verwenden Sie das in „Entfernen eines Warteschlangenmanagers aus MSCS-Steuerung“ auf Seite 461 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, Support-Pacs, Pfade und Exits identisch ist und dass ein allgemeiner Namensbereich (Sicherheitsumgebung) wie in „MSCS-Sicherheit“ auf Seite 450 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

Mit dieser Prozedur wird sichergestellt, dass ein neuer Warteschlangenmanager so erstellt wird, dass er für die Vorbereitung und Platzierung unter MSCS-Steuerung geeignet ist.

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 452 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 452 fort.

### **Warteschlangenmanager in MSCS-Speicher versetzen**

Mit dieser Prozedur wird ein vorhandener Warteschlangenmanager konfiguriert, um ihn für die MSCS-Steuerung geeignet zu machen.

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\key` gesetzt, 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 Warteschlangenma-

nagerattribut SSLKEYR mit IBM MQ Explorer oder mit dem MQSC-Befehl ALTER QMGRändern, um auf die neue TLS-Schlüsselrepositorydatei zu verweisen.

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 454 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 464 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. Beispiel:
  - 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. Setzen Sie in der IBM MQ Explorer das Startfeld in der Eigenschaftsanzeige des Warteschlangenmanagers auf `manual`.
10. Fahren Sie mit dem Abschnitt „[WS-Manager unter MSCS-Steuerung einschalten](#)“ auf Seite 454 fort.

### **Windows** *WS-Manager unter MSCS-Steuerung einschalten*

Die Tasks, die zum Platzieren eines Warteschlangenmanagers unter MSCS-Steuerung gehören, einschließlich vorausgesetzter Tasks.

### **Bevor Sie einen Warteschlangenmanager unter die MSCS-Steuerung stellen**

Führen Sie die folgenden Tasks aus, bevor Sie einen Warteschlangenmanager unter die MSCS-Steuerung stellen:

1. Stellen Sie sicher, dass IBM MQ und der zugehörige MSCS-Support 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 449 beschrieben.
2. Verwenden Sie das Dienstprogramm **haredtyp**, um IBM MQ als MSCS-Ressourcentyp auf allen Clusterknoten zu registrieren. Weitere Informationen enthält der Abschnitt „[Unterstützung für MSCS-Dienstprogramme](#)“ auf Seite 465.
3. Wenn Sie den Warteschlangenmanager noch nicht erstellt haben, lesen Sie den Abschnitt „[Warteschlangenmanager für die Verwendung mit MSCS erstellen](#)“ auf Seite 451.
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 452 ausgeführt haben.
5. Stoppen Sie den Warteschlangenmanager, falls er aktiv ist. Verwenden Sie dazu entweder eine Eingabeaufforderung oder den IBM MQ-Explorer.
6. Testen Sie die MSCS-Operation der gemeinsam genutzten Laufwerke, bevor Sie zu einer der folgenden Windows-Prozeduren in diesem Thema gehen.

### **Windows Server 2012**

Gehen Sie wie folgt vor, um einen Warteschlangenmanager unter MSCS-Steuerung auf Windows Server 2012 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 WS-Manager wie folgt unter MSCS 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 MSCS das Routing dieser Adresse handhabt, prüft es **nicht**, ob die Adresse erreicht werden kann.

- d. Ordnen 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 464.
- 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 464.

**Anmerkung:** Das Abfrageintervall für *LooksAlive* wird auf den Standardwert 5000 ms gesetzt. Das Abfrageintervall für *isAlive* wird auf den Standardwert 60000 ms gesetzt. Diese Standard-

werte können nur geändert werden, nachdem die Ressourcendefinition abgeschlossen wurde. Weitere Informationen finden Sie im Abschnitt „looksAlive und isAlive-Polling für MSCS“ auf Seite 460.

- d. Legen Sie optional einen bevorzugten Knoten fest (aber beachten Sie die Kommentare in „Bevorzugte Knoten in MSCS verwenden“ auf Seite 465).
  - 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. Beispiel:
- 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 452 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. Ordnen 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 464.
- `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 464.

**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 460.

- d. Legen Sie optional einen bevorzugten Knoten fest (aber beachten Sie die Kommentare in „[Bevorzugte Knoten in MSCS verwenden](#)“ auf Seite 465).
  - 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. Beispiel:
- 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 452 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 Workstati-

on 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 450 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 464.
  - `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 464.
- 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 460.
8. Legen Sie optional einen bevorzugten Knoten fest (aber beachten Sie die Kommentare in „[Bevorzugte Knoten in MSCS verwenden](#)“ auf Seite 465).
  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. Beispiel:
    - 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 452 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.

### **looksAlive und isAlive-Polling für MSCS**

`looksAlive` und `isAlive` sind Intervalle, in denen MSCS-Aufrufe zurück in die vom bereitgestellten Bibliothekscode gelieferten Ressourcentypen zurückrufen, und fordert, dass die Ressource prüft, ob der Arbeitsstatus von sich selbst bestimmt wird. Dies bestimmt schließlich, ob MSCS über die Ressource fehlschlagen soll.

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 Ressourcentyp-überprüfung 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 die Warteschlangenmanagerressource nicht aktiv sein soll, wird MSCS auf der Basis anderer erweiterter MSCS-Optionen für die Ressource und die zugehörigen abhängigen Ressourcen zu einem anderen Knoten im Cluster ausgelöst. Weitere Informationen finden Sie in der [MSCS-Dokumentation](#) .

## **Windows Entfernen eines Warteschlangenmanagers aus MSCS-Steuerung**

Sie können Warteschlangenmanager aus der MSCS-Steuerung entfernen und sie zur manuellen Verwaltung zurückgeben.

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-Schlüsselrepositoriedatei 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 461).
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 461.
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\mqmgs\QMname haben. Versuchen Sie nicht, die Dateien manuell zu verschie-

ben; 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 [„Warteschlangenmanagerprotokoll-dateien“](#) auf Seite 464 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 Laufwerk-buchstaben 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 452 beschrieben).

### **Windows** *Hinweise und Tipps zur Verwendung von MSCS*

Dieser Abschnitt enthält einige allgemeine Informationen, die Ihnen bei der effektiven Verwendung von IBM MQ-Unterstützung für MSCS helfen.

Dieser Abschnitt enthält einige allgemeine Informationen, die Ihnen bei der effektiven Verwendung von IBM MQ-Unterstützung für MSCS helfen.

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 451 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 im Abschnitt [„Warteschlangenmanager in MSCS-Speicher versetzen“](#) auf Seite 452.

## **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 452 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 465 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 73](#) auf Seite 449) 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 451
- [„Warteschlangenmanager in MSCS-Speicher versetzen“](#) auf Seite 452
- [„WS-Manager unter MSCS-Steuerung einschalten“](#) auf Seite 454

### **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 452).

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 schwerwiegender 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 MSCSan. 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\System64-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 Requesterkannäle 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 IBM MQ-Unterstützung für MSCS-Dienstprogramme, die Sie an einer Eingabeaufforderung ausführen können.

IBM MQ-Unterstützung für MSCS umfasst die folgenden Dienstprogrammprogramme:

##### **Ressourcentyp registrieren/deregistrieren**

haregtyp.exe

Nachdem Sie den IBM MQ-MSCS-Ressourcentyp *deregistriert* 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 461 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 in „[Warteschlangenmanager in MSCS-Speicher versetzen](#)“ auf Seite 452.

### **WS-Manager aus einem Knoten löschen**

```
hadl1tmqm.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:

```
hadl1tmqm /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 74 auf Seite 467 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.

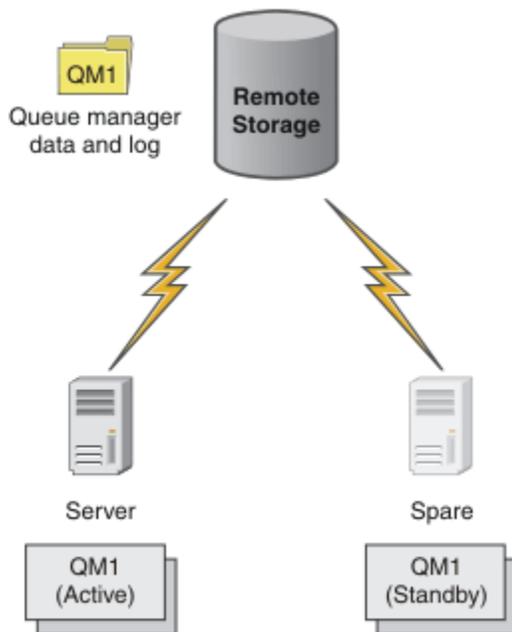


Abbildung 74. Multi-Instanz-Warteschlangenmanager

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-/Schreibsperrern 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 genutztes 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 jeder Nicht-z/OS-Plattform erstellen, die von IBM WebSphere MQ 7.0.1 und höher unterstützt wird.

Die automatische Wiederherstellung von Clientverbindungen wird für MQI-Clients ab IBM WebSphere MQ 7.0.1 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 UNIX 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 **dspmqlinf** auf dem ersten Server aus, um die Konfigurationsdaten des Warteschlangenmanagers zu erfassen und ihn in die Zwischenablage zu kopieren.

4. Führen Sie den Befehl **addmqinf** mit den kopierten Daten aus, um die WS-Manager-Konfiguration auf dem zweiten Server zu erstellen.

Sie führen **crtmqm** nicht aus, um den WS-Manager auf dem zweiten Server erneut 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 UNIX 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 499. 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 472 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 (" domain local ") hat. Sichern Sie die Dateifreigabe mit der lokalen `mqm`-Domäne und führen Sie WS-Manager-Prozesse für alle Instanzen eines Warteschlangenmanagers in derselben Domäne aus, die sich in derselben Domäne wie die lokale `mqm`-Gruppe enthält. Die Task [„WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen“](#) auf Seite 488 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 WS-Manager-Instanzen. Alternativ können Sie die Dateien `mq5.ini` und `qm.ini` direkt bearbeiten. Die Themen [„Mehrinstanz-Warteschlangenmanager unter Linux erstellen“](#) auf Seite 512, [„WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen“](#) auf Seite 472 und [„WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen“](#) auf Seite 488 sind Beispiele für die Konfiguration eines Multi-Instanz-Warteschlangenmanagers.

Auf Windows-, UNIX and Linux-Systemen können Sie eine einzelne `mq5.ini`-Datei gemeinsam nutzen, indem Sie sie in die Netzfrequenz stellen und die Umgebungsvariable **AMQ\_MQS\_INI\_LOCATION** so festlegen, dass sie darauf 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 UNIX 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 UNIX-Systeme erstellen, solange die Freigaben unterschiedlich sind. Sie können verschiedene Eigentümern verschiedene Freigaben geben. Die Einschränkung ist eine Folge der verschiedenen Namen, die für die IBM MQ-Benutzer und -Gruppen zwischen UNIX und IBM i verwendet werden. Die Tatsache, dass der Benutzer und die Gruppe denselben uid und gid haben können, lockt die Einschränkung nicht ab.

5. Auf UNIX 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.

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 502 und „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern“ auf Seite 499, 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 472.

## **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. Vor IBM WebSphere MQ 7.1 wurden Multi-Instanz-Warteschlangenmanager nur auf Domänencontrollern ausgeführt (siehe Abschnitt „Multi-Instanz-WS-Manager auf Domänencontrollern ausführen“ auf Seite 471). 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.

Die Task zum Erstellen eines Multi-Instanz-Warteschlangenmanagers ist identisch mit der Task in IBM WebSphere MQ 7.0.1, bis auf eine Änderung. Sie müssen den zusätzlichen Sicherheitsgruppennamen zu den Parametern des Befehls **crtmqm** hinzufügen. Die Task wird im Abschnitt „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen“ auf Seite 472 beschrieben.

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 502. 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 504 und „Lesen und Schreiben von Daten- und Protokolldateien, die von einer alternativen lokalen Sicherheitsgruppe autorisiert sind“ auf Seite 508. In einem weiteren Abschnitt, „Gemeinsam genutzte WS-Manager-Daten und Protokollverzeichnisse und -dateien in Windows sichern“ auf Seite 499, 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 475.
2. „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren“ auf Seite 479.
3. „Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows“ auf Seite 482.
4. „Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind“ auf Seite 485.

Anschließend können Sie die Task „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen“ auf Seite 472 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**

In IBM WebSphere MQ 7.0.1 wurden Multi-Instanz-Warteschlangenmanager nur auf Domänencontrollern ausgeführt. 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 499 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 in-

stallieren, werden die WS-Manager-Daten und -Protokolldateien mit der Domänengruppe *mqm* gesichert, die gemeinsam genutzt werden kann. Führen Sie die Schritte in der Task „[WS-Manager mit mehreren Instanzen auf Windows-Domänencontrollern erstellen](#)“ auf Seite 488 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-WS-Managers 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-WS-Managers 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 WS-Manager-Daten und Protokolldateien gespeichert werden sollen, nicht überprüft werden. Die Prüfprozedur [Verhalten des gemeinsam genutzten Dateisystems überprüfen](#) ist auf UNIX 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 Server 2008 zu testen. Sie können die Schritte an Ihre Domäne anpassen.

1. [„Active Directory- und DNS-Domäne unter Windows erstellen“](#) auf Seite 475.
2. [„IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren“](#) auf Seite 479.
3. [„Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows“](#) auf Seite 482.
4. [„Lesen und Schreiben von gemeinsam genutzten Daten und Protokolldateien, die von einer alternativen globalen Sicherheitsgruppe autorisiert sind“](#) auf Seite 485.

## 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 weitere 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 479 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 *mq5.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* auf *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 485 aus, um sie zu erstellen.
  - b) Überprüfen Sie die Werte der Parameter *Prefix* und *InstallationName* für *venus*.

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 `Prefix` 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) Die Option *QMGR* auf *mars* wird gestoppt.

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 7.1.0.0.
```

## 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
```

Systemantwort auf *venus*:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended, permitting switchover to  
a standby instance.
```

Und auf *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 496.

### **Windows** Active Directory- und DNS-Domäne unter Windows erstellen

Mit dieser Task erstellen Sie die Domäne *wmq.example.com* auf einem Windows 2008-Domänencontroller namens *sun*. 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 Beispielfunktion 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-WS-Managers 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-WS-Managers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Es sind dieselben Schritte auszuführen wie bei einem Windows Server 2008, der installiert, jedoch nicht mit Rollen konfiguriert wird. Wenn Sie einen bereits vorhandenen Domänencontroller konfigu-

rieren möchten, ist es unter Umständen hilfreich, die durchzuführenden Schritte an einem neuen Windows Server 2008 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 `Doma.in\mqm` und geben Sie ihr die richtigen Berechtigungen.
5. Fügen Sie einen Benutzer hinzu und machen Sie ihn zu einem Mitglied der globalen Gruppe `Doma.in\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 470.

Für die Zwecke der Task ist der Hostname des Domänencontrollers `sun` und die beiden IBM MQ-Server heißen `mars` und `venus`. 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 auf **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**.
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.

**Tip:** 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 einen Warmstart durchgeführt hat, melden Sie sich als *wmq\Administrator* 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 den Aufruf der Gruppe *Domain mqm* ändert sich das Verhalten des Assistenten "Prepare IBM MQ" (WebSphere MQ vorbereiten) auf einer Workstation bzw. einem Server der Domäne. Der "IBM MQ-Vorbereitungsassistent" fügt die Gruppe *Domain mqm* automatisch der lokalen Gruppe *mqm* jeder neuen IBM MQ-Installation in der Domäne hinzu.
- Sie können Workstations oder Server in einer Domäne mit keiner globalen *Domain mqm*-Gruppe installieren. Wenn dies der Fall ist, müssen Sie eine Gruppe mit den gleichen Eigenschaften wie die Gruppe *Domain mqm* definieren. Machen Sie die Gruppe oder ihre Benutzermitglieder zu Mitgliedern der lokalen Gruppe *mqm*, sobald IBM MQ in einer Domäne installiert wird. 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 angeben, muss Mitglied der Gruppe *Domain mqm* oder einer alternativen Gruppe sein, die Sie mit denselben Eigenschaften wie 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 "Assistent für die Vorbereitung von IBM MQ " konfiguriert Domain mqm automatisch als Mitglied der lokalen Gruppe mqm, wenn er 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 479, 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-WS-Managers 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-WS-Managers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Führen Sie die im Abschnitt „[Active Directory- und DNS-Domäne unter Windows erstellen](#)“ auf Seite 475 beschriebenen Schritte zum Erstellen eines Domänencontrollers *sun* für die Domäne *wmq.example.com* aus. 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

Bei dieser Task konfigurieren Sie einen Windows Server 2008 mit der Bezeichnung *mars* als Mitglied der Domäne *wmq.example.com*. Sie installieren IBM MQ und konfigurieren die Installation so, dass sie als Mitglied der Domäne *wmq.example.com* 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 470.

Für die Zwecke der Task ist der Hostname des Domänencontrollers *sun* und die beiden IBM MQ-Server heißen *mars* und *venus*. 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 **Starten**.
  - 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 Version 6** in der Liste der Einträge im Fenster "Eigenschaften der LAN-Verbindung" 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 auf *mars* den Computer zur *wmq.example.com*-Domäne hinzu.
  - a) Klicken Sie auf **Start**.
  - b) Klicken Sie mit der rechten Maustaste auf **Computer > Eigenschaften**. Klicken Sie im Bereich "Computernamen", "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 > Restart Now** (Jetzt erneut starten) als Antwort auf die Nachricht "Welcome to the *wmq.example.com* domain" (Willkommen bei der Domäne "*wmq.example.com*").
3. Überprüfen Sie, ob der Computer Mitglied der *wmq.example.com*-Domäne ist.
  - a) Melden Sie sich unter *sun* am Domänencontroller als *wmq\Administrator* an.
  - b) Öffnen Sie **Server Manager > Active Directory Domain Services > wmq.example.com > Computers** und prüfen Sie, ob *mars* im Fenster "Computers" ordnungsgemäß aufgeführt ist.
4. Installieren Sie IBM MQ for Windows auf *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 (*mars\Administrator*) an.
- b) Führen Sie den Befehl **Setup** auf 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 **Netzwerkconfiguration** > **Ja**, um eine Domänenbenutzer-ID zu konfigurieren.  
Bei der Task „Active Directory- und DNS-Domäne unter Windows erstellen“ auf Seite 475 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 ist installiert und das Installationsprogramm startet den "Assistenten für die Vorbereitung von IBM MQ".

**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 **Security Settings** > **Local Policies** > **User Rights Assignments** (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 den Assistenten "IBM MQ vorbereiten" aus.

Weitere Informationen zur Ausführung des "Assistenten für die Vorbereitung von IBM MQ " finden Sie im Abschnitt IBM MQ mit dem IBM MQ-Vorbereitungsassistenten konfigurieren.

- a) Das IBM MQ-Installationsprogramm führt den "Assistenten für die Vorbereitung von IBM MQ" automatisch aus.  
Wenn Sie den Assistenten manuell starten möchten, suchen Sie nach dem Direktaufruf für den "Assistenten für die Vorbereitung von IBM MQ " 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 wird für Windows-Domänenbenutzer konfiguriert" 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 504 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 482 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 Informationen

[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-WS-Managers 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-WS-Managers mit dem Namen *QMGR*.

Ersetzen Sie die kursiv dargestellten Namen im Beispiel durch die Namen Ihrer Wahl.

## Vorbereitende Schritte

1. Wenn Sie diese Task genau wie dokumentiert ausführen möchten, führen Sie die Schritte in der Task „Active Directory- und DNS-Domäne unter Windows erstellen“ auf Seite 475 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 470.

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 Freigabe 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. Sie muss Mitglied der lokalen Gruppe *mqm* auf einem IBM MQ -Server und der globalen Gruppe *wmqha* sein.

Wenn der Warteschlangenmanager in dieser Gruppe von Tasks als Service ausgeführt wird, wird er unter der Benutzer-ID *wmquser1* ausgeführt, sodass *wmquser1* ein Mitglied von *wmqha* sein muss. Wenn der Warteschlangenmanager interaktiv ausgeführt wird, wird er unter der Benutzer-ID *wmquser2* ausgeführt, sodass *wmquser2* ein Mitglied von *wmqha* sein muss. 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 am 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* 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. Autorisieren Sie die globale Gruppe *wmqha* , um die vollständige Steuerungsberechtigung 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 **Vererbte Berechtigungen für diesen Objekteigner einschließen** ab. Klicken Sie im Fenster "Windows Security" 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 "Permission Entry for wmq " die Option **Full Control** in der Liste **Permissions** 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 Freiteilnamen als *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 *marsh* hinzufügen.
  - a. Melden Sie sich als *wmq\Administrator* bei *mars* 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\Administratorein* 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 durch Öffnen der *wmq* -Freigabe in *sun.wmq.example.com* und listet die Daten auf und protokolliert Verzeichnisse.
    - b. Überprüfen Sie die Berechtigungen von *wmquser1*, indem Sie eine Datei im Datenunterverzeichnis erstellen, einen Inhalt 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 485).

### 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](#)

In dieser Übung wird gezeigt, wie die Markierung `-a` auf dem Befehl `crtmqm` 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-WS-Managers 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-WS-Managers 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 Server 2008 zu testen. Sie können die Schritte an Ihre Domäne anpassen.

1. „Active Directory- und DNS-Domäne unter Windows erstellen“ auf Seite 475.

2. „IBM MQ auf einem Server oder einer Workstation in einer Windows-Domäne installieren“ auf Seite 479.
3. „Erstellen eines gemeinsam genutzten Verzeichnisses für WS-Manager-Daten und -Protokolldateien unter Windows“ auf Seite 482.

## 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 470.

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 Sicherheitsberechtigungenachweise 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, 2023. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

Und:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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.  
Das Verzeichnis '\\sun\wmq\data\QMGR' wurde erstellt.  
Der WS-Manager ist der Installation '1' zugeordnet.  
Standardobjekte für WS-Manager 'QMGR' erstellen oder ersetzen
```

Statistik für Standardobjekte: 74 erstellt. 0 ersetzt.  
Completing setup.  
Setup beendet.

## 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 7.1.0.0.
```

2. Erstellen Sie eine Testwarteschlange.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Systemantwort:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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 **amqsputein**.

```
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 **amqsgetab**.

```
amqsget QTEST QMGR
```

Systemantwort:

```
Sample AMQSGET0 start
```

```
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGE0 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 die Befehle aufgefordert werden, die Datei oder das Verzeichnis zu löschen.

```
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 -Protokoll-dateien 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 472.

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-WS-Managers 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-WS-Managers 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 WS-Manager-Daten und Protokolldateien gespeichert werden sollen, nicht überprüft werden. Die Prüfprozedur Verhalten des gemeinsam genutzten Dateisystems überprüfen ist auf UNIX 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 475 aus, um den ersten Domänencontroller zu erstellen.
3. Führen Sie die Schritte in „Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne“ auf Seite 492 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 494 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 dem Beispiel 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 **Vererbte Berechtigungen für diesen Objekteigner einschließen** ab. Klicken Sie im Fenster "Windows Security" 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 *mqmein*. Klicken Sie auf **Namen überprüfen**.
  - f) Geben Sie als Antwort auf ein Windows-Sicherheitsfenster den Namen und das Kennwort für den Domain Administrator ein und klicken Sie auf **OK > OK**.
  - g) Wählen Sie im Fenster "Permission Entry for wmq " die Option **Full Control** in der Liste **Permissions** 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 Freiteilnamen als *wmq*.
  - l) Klicken Sie auf **Berechtigungen > Hinzufügen ...**, und geben Sie den Namen der lokalen Domänengruppe *mqm* ; Domain Administratorsein. Klicken Sie auf **Namen überprüfen** .
  - m) Geben Sie als Antwort auf ein Windows-Sicherheitsfenster den Namen und das Kennwort für den Domain Administrator ein und klicken Sie auf **OK > OK**.
3. Erstellen Sie den WS-Manager *QMGR* auf dem ersten Domänencontroller, *sun*.

```
cirtmqm -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*, so dass eine Standby-Instanz möglich ist.

```
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 7.1.0.0.
```

5. Erstellen Sie eine zweite Instanz von *QMGR* auf *earth*.

- a) Überprüfen Sie die Werte der Parameter Prefix und InstallationName für *earth*.

Führen Sie unter *sunden* Befehl **dspmqlinf** aus:

```
dspmqlinf 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 **dspmqlinf** erneut mit dem Parameter `-o command` aus.

```
dspmqlinf -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 `Prefix` 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 auf *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 WS-Manager von *sun* in *earth* umschaltet:

1. Führen Sie unter *sun* den folgenden Befehl aus:

```
endmqm -i -r -s QMGR
```

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 unter *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 496.

### Zugehörige Tasks

[„Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne“](#) auf Seite 492

[„IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren“](#) auf Seite 494

### Zugehörige Informationen

[Clusterknoten von Windows 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 WS-Manager mit mehreren Instanzen 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-WS-Managers 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-WS-Managers 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 im Abschnitt [„Active Directory- und DNS-Domäne unter Windows erstellen“](#) auf Seite 475 beschriebenen Schritte zum Erstellen eines Domänencontrollers *sun* für die Domäne *wmq.example.com* aus. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.
2. Installieren Sie Windows Server 2008 auf einem Server in der Standardworkgroup (WORKGROUP). Für das Beispiel hat der Server den Namen *earth*.

## Informationen zu diesem Vorgang

In dieser Task konfigurieren Sie einen Windows Server 2008 mit dem Namen *earth* als zweiter 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 470.

## 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 **Starten**.
  - 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 Version 6** in der Liste der Einträge im Fenster "Eigenschaften der LAN-Verbindung" 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 auf **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* einen Warmstart durchgeführt hat, 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 494.

### Zugehörige Tasks

**Windows** [IBM MQ auf Windows-Domänencontrollern in einer Beispieldomäne installieren](#)  
[„Active Directory- und DNS-Domäne unter Windows erstellen“](#) auf Seite 475

**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.

Geben Sie hier Ihre Kurzbeschreibung an; sie wird für den ersten Absatz verwendet und ist abstrakt.

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-WS-Managers 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-WS-Managers 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 im Abschnitt [„Active Directory- und DNS-Domäne unter Windows erstellen“](#) auf Seite 475 beschriebenen Schritte zum Erstellen eines Domänencontrollers *sun* für die Domäne *wmq.example.com* aus. Ändern Sie die in Kursivdruck angezeigten Namen Ihrer Konfiguration entsprechend.
2. Führen Sie die Schritte im Abschnitt [„Hinzufügen eines zweiten Windows-Domänencontrollers zu einer Beispieldomäne“](#) auf Seite 492 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 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 *sun* und *earth* als Domänenadministrator (*umq\Administrator*) an.
- b) Führen Sie den Befehl **Setup** auf den IBM MQ for Windows -Installationsmedien aus.

Die Launchpandanwendung von IBM MQ wird gestartet.

- c) Klicken Sie auf **Softwarevoraussetzungen**, um zu überprüfen, ob die vorausgesetzte Software installiert ist.
- d) Klicken Sie auf **Netzwerkconfiguration > 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 ist installiert und das Installationsprogramm startet den "Assistenten für die Vorbereitung von IBM MQ".

Bei der IBM MQ for Windows-Installation werden eine lokale Gruppe *mqm* und eine Domänengruppe *Domain mqm* konfiguriert. *Domain mqm* wird zu einem Mitglied von *mqm* gemacht. Subsequente Domänencontroller in derselben Domäne nutzen die Gruppen *mqm* und *Domain mqm* gemeinsam.

2. Führen Sie sowohl auf *earth* als auch auf *sun* den "Assistenten für die Vorbereitung von IBM MQ" aus.

Weitere Informationen zur Ausführung des "Assistenten für die Vorbereitung von IBM MQ" finden Sie im Abschnitt [IBM MQ mit dem IBM MQ-Vorbereitungsassistenten konfigurieren](#).

- a) Das IBM MQ-Installationsprogramm führt den "Assistenten für die Vorbereitung von IBM MQ" automatisch aus.

Wenn Sie den Assistenten manuell starten möchten, suchen Sie nach dem Direktaufruf für den "Assistenten für die Vorbereitung von IBM MQ" 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.<sup>1</sup>
- c) Wählen Sie auf der letzten Seite des Assistenten die Markierungsfelder aus, die Sie benötigen, und klicken Sie auf **Fertig stellen**.

Der "Assistent für die Vorbereitung von IBM MQ" erstellt einen lokalen Domänenbenutzer *MUSR\_MQADMIN* auf dem ersten Domänencontroller und einen weiteren 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.

---

<sup>1</sup> 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.

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 der Domäne 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. Vergewissern Sie sich, dass der "Assistent für die Vorbereitung von IBM MQ " 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 mqmist
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**.
  - b. Öffnen Sie **Security Settings > Local Policies > User Rights Assignments** (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 504 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 488 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 Informationen

[Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service](#)

**Windows** [Multi-Instanz-Warteschlangenmanager unter Windows überprüfen](#)

Verwenden Sie die Beispielprogramme **amqsgshac**, **amqspshac** und **amqsmhac**, um eine Konfiguration mit mehreren Instanzen des Warteschlangenmanagers zu prüfen. Dieser Abschnitt enthält eine Beispielf Konfiguration 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 488 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, Option. 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. Der Ordner /MQHA, der die Clientverbindungstabelle enthält, wird gemeinsam genutzt, 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 Host des MQHA -Ordners handelt!

**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.

Sie haben möglicherweise einen der Server ausgewählt, um das MQHA -Verzeichnis oder das Dateisystem zu hosten. Wenn Sie die Übernahme testen möchten, indem Sie das VMware-Fenster mit dem aktiven Server schließen, stellen Sie sicher, dass es sich nicht um das Hosting von MQHA handelt!

#### 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. Stellen Sie sicher, dass der Listener aktiv ist und die Steuerung des Warteschlangenmanagers aufweist, 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 dem 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 in einer Eingabeaufforderung, damit der Ursachencode auf der Konsole ausgegeben wird, oder schauen Sie sich die Datei AMQERR01.LOG 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:\>amqspshac SOURCE QM1
Beispiel für AMQSPHAC-Start
Zielwarteschlange ist SOURCE
Nachricht Message 1
Nachricht Message 2
Nachricht Message 3
Nachricht Message 4
Nachricht Message 5
17:05:25: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:47: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:52: EVENT: Verbindung wurde wieder hergestellt
Nachricht Message 6
Nachricht Message 7
Nachricht Message 8
Nachricht Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Start Beispiel AMQSMHA0
17:05:25: EVENT: Connection Reconnecting (Delay: 97ms)
17:05:48: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgshac TARGET QM1
Beispiel für AMQSGHAC-Start
Nachricht Message 1
Nachricht Message 2
Nachricht Message 3
Nachricht Message 4
Nachricht Message 5
17:05:25: EVENT: Connection Reconnecting (Delay: 156ms)
17:05:47: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:52: EVENT: Verbindung wurde wieder hergestellt
Nachricht Message 6
Nachricht Message 7
Nachricht Message 8
Nachricht 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 WS-Manager auf einem

anderen Server als *mars* ausgeführt wird, kann er nicht auf die Daten und Protokolldateien zugreifen, die er bei der Ausführung unter *mars* erstellt hat. 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.

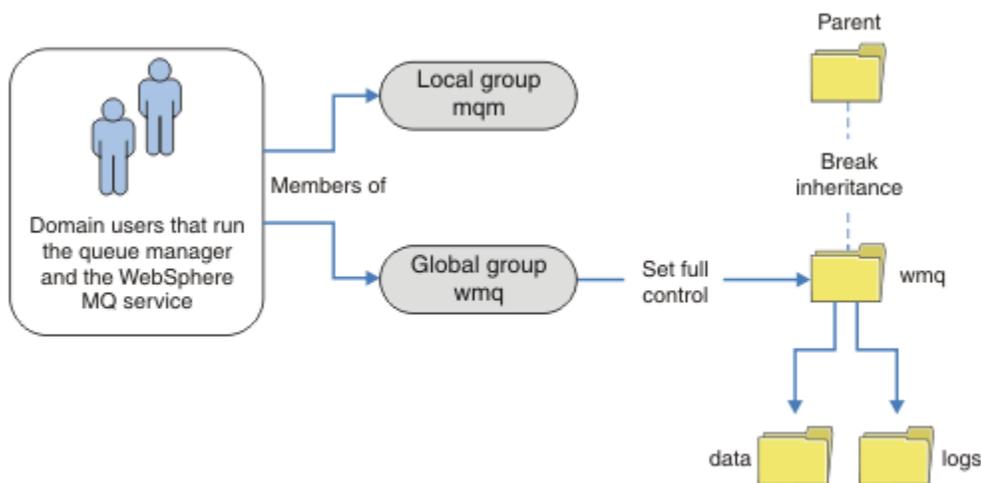
Die Bereitstellung des Warteschlangenmanagers mit einer globalen alternativen Sicherheitsgruppe löst das Problem (siehe [Abbildung 75](#) auf Seite 500). Sichern Sie einen fernen Ordner mit einer globalen Gruppe. Übergeben Sie den Namen der globalen Gruppe an den WS-Manager, wenn Sie ihn unter *mars* erstellen. Übergeben Sie den Namen der globalen Gruppe 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 75. 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 75](#) auf Seite 500 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 75](#) auf Seite 500 nicht praktisch. Sie müssen den Prozess zum Hinzufügen von Benutzern zu lokalen Gruppen für jeden IBM MQ-Server wiederholen. Erstellen Sie stattdessen eine globale `mqm`-Gruppe auf dem Domänencontroller und machen Sie die Benutzer, die IBM MQ ausführen zu Mitgliedern der `mqm`-Gruppe. Informationen hierzu finden Sie im Artikel [Abbildung 76](#) auf Seite 501. Wenn Sie IBM MQ als Domäneninstallation installieren, macht der Assistent "IBM MQ vorbereiten" automatisch die Gruppe

Domain mqm zu einer 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.

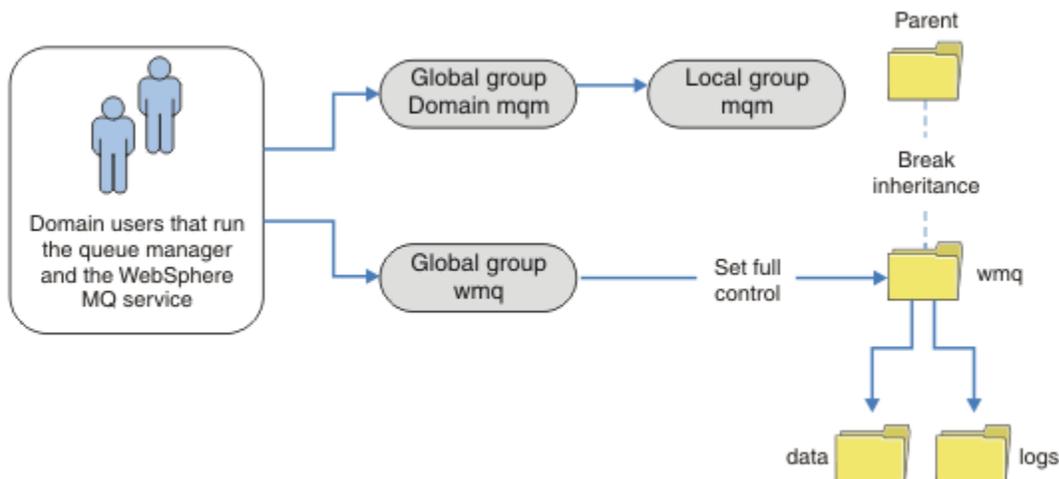


Abbildung 76. Sichern von Warteschlangenmanagerdaten und -protokollen mithilfe einer alternativen globalen Sicherheitsgruppe (2)

Die Organisation in [Abbildung 76](#) auf Seite 501 ist nach dem Stand der Dinge unnötig kompliziert. Die Anordnung hat zwei globale Gruppen mit identischen Mitgliedern. Sie können die Organisation vereinfachen und nur eine globale Gruppe definieren (siehe [Abbildung 77](#) auf Seite 501).

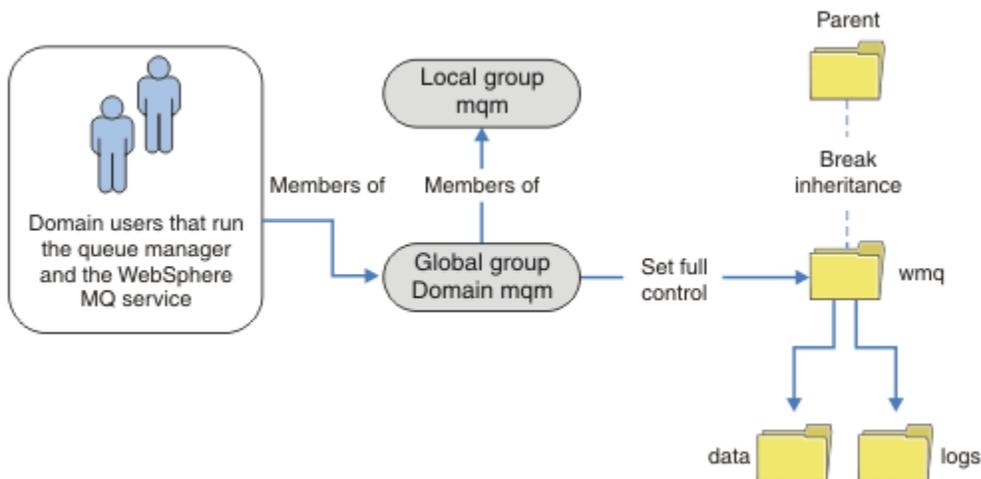


Abbildung 77. Sichern von WS-Manager-Daten und -Protokollen mithilfe einer alternativen globalen Sicherheitsgruppe (3)

Alternativ hierzu benötigen Sie möglicherweise eine feinere Zugriffssteuerung, wobei die verschiedenen Warteschlangenmanager auf verschiedene Ordner zugreifen können. Weitere Informationen finden Sie im Abschnitt [Abbildung 78](#) auf Seite 502. In [Abbildung 78](#) auf Seite 502 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 77](#) auf Seite 501 weitergegeben wird, da es zwei Benutzergruppen gibt.

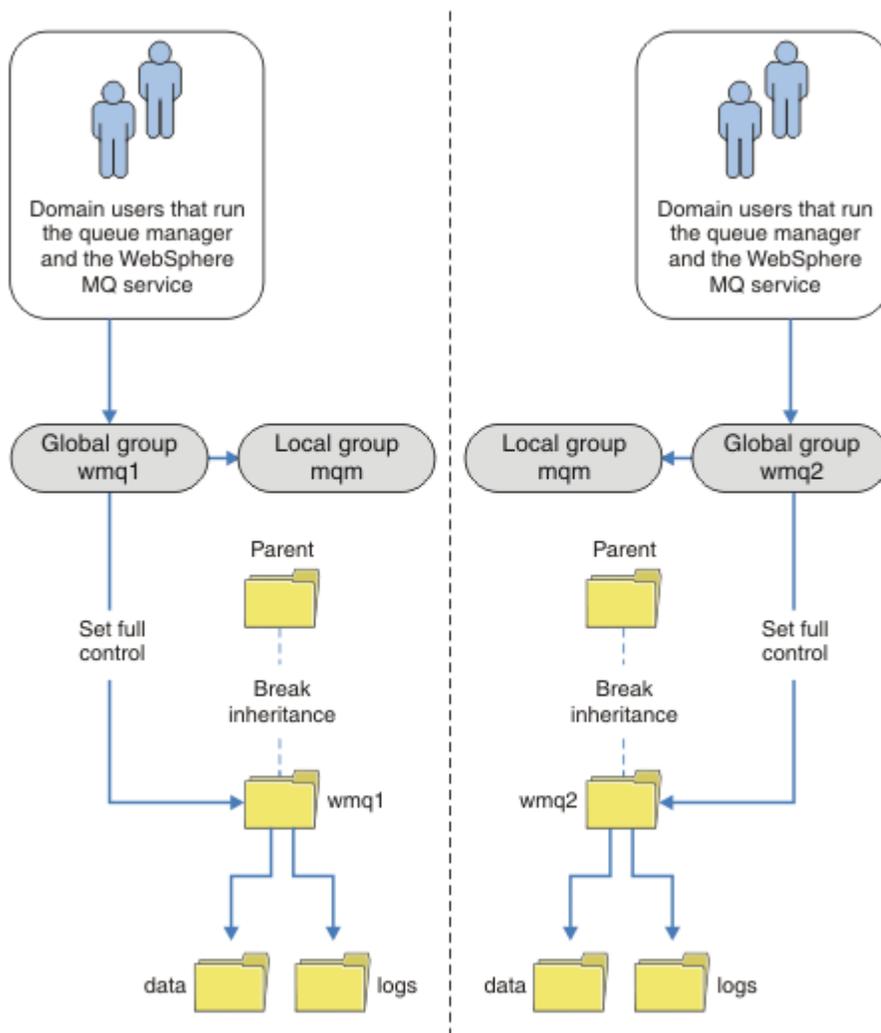


Abbildung 78. Sichern von Warteschlangenmanagerdaten und -protokollen mit einem alternativen globalen Sicherheitsanbieter (4)

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 77](#) auf Seite 501 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 504 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 508, 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 499).

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 504 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 79 auf Seite 504 beziehen. Die lokale Gruppe wmq ist ein Beispiel für eine alternative Sicherheitsgruppe.

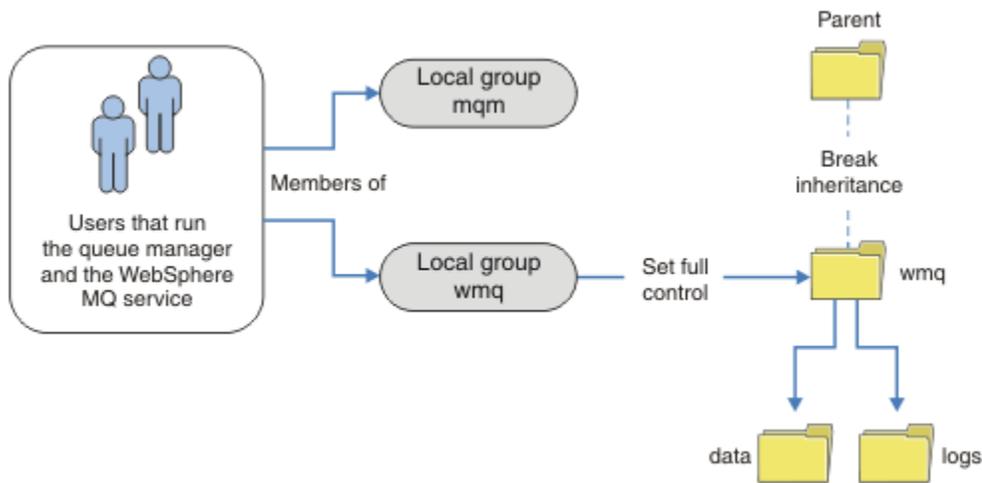


Abbildung 79. Sichern von WS-Manager-Daten und -Protokollen mit einer alternativen lokalen Sicherheitsgruppe, wmq

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 Steuerung der Berechtigung für die Verzeichnisse erteilen.
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 499.
5. Erstellen Sie den Warteschlangenmanager, indem Sie die alternative Sicherheitsgruppe oder den alternativen Principal im Befehl **crtmqm** mit dem Parameter **-a** angeben.

**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 den Assistenten "IBM MQ vorbereiten" aus. 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 470 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 470.

Unter Windows können Sie als Standardpfad für Daten- und Protokolldateien für einen IBM MQ for Windows beliebige Verzeichnisse festlegen. Der Installations- und Konfigurationsassistent gibt automatisch die lokale `mqm` -Gruppe und die Benutzer-ID, die die WS-Manager-Prozesse ausführt, 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 volle Kontrolle über seine Daten und Protokolldateien, indem er die lokale `mqm` -Gruppenberechtigung für das Verzeichnis `c:\wmq` erteilt.

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 den Lese- und Schreibzugriff auf die lokale `mqm` -Gruppe zu ermöglichen.

```
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 ein 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 Gruppe `mqm`, d. h., Sie müssen die Benutzer-ID nicht direkt zu einem Mitglied der lokalen Gruppe `mqm` machen.

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. Überprüfen Sie, ob sich die vom WS-Manager erstellten Verzeichnisse im Verzeichnis `c:\wmq` befinden.

```
dir c:\wmq/D /B /S
```

7. Überprüfen Sie, ob die Dateien über Lese- und Schreibzugriff oder die vollständige Steuerberechtigung für die lokale `mqm`-Gruppe verfügen.

```
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 7.1.0.0.
```

2. Erstellen Sie eine Testwarteschlange.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Systemantwort:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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 **amqspu**tein.

```
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 die Befehle aufgefordert werden, die Datei oder das Verzeichnis zu löschen.

```
del /F /S C:\wmq\*.*  
rmdir /S C:\wmq
```

### Zugehörige Konzepte

„[Windows-Domänen und Multi-Instanz-Warteschlangenmanager](#)“ auf Seite 470

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 485

„WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen“ auf Seite 472

**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 den Assistenten "IBM MQ vorbereiten" aus. 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 470 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 470.

Unter Windows können Sie als Standardpfad für Daten- und Protokolldateien für einen IBM MQ for Windows beliebige Verzeichnisse festlegen. Der Installations- und Konfigurationsassistent gibt automatisch die lokale mqm -Gruppe und die Benutzer-ID, die die WS-Manager-Prozesse ausführt, 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 Steuerberechtigung 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** *al<sup>t</sup>mqm* 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, mit der der IBM MQ-Service ausgeführt wird, zur Gruppe *al<sup>t</sup>mqm* 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 von der Gruppe *al<sup>t</sup>mqm*, die die Verzeichnisse autorisiert hat, sein.

Wenn Sie IBM MQ als Teil einer Domäne installiert haben und die Tasks in „WS-Manager mit mehreren Instanzen auf Domänenworkstations oder Servern unter Windows erstellen“ auf Seite 472 ausführen werden, sind die in „Active Directory- und DNS-Domäne unter Windows erstellen“ auf Seite 475 erstellten Domänenbenutzer-IDs *wmquser1* und *wmquser2*.

Wenn Sie den WS-Manager nicht als Teil einer Domäne installiert haben, lautet die lokale Standardbenutzer-ID, die den IBM MQ-Service ausführt, *MUSR\_MQADMIN*. Wenn Sie die Tasks ohne Administratorberechtigung ausführen wollen, erstellen Sie einen Benutzer, der Mitglied der lokalen *mqm* -Gruppe ist.

Führen Sie die folgenden Schritte aus, um *wmquser1* und *wmquser2* zu *al<sup>t</sup>mqm* 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 **al<sup>t</sup>mqm > 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, 2023. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

Und:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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 ermöglichen.

```
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 ein 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 Gruppe *mqm*, d. h., Sie müssen die Benutzer-ID nicht direkt zu einem Mitglied der lokalen Gruppe *mqm* machen.

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. Überprüfen Sie, ob sich die vom WS-Manager erstellten Verzeichnisse im Verzeichnis *c:\wmq* befinden.

```
dir c:\wmq\D /B /S
```

- Überprüfen Sie, ob die Dateien über Lese- und Schreibzugriff oder die vollständige Steuerberechtigung für die lokale mqm -Gruppe verfügen.

```
cacls c:\wmq\*.*
```

## Nächste Schritte

Testen Sie den Warteschlangenmanager, indem Sie eine Nachricht in eine Warteschlange einreihen und eine Nachricht erhalten.

- 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 7.1.0.0.
```

- Erstellen Sie eine Testwarteschlange.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Systemantwort:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. 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.
```

- Reihen Sie eine Testnachricht mit dem Beispielprogramm **amqspu**tein.

```
echo 'A test message' | amqsput QTEST QMGR
```

Systemantwort:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

- Rufen Sie die Testnachricht mit dem Beispielprogramm **amqsget**ab.

```
amqsget QTEST QMGR
```

Systemantwort:

```
Sample AMQSGETO start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGETO 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 die Befehle aufgefordert werden, die Datei oder das Verzeichnis zu löschen.

```
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

*Mehrinstanz-Warteschlangenmanager unter Linux erstellen*

Anhand eines Beispiels sehen Sie, wie Sie einen Warteschlangenmanager mit mehreren Instanzen unter Linux konfigurieren können. 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-Netztes 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. Beispiel:
    - 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 anzuhängen. Behalten Sie den Pfad auf dem Pfad zu Server1. Beispiel:
    - 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 d1wx1wx1-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<sup>2</sup>/MQHA \*(rw, sync, no\_wdelay, fsid=0) zu /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:

---

<sup>2</sup> 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 Befehl zum Starten der Warteschlangenmanagerinstanzen muss in 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 **amqsgbac**, **amqspbac** und **amqsmbac**, um eine Konfiguration mit mehreren Instanzen des Warteschlangenmanagers zu prüfen. Dieser Abschnitt enthält eine Beispielfunktion 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 [„Mehrinstanz-Warteschlangenmanager unter Linux erstellen“](#) auf Seite 512 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, Option. 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 das Hosting von /MQHA handelt!

**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.

Sie haben möglicherweise einen der Server ausgewählt, um das MQHA -Verzeichnis oder das Dateisystem zu hosten. Wenn Sie die Übernahme testen möchten, indem Sie das VMware-Fenster mit dem aktiven Server schließen, stellen Sie sicher, dass es sich nicht um das Hosting von MQHA handelt!

### 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. Stellen Sie sicher, dass der Listener aktiv ist und die Steuerung des Warteschlangenmanagers aufweist, 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 TCP/IP-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 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
Beispiel für AMQSPHAC-Start
Zielwarteschlange ist SOURCE
Nachricht Message 1
Nachricht Message 2
Nachricht Message 3
Nachricht Message 4
Nachricht Message 5
17:05:25: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:47: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:52: EVENT: Verbindung wurde wieder hergestellt
Nachricht Message 6
Nachricht Message 7
Nachricht Message 8
Nachricht Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Start Beispiel AMQSMHA0
17:05:25: EVENT: Connection Reconnecting (Delay: 97ms)
17:05:48: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```

N:\>amqsgnac TARGET QM1
Beispiel für AMQSGHAC-Start
Nachricht Message 1
Nachricht Message 2
Nachricht Message 3
Nachricht Message 4
Nachricht Message 5
17:05:25: EVENT: Connection Reconnecting (Delay: 156ms)
17:05:47: EVENT: Connection Reconnecting (Delay: 0ms)
17:05:52: EVENT: Verbindung wurde wieder hergestellt
Nachricht Message 6
Nachricht Message 7
Nachricht Message 8
Nachricht Message 9

```

## Multi **Multi-Instanz-WS-Manager löschen**

Wenn auf Multiplattformen ein Warteschlangenmanager mit mehreren Instanzen vollständig gelöscht werden soll, verwenden Sie den Befehl **dltmqm**, um den Warteschlangenmanager zu löschen und anschließend Instanzen von anderen Servern mit den Befehlen **rmvmqinf** oder **dltmqm** zu entfernen.

Führen Sie den Befehl **dltmqm** aus, um einen Warteschlangenmanager zu löschen, der Instanzen definiert, die 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 Sie ihn erstellt haben. 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 WS-Manager-Daten und -Protokolle gelöscht, und die Zeilengruppe des Warteschlangenmanagers wird aus der Datei `mqs.ini` auf dem Server entfernt, auf dem der Befehl **dltmqm** ausgeführt wurde. 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 im Gegenzug besuchen und die WS-Manager-Zeilengruppe entfernen, indem Sie den Befehl **rmvmqinf Queue manager stanza name** ausführen.

## Linux UNIX

Wenn Sie auf UNIX and Linux-Systemen eine allgemeine `mqs.ini`-Datei in den Netzspeicher gestellt und von allen Servern referenziert haben, indem Sie die Umgebungsvariable `AMQ_QMS_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.

Wenn Sie einen Warteschlangenmanager mit mehreren Instanzen ausführen möchten, starten Sie den Warteschlangenmanager auf einem der Server mit dem Befehl **strmqm -x QM1**. Die Option **-x** ermöglicht die Übernahme der Instanz. Es wird zur *aktiven Instanz*. Starten Sie die Standby-Instanz auf dem anderen Server mit demselben **strmqm -x QM1**. Die Option **-x** ermöglicht es der Instanz, als Standby-Instanz zu starten.

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 auf die Standby-Instanz umschalten, indem Sie den Befehl **endmqm -s** für die aktive Instanz ausgeben. Mit dem Befehl **endmqm -s** wird die aktive Instanz heruntergefahren, 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 die aktiven Instanzen und die Standby-Instanzen gestoppt. Wenn Sie die Standby-Instanz mit dem Befehl **endmqm** mit der Option **-x** stoppen, wird die Standby-Instanz gestoppt, und die aktive Instanz wird weiterhin ausgeführt. Sie können **endmqm** nicht ohne die Option **-x** in der Bereitschaftsdatenbank ausgeben.

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.

### **Gemeinsam genutztes Dateisystem**

Auf Multiplatforms verwendet ein Warteschlangenmanager mit mehreren Instanzen ein Netzdateisystem, um Warteschlangenmanagerinstanzen zu verwalten.

Ein Warteschlangenmanager mit mehreren Instanzen automatisiert die Funktionsübernahme mit einer Kombination aus Dateisystemsperrern 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 Protokoll-datenträger 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 WS-Manager-Instanz abgesetzt wird. Sie können die Parameter **endmqm -c** , **-i** oder **-p** angeben, um zu steuern, wie abrupt der WS-Manager 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 Switchover: Es startet keine Standby-Warteschlangenmanagerinstanz.

- *Failover* tritt auf, wenn die Sperre für WS-Manager-Daten, die von der aktiven Instanz gehalten werden, freigegeben wird, weil die Instanz anscheinend unerwartet gestoppt wurde (d. B. ohne dass ein Befehl **endmqm** ausgegeben wird).

Wenn die Standby-Instanz als aktive Instanz übernimmt, schreibt sie eine Nachricht in das Fehlerprotokoll des Warteschlangenmanagers.

Bei einem Ausfall oder Umschalten eines WS-Managers werden die wiederverbindbaren Clients automatisch wieder verbunden. Sie müssen das Flag **-r** nicht auf den Befehl **endmqm** einschließen, um eine Clientverbindungswiederverbindung anzufordern. Die automatische Wiederherstellung einer Clientverbindung 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. Die automatische Wiederherstellung einer Clientverbindung 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 `CONNNAME` von Kanälen ist eine durch Kommas getrennte Liste mit Verbindungsnamen, z. B. `CONNNAME ('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 `CONNNAME` des `CLUSRCVR`-Kanals auf dem Warteschlangenmanager. Bei `TCPIP` setzt der Warteschlangenmanager den `CONNNAME` 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 `CONNNAME`.

Wenn dies erforderlich ist, können Sie `CONNNAME` manuell mit der Liste der Netzadressen der Warteschlangenmanagerinstanzen konfigurieren.

## Clientverbindungen

Clientverbindungen können Verbindungslisten oder Warteschlangenmanagergruppen verwenden, um alternative Verbindungen auszuwählen. Clients müssen so kompiliert werden, dass sie mit Clientbibliotheken von IBM WebSphere MQ 7.0.1 oder besser ausgeführt werden. Sie müssen mit mindestens einem IBM WebSphere MQ 7.0.1-Warteschlangenmanager verbunden sein.

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 einphasigen Festschreibung, die nur Nachrichten umfasst, wird eine Transaktion, die die COMMIT-Verarbeitung gestartet hat, normalerweise unter der Steuerung des Warteschlangenmanagers abgeschlossen, 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\_CONNECTI-ON\_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 die Eingabe persistenter Nachrichten aus einer Warteschlange und reiht Antworten auf persistente Nachrichten als Teil einer einzelnen Arbeitseinheit in eine andere Warteschlange ein: Wenn sie einen MQRC\_BACKED\_OUT -Ursachencode von MQPUT, MQGET oder MQCMIT innerhalb eines Synchronisationspunkts durch einen Neustart der Arbeitseinheit verarbeitet, 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 Funktionsübernahme (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 523 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 IBM MQ -Statusinformationen, wie z. B. Warteschlangenkennungen, löschen und einen neuen MQCONN -oder MQCONNX -Aufruf ausgeben, um eine neue Verbindung herzustellen, und dann 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 522, 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.

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 zerstörerisches 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 rückgängig gemacht.
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.

- Das kooperative Durchsuchen markierter Nachrichten ist in den meisten Fällen 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\_AVAILABLE zurück.

**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 unter einem Synchronisationspunkt in einer nicht festgeschriebenen Transaktion in eine Warteschlange eingereiht 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 Verbindungswiederherstellung während eines MQPUT -Aufrufs außerhalb des Synchronisationspunkts gibt MQRC\_CALL\_INTERRUPTED zurück, wenn der IBM MQ MQI client nicht weiß, ob die Nachricht erfolgreich an den Warteschlangenmanager 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 mit einem MQPUT-Aufruf außerhalb des Synchronisationspunkts.
  - die Zustellung einer persistenten Nachricht oder einer Nachricht mit Standardpersistenz unter Verwendung eines MQPUT1-Aufrufs außerhalb des Synchronisationspunkts.
  - Festschreibung einer Transaktion mit einem MQCMIT-Aufruf. 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 wiederhergestellt, aber Anwendungen, die die Warteschlange geöffnet hatten oder sich an den Namen der Warteschlange erinnert hatten, 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 Konfigurationsdaten werden entweder von IBM MQ verwaltet, wie z. B. die Datei `mq5.ini` auf Windows-, UNIX and Linux-Systemen oder ein Teil Ihrer eigenen Konfigurationsverwaltung, 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 UNIX 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 beliebigen 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

 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. Die automatische Wiederherstellung einer Clientverbindung 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

V 9.0.4

## RDQM-Hochverfügbarkeit

RDQM (Replicated Data Queue Manager = Warteschlangenmanager mit replizierten Daten) ist eine Hochverfügbarkeitslösung, die auf Linux-Plattformen verfügbar ist.

Eine RDQM-Konfiguration besteht aus drei Servern, die in einer HA-Gruppe mit jeweils einer Instanz des Warteschlangenmanagers konfiguriert sind. 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 teilen sich eine variable IP-Adresse, sodass Clients nur mit einer einzigen IP-Adresse konfiguriert werden müssen. Es kann jeweils nur eine Instanz des Warteschlangenmanagers ausgeführt werden, auch 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.

Drei Knoten werden verwendet, um die Möglichkeit einer Split-Brain-Situation stark zu reduzieren. In einem System mit zwei Knoten kann eine Split-Brain-Bedingung auftreten, wenn die Konnektivität zwischen den beiden Knoten unterbrochen wird. 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 Warteschlangenmanager 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 die Verbindung zu einem einzelnen Knoten unterbrochen wird, kann der Warteschlangenmanager auf diesem Knoten nicht ausgeführt werden, da der Knoten kein Quorum hat. 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 591 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.

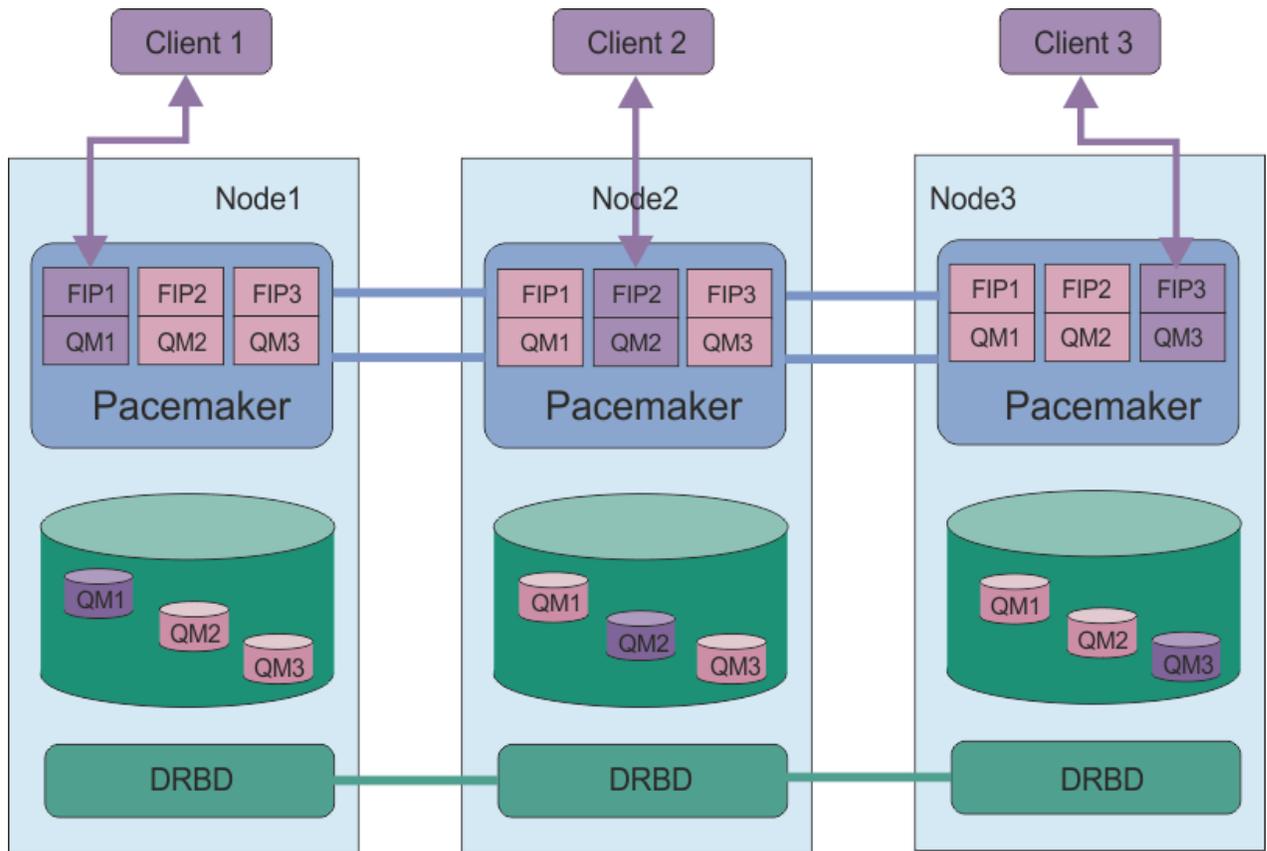


Abbildung 80. Beispiel für eine HA-Gruppe mit drei RDQMs

In der nächsten Abbildung ist Node3 fehlgeschlagen, die Pacemaker-Links sind verloren gegangen, und der Warteschlangenmanager QM3 wird stattdessen auf Knoten2 ausgeführt.

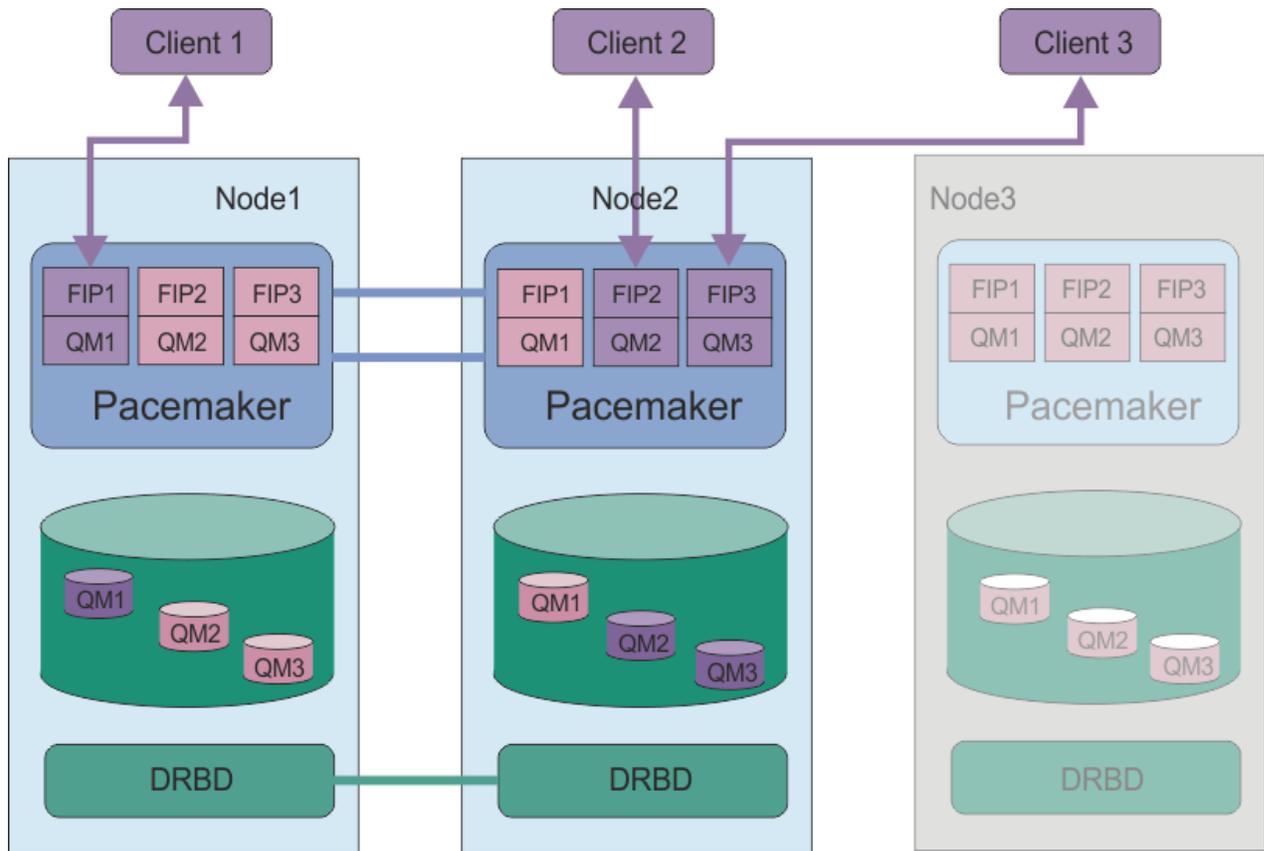


Abbildung 81. Beispiel für Fehlschlagen von node3

### Zugehörige Informationen

[RDQM \(replizierte Datenwarteschlangenmanager\) installieren](#)

[Warteschlangenmanager für replizierte Daten migrieren](#)

### Linux V 9.0.4 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 sollten `drbdpool` erstellen, nachdem Sie die RDQM-HA-Lösung installiert haben, aber bevor Sie tatsächlich RDQMs erstellen. Überprüfen Sie Ihre Datenträgergruppenkonfiguration mit dem Befehl `vg`. 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.

- Führen Sie nach dem Erstellen der Datenträgergruppe `drbdpool` keine weiteren Schritte aus. IBM MQ verwaltet die in `drbdpool` erstellten logischen Datenträger und gibt an, wie und wo sie angehängt 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 über einen gültigen Internet-Host-Namen (der von `uname -n` zurückgegebene Wert) verfügt, 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`, mit dem die erforderlichen Ports geöffnet werden, wenn Sie die Standardfirewall in RHEL ausführen. Sie müssen das Script als `root` ausführen. Wenn Sie eine andere Firewall verwenden, untersuchen Sie die Servicedefinitionen `/usr/lib/firewalld/services/rdqm*`, um festzustellen, welche Ports geöffnet werden müssen.
- 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 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 eine variable IP-Adresse konfigurieren, um einem Client die Verwendung derselben IP-Adresse für einen replizierten Datenwarteschlangenmanager (RDQM) zu ermöglichen, unabhängig davon, welcher Knoten in der HA-Gruppe 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 es nicht als `root` konfigurieren möchten, müssen Sie stattdessen als Benutzer in der Gruppe `mqm` konfigurieren. Damit ein `mqm`-Benutzer den RDQM-Cluster konfigurieren kann, müssen Sie die folgenden Voraussetzungen erfüllen:

- Der Benutzer von `mqm` muss in der Lage sein, die Ausführung von Befehlen auf jedem der drei Server, aus denen die RDQM-HA-Gruppe besteht, mit dem Befehl 'sudo' auszuführen.
- 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.
- Wenn Sie für Ihren `mqm`-Benutzer kennwortunabhängige SSH konfigurieren, muss dieser Benutzer die gleiche UID auf allen drei Servern haben.

Sie müssen `sudo` so konfigurieren, dass der `mqm`-Benutzer 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

Um replizierte Datenwarteschlangenmanager (RDQMs) zu erstellen, zu löschen oder zu konfigurieren, müssen Sie eine Benutzer-ID verwenden, die zu den Gruppen `mqm` und `haclient` gehört (die `haclient`-Gruppe wird während der Installation von Pacemaker erstellt).

**Linux** **V 9.0.4** *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.

## 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. Abschließend sperren Sie die `mqm`-ID.

**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.

## 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) Ändern Sie das Verzeichnis in /etc und bearbeiten Sie die Datei sudoers:

```
cd /etc
vi sudoers
```

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

V 9.0.4

## Definieren des Pacemaker-Clusters (HA-Gruppe)

Die HA-Gruppe ist ein Pacemaker-Cluster. Sie definieren den Pacemaker-Cluster, indem Sie die `/var/mqm/rdqm.ini` -Datei 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 `sudo` verwenden kann. Wenn der Benutzer auch ohne Kennwort SSH für jeden Server verwenden kann, müssen Sie nur die Datei `rdqm.ini` bearbeiten und auf einem der Server den Befehl `rdqmadm` 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.

In der Datei `rdqm.ini` werden die IP-Adressen für alle Knoten im Pacemaker-Cluster angegeben. Sie können angeben, dass der Pacemaker-Cluster eine, zwei oder drei IP-Adressen verwendet. Die Schnittstelle, die für die synchrone Datenreplikation verwendet wird, wird als 'Replikationsschnittstelle' bezeichnet. Die Schnittstelle muss über eine ausreichende Bandbreite verfügen, um Replikationsanforderungen bei der erwarteten Auslastung aller RDQMs, die in der HA-Gruppe ausgeführt werden, zu unterstützen. Die primären und sekundären Schnittstellen werden für den Pacemaker verwendet, um das System zu überwachen, aber Pacemaker kann die Replikationsschnittstelle zu diesem Zweck verwenden, falls erforderlich.

Die folgende Beispieldatei zeigt die Konfiguration für einen Beispiel-Pacemaker-Cluster an, der für jede Schnittstelle eine separate IP-Adresse verwendet:

```
Node:
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3
```

Die folgende Beispieldatei zeigt die Konfiguration für einen Beispiel-Pacemaker-Cluster, der für jede Schnittstelle die gleiche IP-Adresse verwendet. Geben Sie in diesem Fall nur die Replikationsschnittstelle an:

```
Node:
  HA_Replication=192.168.4.1
Node:
  HA_Replication=192.168.4.2
Node:
  HA_Replication=192.168.4.3
```

Wenn Sie zwei IP-Adressen verwenden möchten, verfügt Ihre `rdqm.ini`-Datei über ein `Primary`- und ein `Replication`-Feld für jeden Knoten, aber kein `Alternate`-Feld:

```
Node:
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

## Prozedur

- Gehen Sie wie folgt vor, um den Pacemaker-Cluster als `mqm`-Benutzer zu
  - a) Stellen Sie sicher, dass der Benutzer `mqm sudo` für die Ausführung von Befehlen verwenden kann, und kann optional mit SSH ohne Kennwort eine Verbindung zu jedem Server herstellen.

- 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
```

(Wenn Sie SSH nicht ohne Kennwort verwenden können, müssen Sie die Datei `.ini` auf jeden Server kopieren und den Befehl auf jedem Server ausführen.)

- 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
```

### Zugehörige Informationen

[rdqmadm \(Verwaltung replizierter Daten-WS-Manager-Cluster\)](#)

**Linux** **V 9.0.4** *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 Informationen

[rdqmadm \(Verwaltung replizierter Daten-WS-Manager-Cluster\)](#)

**Linux** **V 9.0.4** *HA-RDQM erstellen*

Verwenden Sie den Befehl `crtmqm`, um einen replizierten Datenwarteschlangenmanager mit hoher Verfügbarkeit (RDQM) 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.

### 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 zu können.

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 dann den Befehl **crtmqm -sx** ausführen (wie für den 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.

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.

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.

Abgesehen von den Argumenten DataPath ( **-md** ) und LogPath ( **-ld** ) sind alle Argumente, die für die Erstellung eines Standard-Linux-Warteschlangenmanagers gültig sind, auch für einen primären replizierten Datenwarteschlangenmanager gültig.

## Zugehörige Informationen

[crtmqm](#)

Linux V 9.0.4 HA-RDQM löschen

Sie verwenden den Befehl **dltmqm** , um einen replizierten Datenwarteschlangenmanager mit hoher Verfügbarkeit (RDQM) zu 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 Informationen

[dltmqm](#)

Linux

V 9.0.4

### **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 Informationen

[rdqmadm](#) (Verwaltung replizierter Daten-WS-Manager-Cluster)

Linux

V 9.0.4

### **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.

## 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.

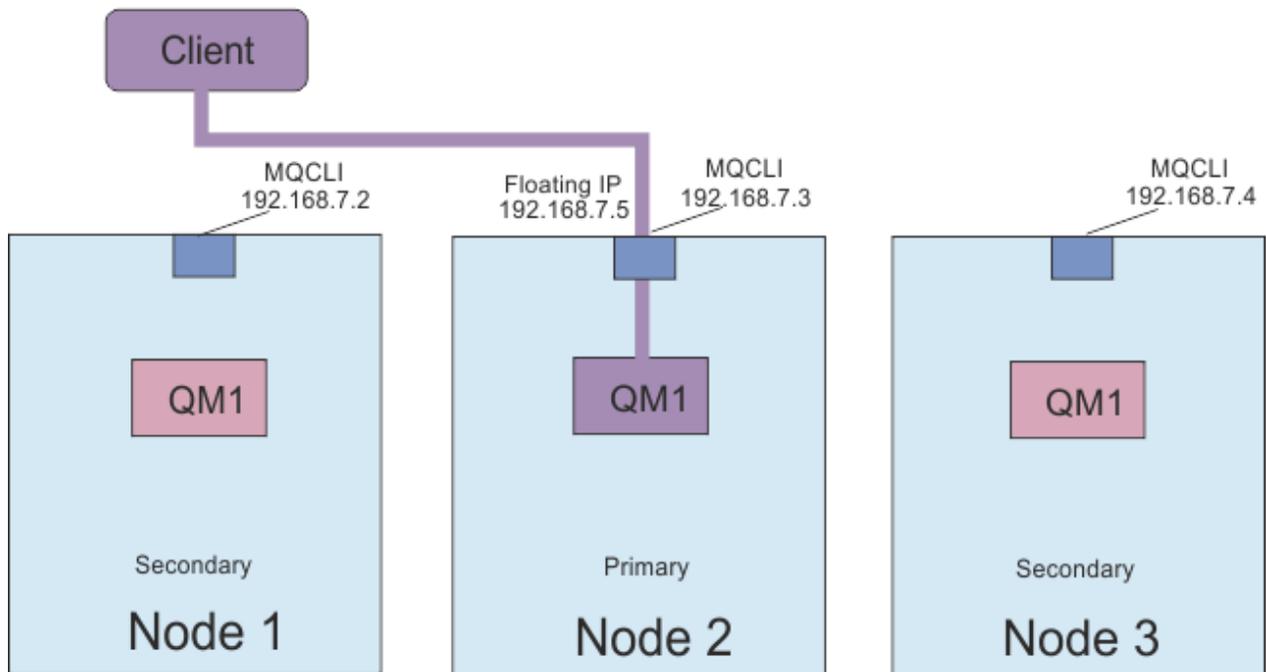
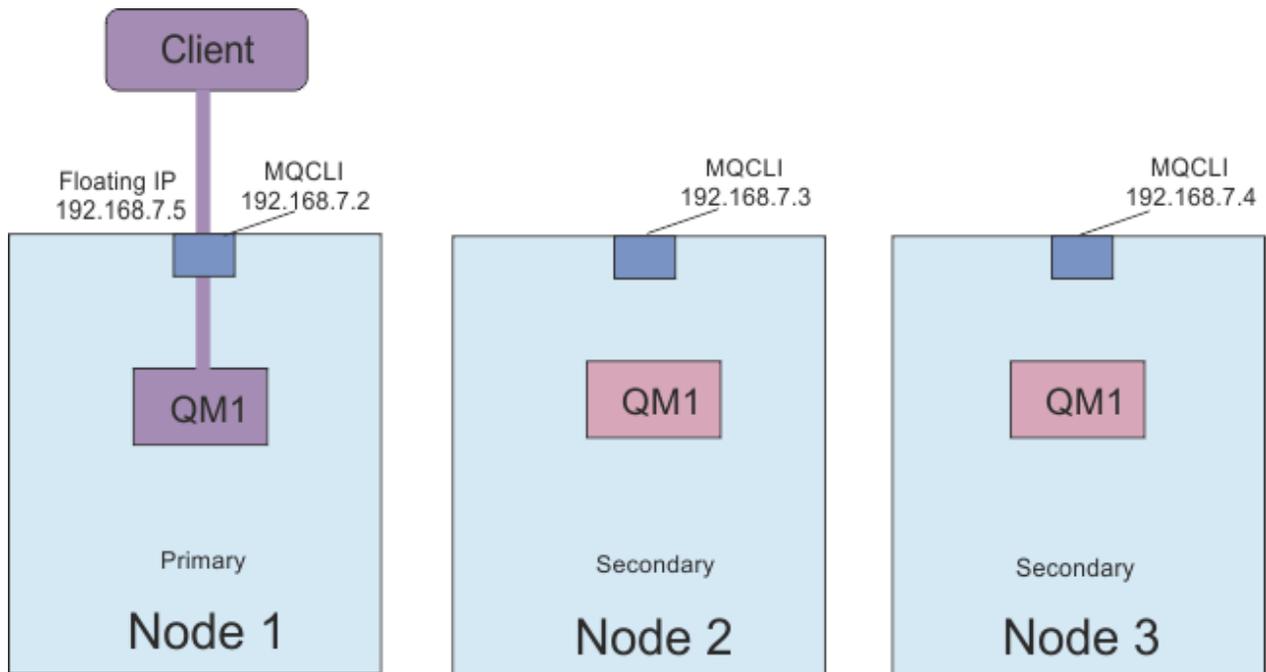


Abbildung 82. Variable IP-Adresse

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 variable IP-Adresse muss eine gültige IPv4-Adresse sein, die nicht bereits in einer anderen Appliance definiert ist, und sie muss zum gleichen Teilnetz wie die statischen IP-Adressen gehören, die für die lokale Schnittstelle definiert sind.

### **interfacename**

Der Name der physischen Schnittstelle auf dem Primärknoten, an die die Bindung gebunden werden soll.

Beispiel:

```
rdqmint -m QM1 -a -f 192.168.7.5 -l MQCLI
```

- Geben Sie den folgenden Befehl ein, um eine vorhandene variable IP-Adresse zu löschen:

```
rdqmint -m qmname -d
```

## **Zugehörige Informationen**

rdqmint (variable IP-Adresse für RDQM hinzufügen oder löschen)

Linux

V 9.0.4

## **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 Parameter **endmqm** können beim Stoppen eines RDQM-Systems 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, 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 `haclient` -Gruppe ist, wird der Status `Unavailable` gemeldet. Beispiel:

```
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 WS-Manager anzuzeigen, die einem Knoten bekannt sind, und ob sie RDQMs sind oder nicht. Beispiel:

```
dspmq -o ha
QMNAME (RDQM8)          HA(Replicated)
QMNAME (RDQM9)          HA(Replicated)
QMNAME (RDQM7)          HA(Replicated)
QMNAME (QM7)            HA()
```

## Zugehörige Informationen

[dspmq \(Warteschlangenmanager anzeigen\)](#)

[endmqm \(Warteschlangenmanager beenden\)](#)

[strmqm \(Warteschlangenmanager starten\)](#)

Linux

V 9.0.4

## 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.

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 Status eines Knotens und der RDQMs anzuzeigen, die Teil der HA-Konfiguration sind:

```
rdqmstatus
```

Die Angabe des Knotens, auf dem der Befehl ausgeführt wurde, und der Status der RDQMs in der HA-Konfiguration werden angezeigt, z. B.:

```
Node:                    mqhavm07.exampleco.com
Queue manager name:     RDQM8
Queue manager status:   Running elsewhere
HA current location:    mqhavm08.exampleco.com
```

```

Queue manager name:          RDQM9
Queue manager status:       Running elsewhere
HA current location:        mqhavam09.exampleco.com

Queue manager name:          RDQM7
Queue manager status:       Running
HA current location:        This node

```

- 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. Beispiel:

```

Node mqha04(mqhavam04.example.com) is online
Node mqha05(mqhavam05.example.com) is offline
Node mqha06(mqhavam06.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.

In der folgenden Tabelle sind die Informationen zu dem aktuellen Knoten zusammengefasst, der vom Befehl **rdqmstatus** für einen RDQM zurückgegeben werden kann.

Tabelle 30. Aktueller Knotenstatus		
Statusattribut	Mögliche Werte	Wird wann angezeigt
Node	<i>Knotenname</i>	Immer
Status des Warteschlangenmanagers	Aktiv Wird an anderer Stelle ausgeführt Beendet Nicht verfügbar.	Immer
CPU	<i>n.nn%</i>	Wird nur angezeigt, wenn der aktuelle Knoten über eine primäre Rolle verfügt (d. a. der RDQM wird auf diesem Knoten ausgeführt)
Hauptspeicher	<i>nnn</i> MB verwendet, <i>y.y</i> 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	<i>nnn</i> MB used, <i>y.y</i> GB allocated [ <i>z%</i> ]	Wird nur angezeigt, wenn der aktuelle Knoten über eine primäre Rolle verfügt (d. a. der RDQM wird auf diesem Knoten ausgeführt)
HA-Rolle	Primär sekundär unbekannt	Immer

<i>Tabelle 30. Aktueller Knotenstatus (Forts.)</i>		
<b>Statusattribut</b>	<b>Mögliche Werte</b>	<b>Wird wann angezeigt</b>
HA status	Alle Knoten im Standby-Modus This node in standby Remote nodes in standby Gemischt  <i>status of remote nodes</i>	Alle Knoten im Standby-Modus Aktueller Knoten im Standby-Modus Beide fernen Knoten im Standby-Modus Anderer Status für jeden fernen Knoten (siehe nächste Tabelle für den einzelnen Status)  Derselbe Status für beide fernen Knoten (siehe nächste Tabelle für alle Werte)
HA control	Enabled Inaktiviert Unbekannt	Immer. Zeigt an, ob RDQM von Pacemaker gesteuert wird.
HA preferred location	-- This node Unbekannt <i>Knotenname</i>	Immer
HA floating IP interface	<i>Schnittstellename</i>	Immer
HA floating IP address	<i>IPV4_address</i>	Immer

In der folgenden Tabelle sind die Informationen zusammengefasst, die vom Befehl **rdqmstatus** für die anderen Knoten in der HA-Gruppe zurückgegeben werden.

<i>Tabelle 31. Status des anderen Knotens</i>		
<b>Statusattribut</b>	<b>Mögliche Werte</b>	<b>Wird wann angezeigt</b>
Node	<i>nodename</i>	Immer
HA status	Normal (Normal) Synchronization in progress Fern nicht verfügbar Inkonsistent Angehalten Ferner Knoten in Bereitschaft Unbekannt	Knoten sind miteinander synchronisiert. Synchronisieren mit fernem Knoten Kommunikation mit fernem Knoten nicht möglich Nicht synchron zum fernen Knoten und nicht Synchronisieren Replikation angehalten Ferner Knoten in Bereitschaft
HA-Synchronisation in Bearbeitung	<i>n. n%</i>	Wird angezeigt, wenn die Synchronisation in Bearbeitung ist und der Befehl als <b>root</b> ausgeführt wird.
HA-geschätzte Synchronisationszeit	<i>yyyy-mm-dd hh:mm:ss.nnn</i>	Wird angezeigt, wenn die Synchronisation in Bearbeitung ist.

Tabelle 31. Status des anderen Knotens (Forts.)

Statusattribut	Mögliche Werte	Wird wann angezeigt
HA aus Synchronisationsdaten	<i>n</i> KB	Wird angezeigt, wenn der ferne Knoten nicht verfügbar oder inkonsistent

### Beispiel

Beispiel für einen normalen Status auf dem Primärknoten:

```

Node:                               mqhavam07.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 floating IP interface:             Eth4
HA floating IP address:               192.0.2.4

Node:                               mqhavam08.exampleco.com
HA status:                           Normal

Node:                               mqhavam09.exampleco.com
HA status:                           Normal

```

Beispiel für einen normalen Status auf einem Sekundärknoten:

```

Node:                               mqhavam08.exampleco.com
Queue manager status:               Running elsewhere
HA role:                             Secondary
HA status:                           Normal
HA control:                           Enabled
HA current location:                 mqhavam07.exampleco.com
HA preferred location:                mqhavam07.exampleco.com
HA floating IP interface:             Eth4
HA floating IP address:               192.0.2.4

Node:                               mqhavam07.exampleco.com
HA status:                           Normal

Node:                               mqhavam09.exampleco.com
HA status:                           Normal

```

Beispiel für den Status auf dem Primärknoten, wenn die Synchronisation in Bearbeitung ist:

```

Node:                               mqhavam07.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 floating IP interface:             Eth4
HA floating IP address:               192.0.2.4

Node:                               mqhavam08.exampleco.com
HA status:                           Synchronization in progress
HA synchronization progress:         11.0%
HA estimated time to completion:      2017-09-06 14:55:05

Node:                               mqhavam09.exampleco.com
HA status:                           Synchronization in progress
HA synchronization progress:         11.0%
HA estimated time to completion:      2017-09-06 14:55:06

```

Beispiel für einen Primärknoten mit mehreren Status:

```
Node: mqhvm07.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 floating IP interface: Eth4
HA floating IP address: 192.0.2.4

Node: mqhvm08.exampleco.com
HA status: Normal

Node: mqhvm09.exampleco.com
HA status: Inconsistent
```

## Zugehörige Informationen

 [rdqmstatus](#)

  **Fehlgeschlagenen Knoten 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.

## 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 534).
  - b) Führen Sie den Befehl `crtmqm -sxs qmanager` aus, der jeden Warteschlangenmanager mit replizierten Daten neu erstellt (siehe „HA-RDQM erstellen“ auf Seite 536).
- Wenn der Ersatzknoten über eine andere Konfiguration für den fehlgeschlagenen Knoten verfügt:
  - a) 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 537).
  - b) Dekonfigurieren Sie den Pacemaker-Cluster mit dem Befehl `rdqmadm -u` (siehe „Pacemaker-Cluster löschen (HA-Gruppe)“ auf Seite 536).
  - c) 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 534).
  - d) Führen Sie den Befehl `crtmqm -sxs qmanager` aus, der jeden Warteschlangenmanager mit replizierten Daten neu erstellt (siehe „HA-RDQM erstellen“ auf Seite 536).

 **Auflösen einer Split-Brain-Situation**

Es gibt Situationen, in denen bestimmte Ausfallfolgen in einer HA-Gruppe zu einer gespaltenen Situation führen können, die gemeldet wird.

Beispiel: Alle drei Knoten verlieren die Konnektivität. Wenn beide Sekundärknoten die Konnektivität vor dem Primärknoten wiedererlangen, bilden sie ein neues Quorum und einer von ihnen führt den Warteschlangenmanager aus. Wenn der ursprüngliche Primärknoten wieder in Verbindung steht, ist es möglich, dass eine Split-Brain-Situation gemeldet wird.

In dieser Situation zeigt die Ausführung von `rdqmstatus -m QMname` auf dem ursprünglichen Primärknoten den HA-Status wie folgt an: Inconsistent:

```
Node: node1
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Inconsistent
HA control: Enabled
HA current location: hanode2
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None

Node: node2
HA status: Inconsistent
HA out of sync data: 8KB

Node: node3
HA status: Inconsistent
HA out of sync data: 8KB
```

In diesem Fall sollten Sie die Daten auf den ursprünglichen Sekundärknoten (die das neue Quorum gebildet haben) beibehalten. Führen Sie die folgenden Schritte aus:

1. Führen Sie auf dem ursprünglichen Primärknoten als Root den folgenden Befehl aus:

```
drbdadm connect --discard-my-data QMname
```

2. Führen Sie auf jedem der Sekundärknoten als Root den folgenden Befehl aus:

```
drbdadm connect QMname:first-node-name
```

**drbdadm** ist ein Befehl, der von DRBD bereitgestellt wird. Sie wird als Teil des Pakets `drbd-utils` in `/usr/sbin/drbdadm` installiert.

## **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 Wiederherstellung nach einem Katastrophenfall hinzufügen, und ein WS-Manager kann nicht mit RDQM-Disaster-Recovery und RDQM-Hochverfügbarkeit konfiguriert werden.

Es können mehrere Paare von RDQM-Warteschlangenmanagern auf einer Reihe unterschiedlicher Server ausgeführt werden. Sie könnten zum Beispiel sechs primäre DR-Warteschlangenmanager auf demselben Knoten ausführen, während ihre Sekundärwarteschlangen auf sechs verschiedenen Knoten in sechs verschiedenen Rechenzentren konfiguriert sind. Ebenso können die Warteschlangenmanager von primären Disaster Recovery-WS-Managern auf verschiedenen Knoten ausgeführt werden, während alle ihre sekundäre Disaster Recovery-Warteschlange auf demselben Knoten ausgeführt werden. Einige Beispielkonfigurationen werden in den folgenden Diagrammen dargestellt.

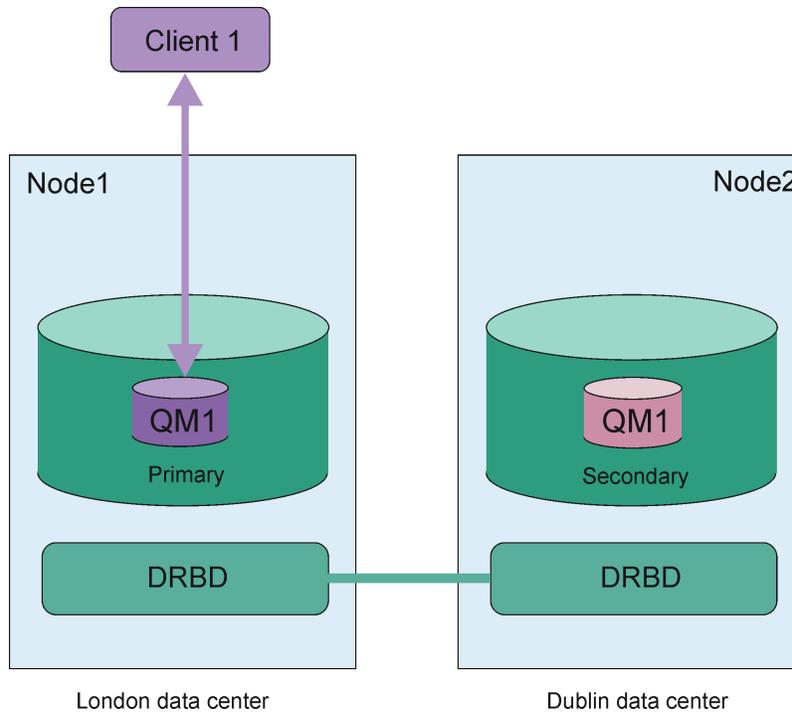


Abbildung 83. Einzelnes RDQM-Paar

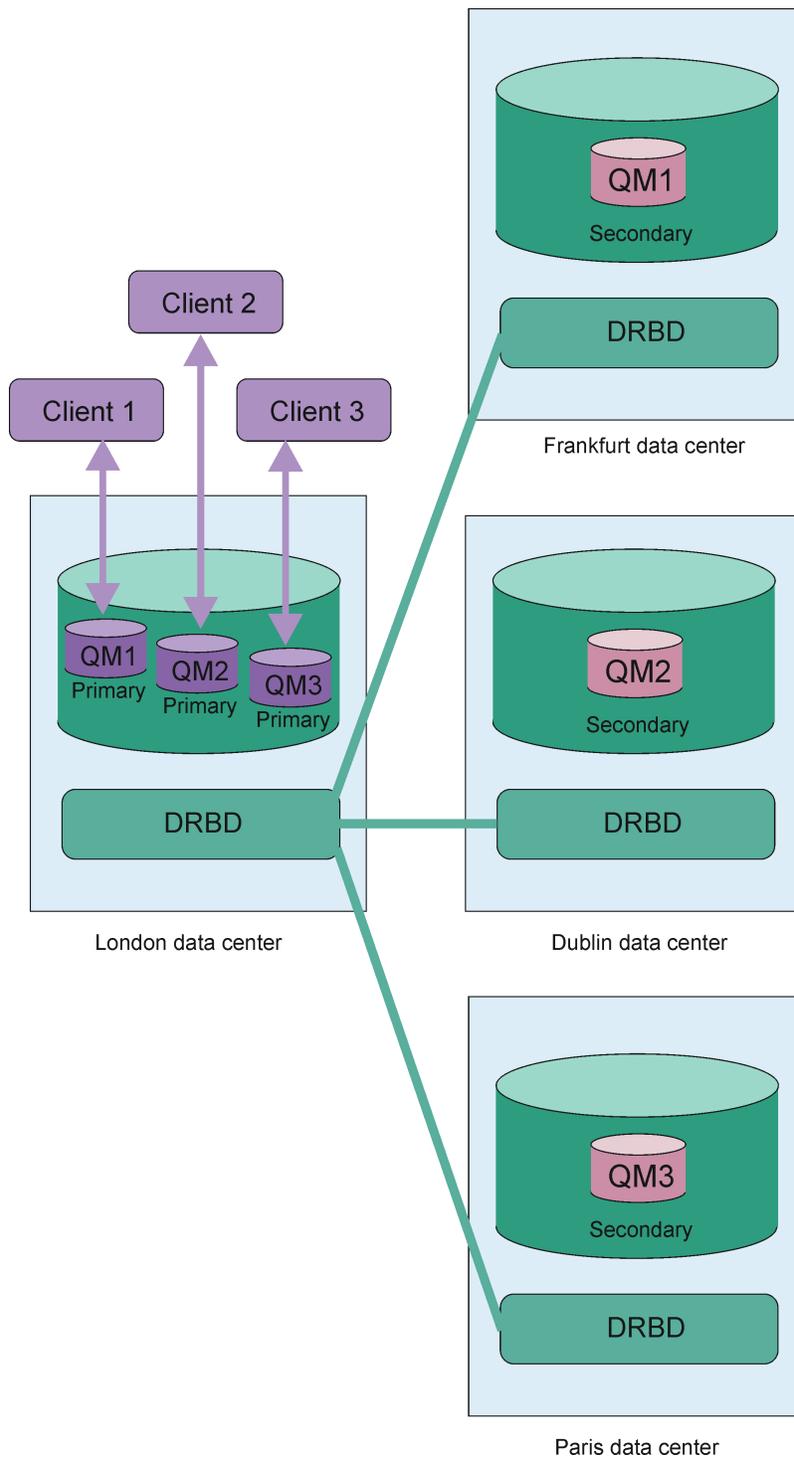


Abbildung 84. Primäre Warteschlangenmanager in demselben Knoten

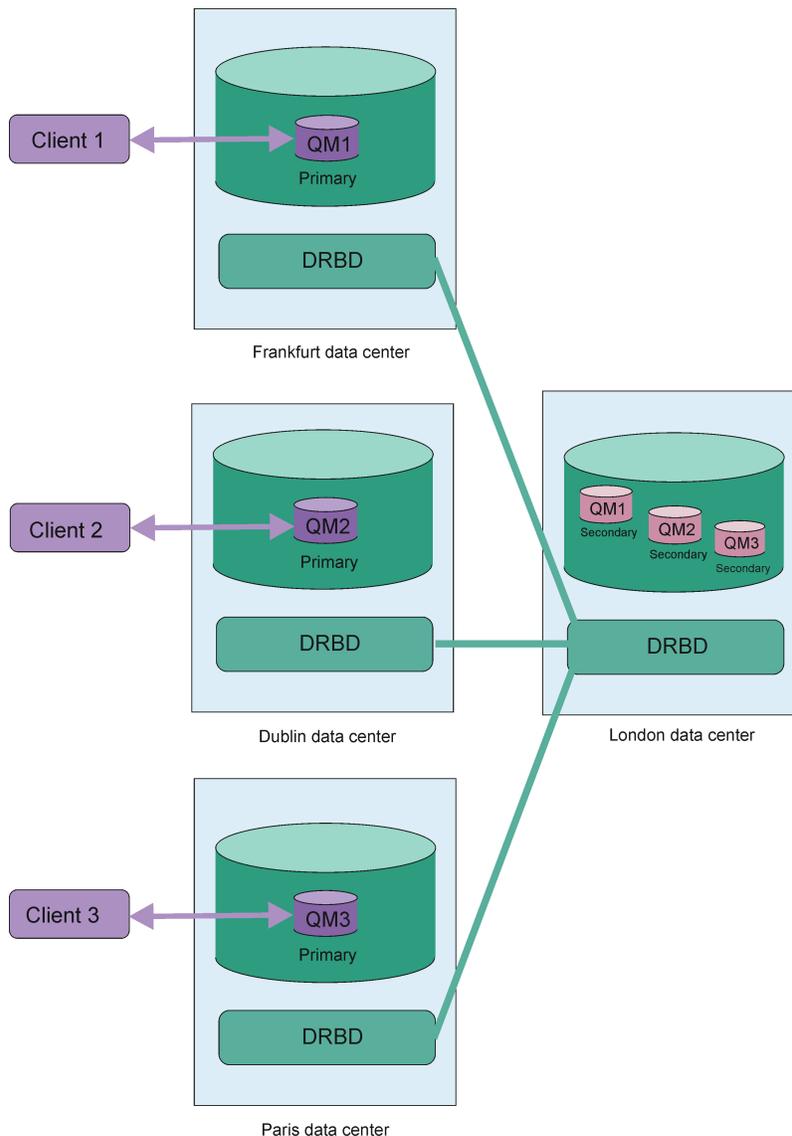


Abbildung 85. Sekundäre Warteschlangenmanager in demselben Knoten

## 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.

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).
- 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.
- 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 50 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 verwendet 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**
- **rdqmdx**

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**

### **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 funktioniert.

### 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.

### 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.

**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 **rdqmdx** 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 554).

- 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.

**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 555 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 Informationen

[crtmqm](#)

 DR RDQM löschen

Sie verwenden den Befehl **dltmqm**, um einen replizierten Datenwarteschlangenmanager (RDQM) für eine Notfallwiederherstellung 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 Informationen

[dltmqm](#)

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

Sie verwenden den Befehl **rdqmdr**, um eine sekundäre Instanz eines RDQM in die primäre Instanz zu 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.

Sie verwenden auch den Befehl **rdqmdr**, um eine primäre Instanz eines RDQM in die sekundäre Instanz zu ändern. Möglicherweise müssen Sie diese Aktion ausführen, z. B. wenn Sie Ihr System neu konfigurieren.

Sie können auch den Befehl **rdqmdr** auf einem primären Warteschlangenmanager verwenden, um den exakten Befehl abzurufen, den Sie zum Erstellen einer sekundären Instanz dieses Warteschlangenmanagers auf Ihrem Wiederherstellungsknoten benötigen.

Sie können den Befehl **rdqmdr** als Benutzer in der mqm -Gruppe 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 dem Primärknoten ein, um den Befehl **crtmqm** anzuzeigen, der für die Konfiguration der sekundären Instanz eines Warteschlangenmanagers erforderlich ist:

```
rdqmdr -d -m QMname
```

Sie können den zurückgegebenen Befehl **crtmqm** auf Ihrem Sekundärknoten eingeben, um das sekundäre Exemplar des DR RDQM zu erstellen.

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:

```
dspmqr -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 **dspmqr** zurückgegeben werden. Wenn Sie den Befehl auf einem Sekundärknoten ausführen, wird der Status `Ended immediately` angezeigt. Wird **dspmqr** beispielsweise auf dem Knoten RDQM7 ausgeführt, werden möglicherweise folgende Informationen zurückgegeben:

```
QMNAME(DRQM8)          STATUS(Ended immediately)
QMNAME(DRQM7)          STATUS(Running)
```

Sie können Argumente mit `dspmqr` verwenden, um festzustellen, ob ein RDQM für Disaster-Recovery konfiguriert ist und ob es sich aktuell um die Primär- oder Sekundärinstanz handelt:

```
dspmqr -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 Informationen

[dspmqr](#)

[endmqm](#)

[strmqm](#)

Linux

V 9.0.4

## 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.

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 Status 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.:

```
Queue manager name:      DRQM8
Queue manager status:    Ended immediately
DR role:                 Secondary

Queue manager name:      DRQM7
Queue manager status:    Running
DR role:                 Primary
```

- 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.

<i>Tabelle 32. Status-Attribute</i>		
<b>Statusattribut</b>	<b>Mögliche Werte</b>	<b>Wenn angezeigt</b>
Status des Warteschlangenmanagers	Status (wie von dspmq angezeigt)	Immer
CPU	<i>n. nn%</i>	Nur angezeigt, wenn RDQM auf dem aktuellen Knoten die primäre Rolle hat
Hauptspeicher	<i>nnn</i> MB	Nur angezeigt, wenn RDQM auf dem aktuellen Knoten die primäre Rolle hat
Queue manager file system	<i>nnn</i> MB verwendet, <i>n. n</i> GB zugeordnet [ <i>n%</i> ]	Nur angezeigt, wenn RDQM auf dem aktuellen Knoten die primäre Rolle hat
DR role	Primär Sekundär Unbekannt	Immer
DR-Status	Normal (Normal)	Normaler Vorgang.
	Synchronization in progress	Die Synchronisation ist in Bearbeitung.
	Partitioniert	Der WS-Manager wurde auf beiden Knoten gestartet, während das DR-Replikationsnetz nicht verfügbar ist.
	Fernes System nicht verfügbar	Die Verbindung zum anderen Knoten ist verloren gegangen.
	Inkonsistent	Es wurde eine Synchronisation in Bearbeitung, wurde aber unterbrochen

Tabelle 32. Status-Attribute (Forts.)		
Statusattribut	Mögliche Werte	Wenn angezeigt
	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	<i>port_number</i> (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	<i>n</i> KB	Wird angezeigt, wenn der ferne Knoten nicht verfügbar oder inkonsistent
DR synchronization progress	<i>n</i> %	Wird angezeigt, wenn die Synchronisation in Bearbeitung ist.
DR estimated time to completion	YYYY-MM-DD HH: MM: SS	Die geplante Synchronisation ist in Bearbeitung
Snapshot reversion progress	<i>n</i> %	Wird angezeigt, wenn der DR-Status Reverting to snapshot ist. Der Status zählt nach unten, also 0% zeigt die Fertigstellung an.

## 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
```

## Zugehörige Informationen

Linux [rdqmstatus](#)

V 9.0.5

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 559. 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 559.

#### **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 559.

#### **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 559 beschrieben. Sie müssen überprüfen, ob die Zurücksetzung auf Momentaufnahme abgeschlossen ist (mit dem Befehl **rdqmstatus** ), bevor Sie den WS-Manager starten.

V 9.0.5

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.

Sie müssen entweder als Root angemeldet sein 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.

## Zugehörige Informationen

[strmqm](#)

[rdqmdr](#)

V 9.0.5

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 sein 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 Informationen

[strmqm](#)

[rdqmdr](#)

[endmqm](#)

## 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.

### 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 „[Protokollstandardwerte für IBM MQ](#)“ auf Seite 117 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 Protokoll extents 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 im Abschnitt „[Typen der Protokollierung](#)“ auf Seite 562.

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
```

Wenn Sie in IBM MQ für UNIX and Linux-Systeme den Protokollpfad nicht geändert haben, werden die Protokollerweiterungen unter dem folgenden Verzeichnis erstellt:

```
/var/mqm/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 Protokoll extents zugeordnet werden. Sie können diese Standardzuordnung wie in „Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“ auf Seite 93 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 `amqh1ctl.1fh` 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.

### **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 564 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 Protokolldateien 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.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie einen vorhandenen Protokolltyp für einen Warteschlangenmanager mit dem Befehl `migmqlog` von linear zu umlaufend bzw. von umlaufend zu linear ändern.

## Änderungen der Protokollfunktion

**Multi** **V 9.0.2**

Wenn Sie in IBM MQ 9.0.2 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.

**ULW**

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 wiederverwendeter 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 568.)

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 Neustartwiederherstellung nicht mehr benötigt wird, wird sie zu `inactive`. 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 575.

## 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:** Beide 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.

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
-  Nachdem 50.000 Operationen protokolliert wurden, seit der vorherige Prüfpunkt
-  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 Verlauf der Wiedergabephase 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 Verlauf 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 Verlauf 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=000000001E45FF7AD to RBA=000000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE - 816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range 817
from RBA=000000001E48C1882 to RBA=000000001E48C1882
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 86 auf Seite 567 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. Protokoll-datei 1 kann jetzt wiederverwendet werden.

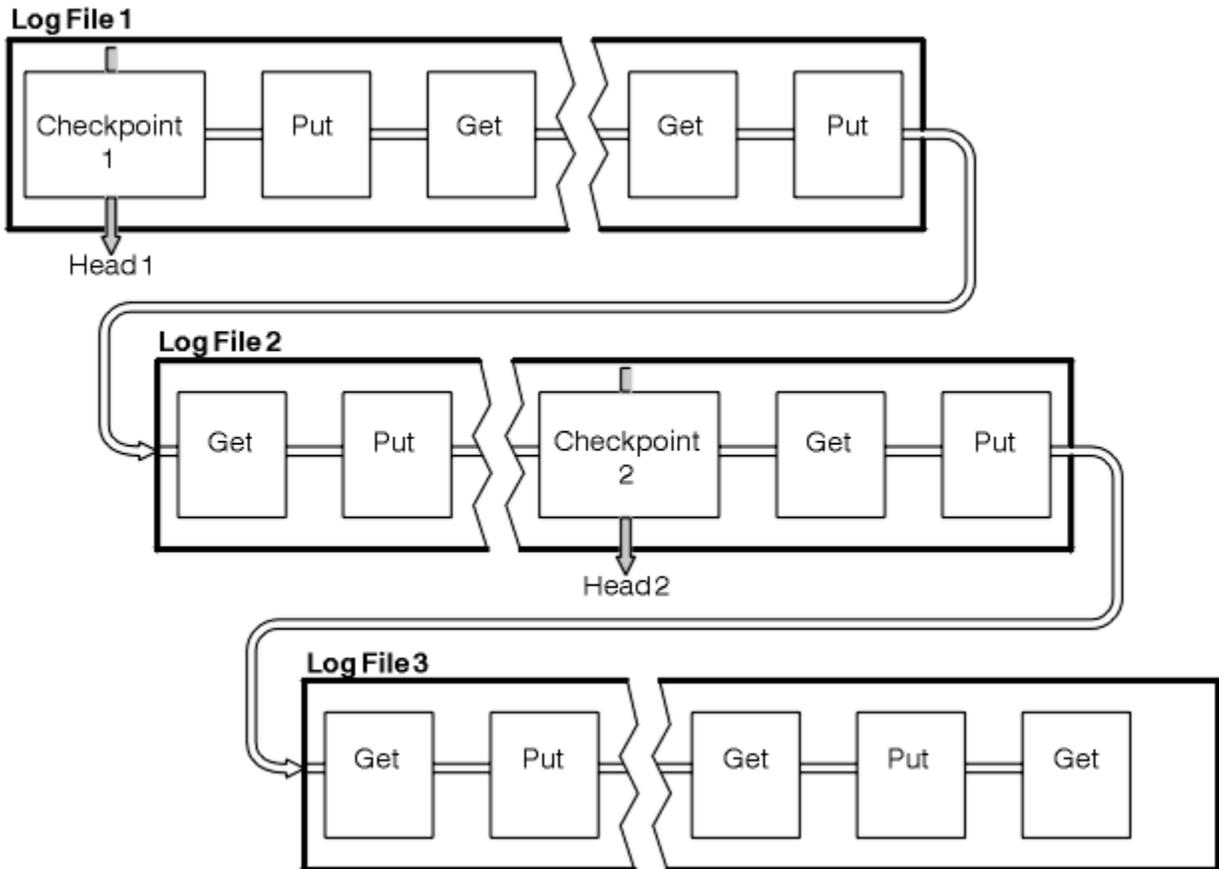


Abbildung 86. Prüfpunktprogramm

### **Prüfpunktverlaufes mit Transaktionen mit langer Laufzeit**

Wie eine Transaktion mit langer Laufzeit die Wiederverwendung von Protokolldateien beeinflusst.

Abbildung 87 auf Seite 568 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.

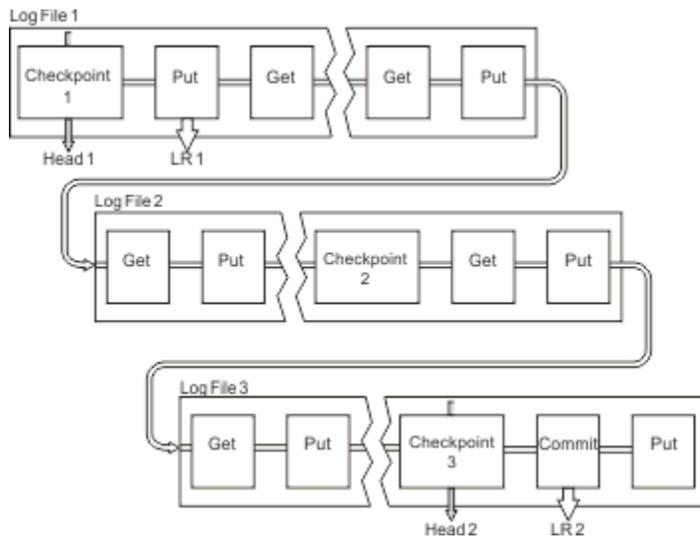


Abbildung 87. Prüfpunktüberprüfung mit einer Transaktion mit langer Laufzeit

## 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 UNIX 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 Protokollspei-

cherbereichs, 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 569. 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.

**V 9.0.2** Ab IBM MQ 9.0.2 versucht der Warteschlangenmanager, Prüfpunkte in einer Weise zu terminieren, die die Protokollbelegung in den primären Speicherbereichen hält.

Weitere Informationen finden Sie unter [„Protokollstandardwerte für IBM MQ“](#) auf Seite 117.

### **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 `dspmqrn -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 Bytes zeigen die Bytes pro Sekunde an, die der Warteschlangenmanager in das Protokoll schreibt. Beispiel:

```
$ 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 Bytes 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:

$$264\text{MiB} / (4096 \times 4096) = 16.5$$

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

Dabei ist Folgendes zu beachten:

- In diesem Beispiel beträgt die Anzahl der 5 sekundärer Server mehr als 20 Prozent des aktiven Protokollspeicherbereichs.

**V 9.0.2** Ab IBM MQ 9.0.2 versucht die Protokollfunktion, die Arbeitslast ausschließlich in den Primärdateien zu halten. Daher terminiert die Protokollfunktion Prüfpunkte, wenn ein Bruchteil der Primärdateien allein voll ist.

**V 9.0.2** 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 Protokoll extents, 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.

- **V 9.0.2** Ab IBM MQ 9.0.2 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 564 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.

**V 9.0.2** 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 579 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 564. 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.

### **V 9.0.2** *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 dimensionsieren, 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 vorhergehende Algorithmus setzt voraus, dass **SET LOG ARCHIVED** für jeden Speicherbereich aufgerufen wird, sobald er für die Datenträgerwiederherstellung nicht mehr benötigt wird, um die Verwaltung des Archivprotokolls zu verwalten.

## Protokolle verwalten

**V 9.0.2** Ab IBM MQ 9.0.2 unterstützt das Produkt die automatische Protokollverwaltung und die automatische Datenträgerwiederherstellung linearer Protokolle. Kreisprotokolle sind fast selbstverwaltet, erfordern aber manchmal einen Eingriff zur Behebung von Speicherplatzproblemen

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.

**V 9.0.2** Ab IBM MQ 9.0.2 können Sie bei Verwendung der Umlaufprotokollierung sekundäre Dateien löschen. Weitere Informationen finden Sie im Abschnitt [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 der Speicherbereich reserviert ist, um alle Inflight-Transaktionen festzuschreiben oder rückgängig zu machen. Daher sollte MQCMIT oder MQBACK nicht fehlschlagen.

Der WS-Manager macht Transaktionen rückgängig, die für eine lange Dauer ausgeführt werden. Eine Anwendung, die über eine Transaktion verfügt, wird auf diese Weise rückgängig gemacht, so dass nachfolgende MQPUT -oder MQGET -Operationen, die Synchronisationspunkt unter derselben Transaktion angeben, nicht ausgeführt werden können.

Transaktionen wurden jedoch manuell gestartet, um ein neues Protokoll zu starten. Beachten Sie, dass der neue Protokollspeicherbereich sofort zugeordnet wird, wenn der Protokollspeicherbereich freigegeben wird, der freigegeben wird, wenn er endlich freigegeben wird.

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**

Die Protokollierungskomponente des Warteschlangenmanagers kann mit einer vollständigen Platte und mit vollständigen Protokolldateien fertig werden. Wenn die Platte, die das Protokoll enthält, gefüllt ist, gibt der WS-Manager die Nachricht AMQ6709 aus, und es wird ein Fehlerdatensatz entnommen.

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.

**V 9.0.2** Ab IBM MQ 9.0.2 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](#)
- [IMGSCHEM](#)

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 „[Protokollstandardwerte für IBM MQ](#)“ auf Seite 117 und „[Warteschlangenmanagerprotokolle](#)“ auf Seite 126). 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:

#### **Protokollfunktionseignisnachrichten**

Wenn ein signifikantes Ereignis auftritt, z. B. ein Plattendatenträgerimage, werden Ereignisnachrichten der Protokollfunktion generiert. Der Inhalt der Protokollfunktionseignisnachrichten 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 im Abschnitt [Scriptbefehle \(MQSC\)](#) und Informationen zu PCF-Befehlen finden Sie im Abschnitt [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



Ab IBM MQ 9.0.2 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 [„Warteschlangenmanagerprotokolle“](#) auf Seite 126.

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 9.0.2 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 **IMGSCHE**  
(*MANUAL*) festlegen.

Sie können **IMGSCHE**D zu einem beliebigen Zeitpunkt während der Workload wechseln.



**Achtung: MEDIALOG** wird nicht vorwärts verschoben, wenn Sie keine Medienimages verwenden, daher müssen Sie entweder Speicherbereiche archivieren oder sicherstellen, dass genügend Platzspeicherplatz vorhanden ist.

Sie können auch automatische und manuelle Medienimages für andere benutzerdefinierte Objekte steuern:

- 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 **IMGR-COVQ** .

Weitere Informationen zum Attribut **IMGRCOVQ** finden Sie in [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 Fehlermeldung 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 Fehlermeldung ausgegeben, und es werden keine automatischen Medienimages geschrieben.
3. Sie können ein Medienimage mit `rcdmqimg` manuell aufzeichnen, wenn Sie **IMGSCHE**(*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. In IBM MQ 9.0.2 schreibt der Warteschlangenmanager nur persistente Nachrichten in Datenträgerimages, keine nicht persistente Nachrichten. Dadurch kann die Größe von Medienimages bei der Migration auf IBM MQ 9.0.2 oder höher reduziert werden.

## Legen Sie fest, wie **IMGLOGLN** und **IMGINTVL** festgelegt werden sollen.

V 9.0.2

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

V 9.0.1

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 lassen 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 Unter UNIX 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 587. `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`.

**Anmerkung:** Die Nachrichten AMQ7467 und AMQ7468 können auch zum Zeitpunkt der Ausführung des Befehls `rcdmqimg` ausgegeben werden.

## Partielle Medienimages

V 9.0.2

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 RECLG ab.
- Datenträgerwiederherstellung ausführen. Geben Sie den Befehl DISPLAY QMSTATUS MEDIALOG aus.
- **V 9.0.2** Bestimmen Sie den Namen für die Archivierungsbenachrichtigung, indem Sie den Befehl DISPLAY QMSTATUS ARCHLOG ausgeben.

**V 9.0.2** 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 RECLG-Wert ist, archivieren, und Sie können jede Protokolldatei mit einer Zahl löschen, die kleiner ist als die Werte für RECLG 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.

## **Protokoll für Wiederherstellung verwenden**

Sie können Informationen aus den Protokollen verwenden, um Sie bei der Wiederherstellung nach Fehlern zu unterstützen.

Es gibt mehrere Möglichkeiten, die Ihre Daten beschädigen können. IBM MQ unterstützt Sie bei der Wiederherstellung von:

- 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**

**V 9.0.2** Ab IBM MQ 9.0.2 können Medienimages auf einem Warteschlangenmanager mit linearer Protokollierung nur für wiederherstellbare Objekte aufgezeichnet werden. Sie müssen z. B. die Optionen **IMGRCOVO** und **IMGRCOVQ** berücksichtigen.

**V 9.0.2** Ebenso können Sie nur eine Untergruppe von Objekten wiederherstellen, die als wiederherstellbar definiert sind, von ihren Datenträgerimages auf einem linearen Protokollierungswarteschlangenmanager. Wenn ein Objekt, das nicht als wiederherstellbares Medium definiert ist, beschädigt ist, sind die Optionen für dieses Objekt mit denen für einen Umlaufprotokollierungswarteschlangenmanager identisch.

Durch die Datenträgerwiederherstellung werden Objekte aus Informationen, die in einem linearen Protokoll aufgezeichnet wurden, erneut erstellt. 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 erforderlich ist, um ein Objekt erneut zu erstellen, wird als *Datenträgerwiederherstellungsdatensatz* bezeichnet. Es ist der Start des letzten Datenträgerimages 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 **rcdmqimg** beschrieben. Dieser Befehl schreibt ein Datenträgerimage des IBM MQ-Objekts.

**V 9.0.2** Der Warteschlangenmanager zeichnet Medienimages automatisch auf, wenn **IMG-SCHED(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.

## Welche Objektdateien vorhanden sind

**V 9.0.1**

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.

Auf UNIX- und Linux-Plattformen werden Kanäle z. B. 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.

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 mit allen Typen und ist in `qmanager/QMQM0BJCAT` 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 Datenträgerwiederherstellung von Objekten erfolgt automatisch nur während des Systemstarts. Zu einem anderen Zeitpunkt, wenn Objektbeschädigung festgestellt wird, wird die Bedienernachricht `AMQ7472` ausgegeben, und die meisten Operationen, die das Objekt verwenden, schlagen fehl. Wenn das WS-Manager-Objekt zu einem beliebigen Zeitpunkt nach dem Start des Warteschlangenmanagers beschädigt ist, führt der Warteschlangenmanager eine bereits emptive Beendigung aus. Wenn ein Objekt beschädigt wurde, können Sie es löschen oder, wenn der Warteschlangenmanager ein lineares Protokoll verwendet, versuchen, ihn mit dem Befehl `rcrmqobj` aus seinem Datenträgerimage wiederherzustellen (weitere Informationen hierzu finden Sie im Abschnitt `rcrmqobj`).

**V 9.0.2** 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 (einschließlich 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.

**V 9.0.2** 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.

**V 9.0.2** 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.

**V 9.0.2** 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.

**V 9.0.2** 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.

## 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 *Protokollfolgennummer (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 instanziiieren, bevor Sie den Befehl ausführen. Sie müssen eine gültige Protokollfolgennummer (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 Protokollfolgennummer 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:

<i>logactive</i>	Die Anzahl primärer Protokollspeicherbereiche.
<i>loginactive</i>	Die Anzahl sekundärer Protokollerweiterungen.
<i>logsize</i>	Die Anzahl der 4-KB-Seiten pro Speicherbereich.
<i>baselsn</i>	Die erste Protokollfolgennummer (LSN) in der Protokollspeicherbereichsdatei, die den Protokollkopf enthält.

<i>nextlsn</i>	Die Protokollfolgennummer (LSN) des nächsten Protokollsatzes, der geschrieben werden soll.
<i>headlsn</i>	Die Protokollfolgennummer (LSN) des Protokollsatzes am Kopf des Protokolls.
<i>tailsn</i>	Die Protokollfolgennummer (LSN), die die Endposition des Protokolls angibt.
<i>hflag1</i>	Gibt an, ob das Protokoll CIRCULAR oder LOG RETAIN (linear) ist.
<i>HeadExtentID</i>	Die Protokollspeicherbereichsdatei, die den Protokollkopf enthält.

### Protokollsatz-Header

Jeder Protokollsatz innerhalb des Protokolls hat einen festen Header mit den folgenden Informationen:

<i>LSN</i>	Die Protokollfolgennummer.
<i>LogRecdType</i>	Der Typ des Protokollsatzes.
<i>XTranid</i>	Die Transaktions-ID, die diesem Protokollsatz zugeordnet ist (falls vorhanden).  Ein <i>TranType</i> -Wert von 'MQI' zeigt eine auf IBM MQ beschränkte Transaktion an. Ein <i>TranType</i> von XA ist an anderen Ressourcenmanagern beteiligt. Aktualisierungen, die in derselben Arbeitseinheit enthalten sind, haben denselben <i>XTranid</i> .
<i>QueueName</i>	Die Warteschlange, die diesem Protokollsatz zugeordnet ist (falls vorhanden).
<i>Qid</i>	Die eindeutige interne Kennung für die Warteschlange.
<i>PrevLSN</i>	Die Protokollfolgennummer (LSN) des vorherigen Protokollsatzes innerhalb derselben Transaktion (falls vorhanden).

### WS-Manager starten

Diese Protokolle, die der Warteschlangenmanager gestartet hat.

<i>StartDate</i>	Das Datum, an dem der WS-Manager gestartet wurde.
<i>StartTime</i>	Die Zeit, zu der der WS-Manager gestartet wurde.

### Warteschlangenmanager stoppen

Diese Protokolle, die der Warteschlangenmanager gestoppt hat, wurden gestoppt.

<i>StopDate</i>	Das Datum, an dem der WS-Manager gestoppt wurde.
<i>StopTime</i>	Die Zeit, zu der der WS-Manager gestoppt wurde.
<i>ForceFlag</i>	Der Typ des verwendeten Systemabschlusses.

### Prüfpunkt starten

Gibt den Start eines Warteschlangenmanagerprüfpunkts an.

### Prüfpunkt beenden

Gibt das Ende eines Warteschlangenmanagerprüfpunkts an.

<i>ChkPtLSN</i>	Die Protokollfolgennummer (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:

<i>MapIndex</i>	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 <i>Get Message</i> -Protokollsatz gefunden werden, der die gleichen <i>QueueName</i> und <i>MapIndex</i> enthält. Zu diesem Zeitpunkt kann die <i>MapIndex</i> -ID für eine nachfolgende Nachricht in diese Warteschlange wiederverwendet werden.
<i>Data</i>	Im Hex-Speicherauszug für diesen Protokollsatz sind verschiedene interne Daten enthalten, gefolgt von einer Darstellung des Nachrichten-deskriptors (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.

<i>Data</i>	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:

<i>MapIndex</i>	Bezeichnet die Nachricht, die aus der Warteschlange abgerufen wurde. Der aktuellste <i>Put Message</i> -Protokollsatz, der denselben <i>QueueName</i> und <i>MapIndex</i> enthält, gibt die Nachricht an, die abgerufen wurde.
<i>QPriority</i>	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 *twoPhaseCommit* 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:

<i>XTranid</i>	Die Transaktions-ID.
<i>FirstLSN</i>	Die Protokollfolgennummer (LSN) des ersten Protokollsatzes, der der Transaktion zugeordnet ist.
<i>LastLSN</i>	Die Protokollfolgennummer (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:

<i>RMName</i>	Der Name des Ressourcenmanagers.
<i>RMID</i>	Die Kennung des Ressourcenmanagers. Dies wird auch in nachfolgenden <i>Transaction Prepared</i> -Protokollsätzen protokolliert, in denen globale Transaktionen aufgezeichnet werden, an denen der Ressourcenmanager beteiligt ist.
<i>SwitchFile</i>	Die Switchloaddatei für diesen Ressourcenmanager.
<i>XAOpenString</i>	Die offene XA-Zeichenfolge für diesen Ressourcenmanager.
<i>XACloseString</i>	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.

<i>ObjName</i>	Der Name des Objekts, das erstellt wurde.
<i>UserId</i>	Die Benutzer-ID, die die Erstellung ausführt.

### Objekt löschen

Hiermit wird das Löschen eines IBM MQ-Objekts protokolliert.

<i>ObjName</i>	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

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 Protokoll-daten 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 Protokoll-daten 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 Protokoll-daten 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 591.
  - [„WS-Manager-Daten zurückschreiben“](#) auf Seite 593.
- Informationen zum Sichern und Wiederherstellen der WS-Manager-Konfiguration finden Sie unter:
  - [„WS-Manager-Konfiguration sichern“](#) auf Seite 593
  - [„Warteschlangenmanagerkonfiguration wird zurückgespeichert“](#) auf Seite 594
- Informationen zum Erstellen, Aktualisieren und Starten eines Backup-Warteschlangenmanagers finden Sie in [„Sicherungswarteschlangenmanager verwenden“](#) auf Seite 595.

## 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 [„Konfigurationsdaten von IBM MQ und von Warteschlangenmanagern ändern“](#) auf Seite 93.

**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 unter [Informationen zu IBM MQ-Dateinamen](#).

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 561 beschrieben, und die Konfigurationsdateien, wie im Abschnitt [„Initialisierungs- und Konfigurationsdateien“](#) auf Seite 204 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.

 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, stellen Sie sicher, dass Sie eine Sicherung der Datei `qm.ini` und der Registrierungseinträge erstellen. Die Warteschlangenmanagerinformationen werden in der `qm.ini`-Datei gespeichert; anhand dieser Informationen ist es möglich, eine frühere Version von IBM MQ wiederherzustellen.

## Zugehörige Tasks

[„Stoppen eines Warteschlangenmanagers“](#) auf Seite 13

Sie können den Befehl **endmqm** verwenden, um einen Warteschlangenmanager zu stoppen. Dieser Befehl bietet drei Möglichkeiten zum Stoppen eines Warteschlangenmanagers: ein gesteuerter oder stiller Systemabschluss, ein sofortiges Herunterfahren und ein präventives Herunterfahren. Alternativ dazu können Sie unter Windows und Linux einen Warteschlangenmanager mit IBM MQ Explorer stoppen.

[„Konfigurationsdateien nach der Erstellung eines Warteschlangenmanagers sichern“](#) auf Seite 11

Die Konfigurationsinformationen für IBM MQ werden unter UNIX, Linux, and Windows in Konfigurationsdateien gespeichert. Sichern Sie nach der Erstellung eines Warteschlangenmanagers Ihre Konfigurations-

dateien. 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 340 und [Clustering: Verfügbarkeit, mehrere Instanzen und Disaster-Recovery](#).

**Anmerkung:** Stellen Sie beim Upgrade auf IBM WebSphere MQ 7.5 und höher sicher, dass Sie ein Backup der **.ini**-Datei und der Registry-Einträge erstellen. Die Warteschlangenmanagerdaten sind in der **.ini**-Datei gespeichert und können zum Zurücksetzen 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 UNIX and Linux-Systemen](#).

### Ergebnisse

Wenn die Daten ordnungsgemäß gesichert und wiederhergestellt wurden, wird der WS-Manager jetzt gestartet.

## 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

**ULW** Unter UNIX, Linux, and Windows können Sie den Befehl **dmpmqcfcg** 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:

- **ULW** Unter UNIX, Linux, and Windows: Führen Sie den Befehl 'Dump MQ Configuration' (**dmpmqcfcg**) unter Verwendung der Standardformatierungsoption (-f mqsc) MQSC und aller Attribute (-a) aus, verwenden Sie die Standardausgabeumleitung, um die Definition in einer Datei zu speichern. Beispiel:

```
dmpmqcfcg -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. Beispiel:

```
DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC) TOMBR(MYQMGR) DEF)
```

## Zugehörige Tasks

„Warteschlangenmanagerkonfiguration wird zurückgespeichert“ auf Seite 594

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 Informationen

[dmpmqcfcg \(Konfiguration des Speicherauszugs-WS-Managers\)](#)

[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

**ULW** Unter UNIX, 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 **STRMQMQSC** 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:

- **ULW** Führen Sie unter UNIX, 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 **dmpmqcfcfg** (Speicherauszug MQ-Konfiguration) generiert wird (siehe „WS-Manager-Konfiguration sichern“ auf Seite 593). Beispiel:

```
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 ausgegeben) generiert wird (siehe „WS-Manager-Konfiguration sichern“ auf Seite 593). Beispiel:

```
STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
```

### Zugehörige Tasks

„WS-Manager-Konfiguration sichern“ auf Seite 593

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 Informationen

[dmpmqcfcfg \(Konfiguration des Speicherauszugs-WS-Managers\)](#)

[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 Hardwarefehlers 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 die Protokolldateien kopiert werden. Sie sollten jedoch nur dann eine Protokolldatei kopieren, wenn der WS-Manager die Schreibversuche abgeschlossen hat. 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 Siche-

rungswarteschlangenmanagers. 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

- Weitere Informationen zum Erstellen, Aktualisieren und Starten eines Sicherungswarteschlangenmanagers finden Sie in den folgenden Themen:
  - [„Sicherungswarteschlangenmanager erstellen“](#) auf Seite 596
  - [„Sicherungswarteschlangenmanager aktualisieren“](#) auf Seite 597
  - [„Sicherungswarteschlangenmanager starten“](#) auf Seite 597

## Zugehörige Konzepte

[„Protokollierung: Stellen Sie sicher, dass die Nachrichten nicht verloren gehen.“](#) auf Seite 561

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 **cxmqm** einen Sicherungswarteschlangenmanager für den vorhandenen WS-Manager.
2. Übernehmen Sie Kopien aller vorhandenen Daten- und Protokolldateiverzeichnisse des vorhandenen Warteschlangenmanagers, einschließlich aller Unterverzeichnisse, wie in [„WS-Manager-Daten sichern“](#) auf Seite 591 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 **stmqm** auf dem Sicherungswarteschlangenmanager aus, wie im folgenden Beispiel gezeigt:

```
stmqm -x BackupQMName
```

Dieser Befehl markiert den Warteschlangenmanager als Sicherungswarteschlangenmanager in IBM MQ und gibt alle kopierten Protokoll extents zurück, um den Sicherungswarteschlangenmanager in Schritt mit dem vorhandenen Warteschlangenmanager zu bringen.

## Zugehörige Informationen

[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 sicherenden 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 597 angegeben ist. Kopieren Sie nur Protokollspeicherbereichsdateien, die mit "S. .." beginnen.

4. Führen Sie den Steuerbefehl **strmqm** auf dem Sicherungswarteschlangenmanager aus, wie im folgenden Beispiel gezeigt:

```
strmqm -r BackupQMName
```

Dadurch werden alle kopierten Protokollreplays und der Sicherungswarteschlangenmanager 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 Informationen

[RESET QMGR](#)

[ANZEIGEN QMSTATUS](#)

[strmqm \(Warteschlangenmanager starten\)](#)

## Sicherungswarteschlangenmanager starten

Sie können einen Sicherungswarteschlangenmanager für einen nicht wiederherstellbaren Warteschlangenmanager ersetzen.

## Informationen zu diesem Vorgang

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 zu aktivieren, wie im folgenden Beispiel gezeigt:

```
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 zu starten, wie im folgenden Beispiel gezeigt:

```
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 195

Wenn ein Kanal in den Status STOPPED wechselt, müssen Sie den Kanal manuell erneut starten.

### Zugehörige Informationen

[strmqm \(Warteschlangenmanager starten\)](#)

## Änderungen an der Cluster-Fehlerbehebung (auf anderen Servern als z/OS)

Ab IBM WebSphere MQ 7.1 führt der Warteschlangenmanager Operationen erneut aus, die Probleme verursacht haben, 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.

Wenn ein Warteschlangenmanager vor IBM WebSphere MQ 7.1 ein Problem mit dem lokalen Repository-Manager, der einen Cluster verwaltet, festgestellt hat, hat er das Fehlerprotokoll aktualisiert. In einigen Fällen wurde die Verwaltung von Clustern dann gestoppt. Der Warteschlangenmanager hat weiterhin Anwendungsnachrichten mit einem Cluster ausgetauscht und dabei auf den Cache mit Clusterdefinitionen vertraut, die immer schneller veralteten. Ab IBM WebSphere MQ 7.1 führt der Warteschlangenmanager Operationen erneut aus, die Probleme verursacht haben, 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. Das geänderte Verhalten in Bezug auf Clusterfehler in 7.1 oder höher gilt nicht für z/OS.

Jeder Aspekt der Clusterverwaltung wird für einen Warteschlangenmanager durch den lokalen Repository-Manager-Prozess `amqrmfa` behandelt. Der Prozess wird auf allen Warteschlangenmanagern ausgeführt, selbst wenn keine Clusterdefinitionen vorhanden sind.

Wenn der Warteschlangenmanager vor IBM WebSphere MQ 7.1 ein Problem im lokalen Repository-Manager festgestellt hat, hat er den Repository-Manager nach einem kurzen Intervall gestoppt. Der Warteschlangenmanager lief weiter und verarbeitete Anwendungsnachrichten und Anfragen zum Öffnen von Warteschlangen und zum Veröffentlichen oder Abonnieren von Topics.

Wenn der Repository-Manager gestoppt wird, wird der Cache der Clusterdefinitionen, die dem Warteschlangenmanager zur Verfügung stehen, immer veralteter. Im Laufe der Zeit wurden Nachrichten an falsche Zieladresse übermittelt und Anwendungen schlugen fehl. Anwendungen schlugen fehl, wenn sie versuchten, Cluster-Warteschlangen oder Publikations-Topics zu öffnen, die nicht an den lokalen Warteschlangenmanager übertragen worden waren.

Sofern ein Administrator nicht nach Repository-Meldungen im Fehlerprotokoll suchte, bemerkte er möglicherweise nicht, dass die Clusterkonfiguration Probleme aufwies. Wenn der Fehler selbst über einen längeren Zeitraum nicht bemerkt wurde und der Warteschlangenmanager die Clusterzugehörigkeit nicht erneuert hat, traten noch weitere Probleme auf. Die Instabilität betraf alle Warteschlangenmanager im Cluster, und der Cluster erschien instabil.

Ab IBM WebSphere MQ 7.1 verfolgt IBM MQ einen anderen Ansatz bei der Behandlung von Fehlern in Clustern. Anstatt den Repository-Manager zu stoppen und ohne ihn weiterzumachen, 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, die zweite darin, es zu überwachen und die Behebung des Problems zurückzustellen, und die letzte Möglichkeit besteht darin, die Cluster-Fehlerbehebung wie in den Versionen vor IBM WebSphere MQ 7.1 zu verwalten.

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)
```

3. Der Warteschlangenmanager kehrt zu demselben Verhalten bei der Wiederherstellung von Clustern zurück wie vor IBM WebSphere MQ 7.1.

Sie können einen Optimierungsparameter für den Warteschlangenmanager festlegen, damit dieser auch bei gestopptem Repository-Manager ausgeführt wird.

Der Optimierungsparameter ist `TolerateRepositoryFailure`, in der Zeilengruppe `Tuningparameter` der `qm.ini`-Datei. Um das Stoppen des Warteschlangenmanagers bei gestopptem Repository-Manager zu verhindern, setzen Sie die Option `TolerateRepositoryFailure` auf `TRUE`; siehe [Abbildung 88 auf Seite 601](#).

Starten Sie den Warteschlangenmanager erneut, um die Option `TolerateRepositoryFailure` zu aktivieren.

Wenn ein Clusterfehler aufgetreten ist, durch den ein erfolgreicher Start des Repository-Managers und somit auch des Warteschlangenmanagers verhindert wird, setzen Sie die Option `TolerateRepositoryFailure` auf `TRUE`, um den Warteschlangenmanager ohne den Repository-Manager zu starten.

## Besondere Hinweise

Vor IBM WebSphere MQ 7.1 haben einige Administratoren, die Warteschlangenmanager verwalten, die nicht Teil eines Clusters waren, den `amqrmfa`-Prozess gestoppt. Das Stoppen von `amqrmfa` hat keinen Einfluss auf den Warteschlangenmanager.

Wenn Sie `amqrmfa` in IBM WebSphere MQ 7.1 stoppen oder später stoppen, wird der Warteschlangenmanager gestoppt, da er dies als Warteschlangenmanager-Fehler angesehen wird. Sie dürfen den `amqrmfa`-Prozess in 7.1 oder höher nicht stoppen, es sei denn, Sie haben den Optimierungsparameter für den Warteschlangenmanager festgelegt, `TolerateRepositoryFailure`.

### Beispiel

```
TuningParameters:  
  TolerateRepositoryFailure=TRUE
```

Abbildung 88. Setzen Sie die Option `TolerateRepositoryFailure` in der Datei `'qm.ini'` auf `TRUE`

### Zugehörige Informationen

[Warteschlangenmanagerkonfigurationsdateien, qm.ini](#)

## JMS-Ressourcen konfigurieren

Eine der Möglichkeiten für JMS-Anwendungen, die zum Herstellen einer Verbindung zu IBM MQ und für den Zugriff auf Ziele zum Senden oder Empfangen von Nachrichten erforderlichen Ressourcen zu erstellen und zu konfigurieren, besteht darin, über die Java Naming and Directory Interface (JNDI) verwaltete Objekte aus einer Position innerhalb des Namens- und Verzeichnisservice abzurufen, die 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

Sie können verwaltete Objekte mit einem der folgenden Tools in IBM MQ erstellen und konfigurieren:

#### IBM MQ Explorer

Über IBM MQ Explorer können Sie JMS-Objektdefinitionen erstellen und verwalten, die in LDAP, einem lokalen Dateisystem oder anderen Positionen gespeichert sind.

#### IBM MQ -JMS-Verwaltungstool

Das IBM MQ-JMS-Verwaltungstool ist ein Befehlszeilentool, mit dem Sie IBM MQ-JMS-Objekte erstellen und konfigurieren können, die in LDAP, einem lokalen Dateisystem oder anderen Positionen gespeichert sind. Das JMS-Verwaltungstool verwendet eine ähnliche Syntax wie `runmqsc` und unterstützt ebenfalls Scripting.

Das Verwaltungstool verwendet 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 [„JMS-Verwaltungstool konfigurieren“](#) auf Seite 608.

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.

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 Informationen**

[Anwendungen der IBM MQ classes for JMS schreiben runmqsc](#)

## **Verbindungsfactorys und Ziele in einem JNDI-Namensbereich konfigurieren**

JMS-Anwendungen greifen über die Java Naming and Directory Interface (JNDI) auf verwaltete Objekte im Namens- und Verzeichnisservice zu. Die von JMS verwalteten Objekte werden in einer Position innerhalb des Namens- und Verzeichnisservice gespeichert, die als JNDI-Namensbereich bezeichnet wird. Eine JMS-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-Anwendungen suchen die Namen der JMS-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, der den Ausgangspunkt bildet, von dem aus eine JMS-Anwendung die Namen der verwalteten Objekte in 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.

Sie können Kontexte mit IBM MQ Explorer oder über das IBM MQ-JMS-Verwaltungstool erstellen.

Bevor eine IBM MQ classes for JMS-Anwendung verwaltete Objekte aus einem JNDI-Namensbereich abrufen kann, müssen Sie die verwalteten Objekte zunächst mit IBM MQ Explorer oder über das IBM MQ-JMS-Verwaltungstool erstellen. Sie können die folgenden Arten von JMS-Objekten erstellen und konfigurieren:

#### **Verbindungsfactory**

Ein JMS-Verbindungsfactoryobjekt definiert eine Reihe von Standardkonfigurationseigenschaften für Verbindungen. Eine JMS-Anwendung verwendet eine Verbindungsfactory, um eine Verbindung zu IBM MQ herzustellen. 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.

## Ziel

Ein JMS-Ziel ist ein Objekt, welches das Ziel von Nachrichten darstellt, die vom Client erstellt werden, und die Quelle von Nachrichten, die von einer JMS-Anwendung verarbeitet werden. Die JMS-Anwendung kann entweder ein einzelnes Zielobjekt verwenden, um Nachrichten darin einzureihen und daraus abzurufen, oder sie kann separate Zielobjekte verwenden. Es gibt zwei Typen von Zielobjekten:

- JMS-Warteschlangenziel, das beim Punkt-zu-Punkt-Messaging verwendet wird
- JMS-Topic-Ziel, das beim Publish/Subscribe-Messaging verwendet wird

Das folgende Diagramm zeigt ein Beispiel für JMS-Objekte, die in einem IBM MQ-JNDI-Namensbereich erstellt wurden.

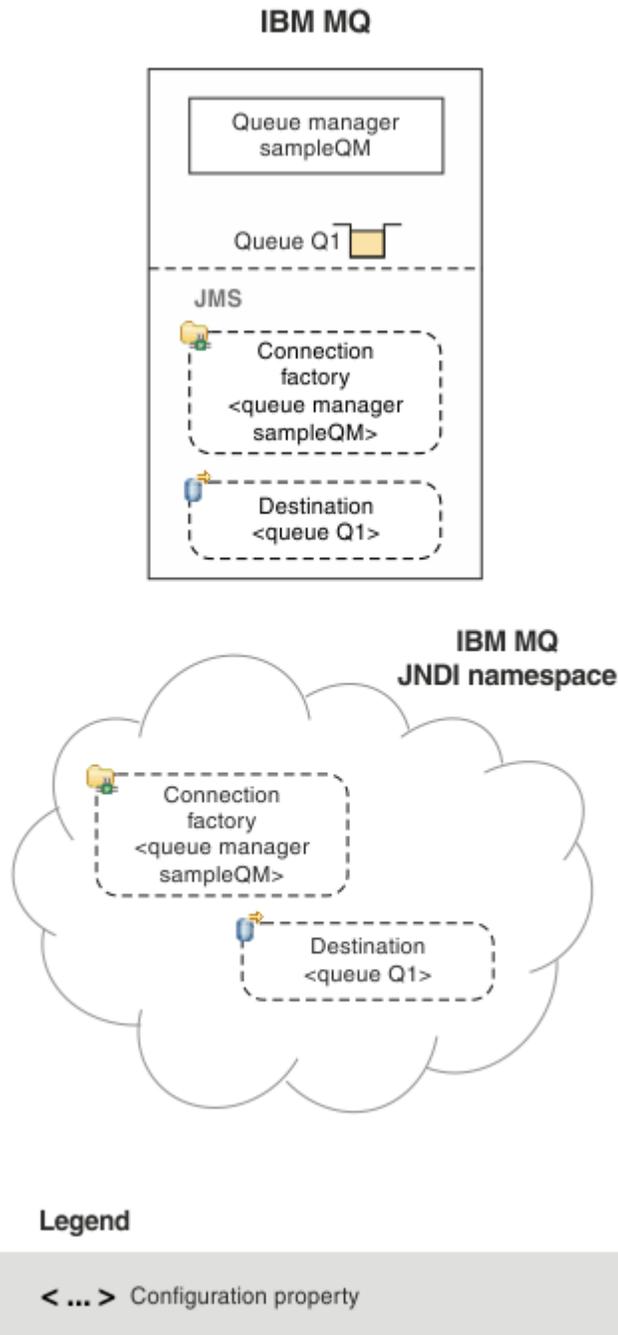


Abbildung 89. In IBM MQ erstellte JMS-Objekte

Wenn Sie JMS-Messaging zwischen WebSphere Application Server und IBM MQ verwenden, müssen Sie entsprechende Objekte in WebSphere Application Server erstellen, die Sie für die Kommunikation mit IBM MQ verwenden. 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.

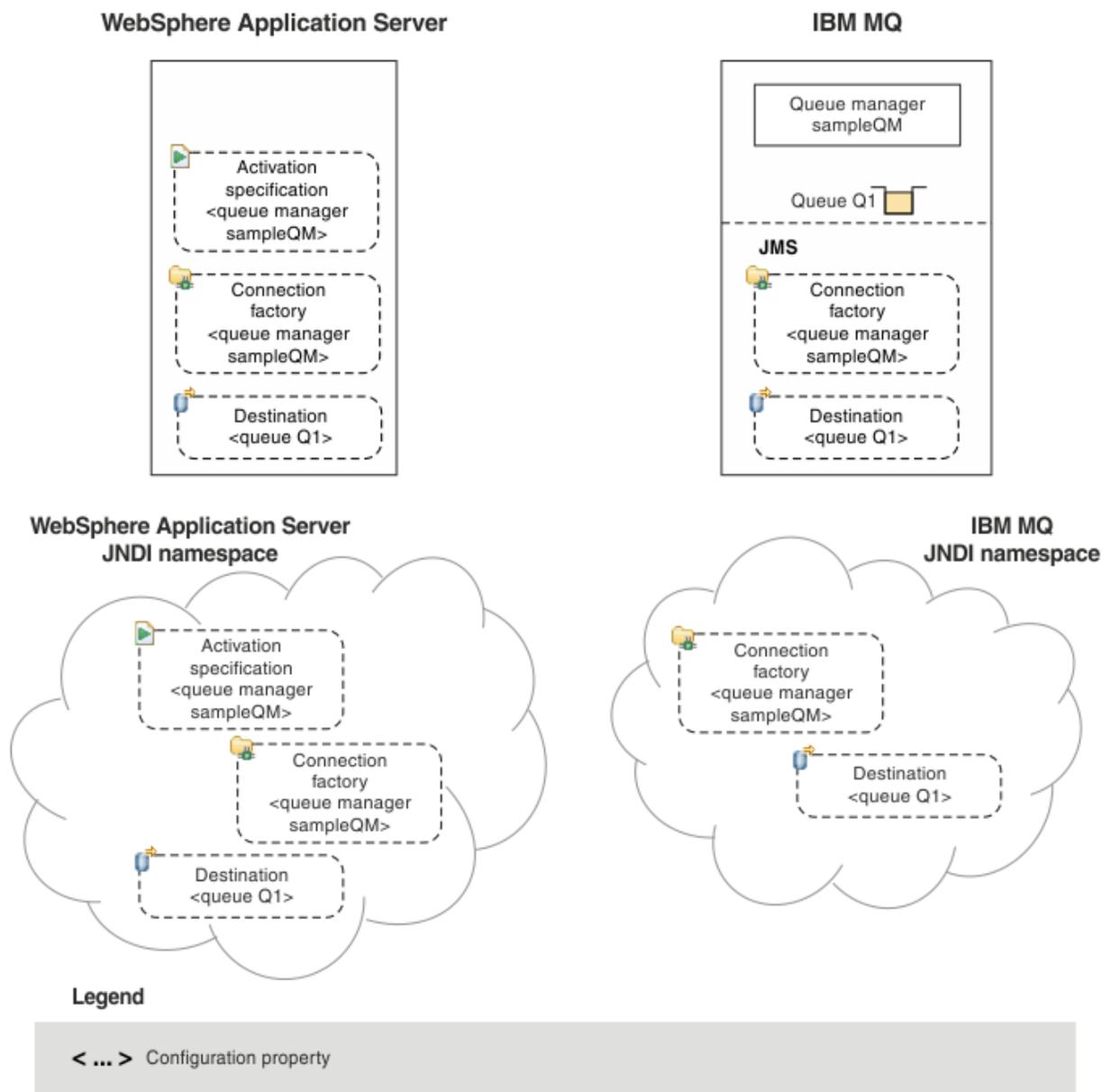


Abbildung 90. In WebSphere Application Server erstellte Objekte und die entsprechenden Objekte in IBM MQ

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.

Zum Erstellen und Konfigurieren der erforderlichen JMS-Ressourcen können Sie entweder die Administrationskonsole von WebSphere Application Server oder wsadmin-Scripting-Befehle verwenden.

## Prozedur

- Informationen zur Konfiguration von JMS-Objekten für IBM MQ mithilfe von IBM MQ Explorer finden Sie im Abschnitt „JMS-Objekte mit IBM MQ Explorer konfigurieren“ auf Seite 605.
- Informationen zur Konfiguration von JMS-Objekten für IBM MQ mit dem IBM MQ-JMS-Verwaltungstool finden Sie im Abschnitt „JMS-Objekte mit dem Verwaltungstool konfigurieren“ auf Seite 606.
- Informationen zur Konfiguration von JMS-Objekten für WebSphere Application Server finden Sie im Abschnitt „JMS-Ressourcen in WebSphere Application Server konfigurieren“ auf Seite 616.

## Ergebnisse

Eine IBM MQ classes for JMS-Anwendung kann die verwalteten Objekte mithilfe der IBM JMS-Erweiterungen oder mithilfe der IBM MQ-JMS-Erweiterungen aus dem JNDI-Namensbereich abrufen und, falls erforderlich, eine oder mehrere ihrer Eigenschaften festlegen oder ändern.

### Zugehörige Informationen

[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-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

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 kann unter Windows und Linux x86-64 ausgeführt werden. 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.

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 Informationen

[Einführung in IBM MQ Explorer](#)

[Verbindungsfactorys und Ziele in einer Anwendung der IBM MQ classes for JMS erstellen und konfigurieren](#)

## JMS-Objekte mit dem Verwaltungstool konfigurieren

Sie können das Verwaltungstool von IBM MQ JMS verwenden, um die Eigenschaften von acht Objekttypen der IBM MQ classes for JMS zu definieren und sie in einem JNDI-Namespace zu speichern. Anwendungen können dann JNDI verwenden, um diese verwalteten Objekte aus dem Namespace abzurufen.

### Informationen zu diesem Vorgang

In der folgenden Tabelle werden die acht Typen von verwalteten Objekten angezeigt, die Sie mit Verben erstellen, konfigurieren und bearbeiten können. In der Spalte Schlüsselwort werden die Zeichenfolgen angezeigt, die Sie für *TYPE* in den in [Tabelle 33 auf Seite 606](#) gezeigten Befehlen verwenden können.

<b>Objekttyp</b>	<b>Schlüsselwort</b>	<b>Beschreibung</b>
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.

Tabelle 33. JMS-Objekttypen, die vom Verwaltungstool bearbeitet werden (Forts.)

Objekttyp	Schlüsselwort	Beschreibung
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 607	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 607	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 607	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 615.

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 612 beschrieben.

Sie können verwaltete JMS-Objekte auch mit dem IBM MQ Explorer erstellen und konfigurieren.

**Zugehörige Informationen**

[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](#)

## JMS-Verwaltungstool konfigurieren

Das Verwaltungstool von IBM MQ JMS verwendet eine Konfigurationsdatei, um die Werte bestimmter Eigenschaften festzulegen. Es wird eine Beispielformatkonfigurationsdatei zur Verfügung gestellt, die Sie bearbeiten können, um sie an Ihr System anzupassen.

### Informationen zu diesem Vorgang

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 Beispielformatkonfigurationsdatei, die als Standardkonfigurationsdatei dient, wird mit IBM MQ bereitgestellt. Die Musterdatei wird als `JMSAdmin.config` bezeichnet und befindet sich im Verzeichnis `MQ_JAVA_INSTALL_PATH/bin`. Sie können diese Musterdatei bearbeiten, um die für Ihr System erforderlichen Einstellungen zu definieren oder eine eigene Konfigurationsdatei zu erstellen.

Wenn Sie das Verwaltungstool starten, können Sie die Konfigurationsdatei angeben, die Sie mit dem Befehlszeilenparameter `-cfg` verwenden möchten, wie in „[Verwaltungstool starten](#)“ auf Seite 610 beschrieben. Wenn Sie beim Aufrufen des Tools keinen Konfigurationsdateinamen angeben, versucht das Tool, die Standardkonfigurationsdatei (`JMSAdmin.config`) zu laden. Es sucht zuerst im aktuellen Verzeichnis nach dieser Datei und anschließend im Verzeichnis `MQ_JAVA_INSTALL_PATH/bin`, wobei `MQ_JAVA_INSTALL_PATH` für den Pfad zu Ihrer IBM MQ classes for JMS-Installation steht.

Die Namen von JMS-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.

**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)
-  `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 drei Parameter verwendet werden, die in der JMSAdmin-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/kontextname
- 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 Sicherheitsberechtigungs-nachweise 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 Sicherheitsberechtigungs-nachweise an den Service-Provider aus, und die *anonyme Authentifizierung* wird ausgeführt.

#### **simple (einfache Authentifizierung)**

Wenn Sie den Parameter auf `simple` setzen, werden die Sicherheitsberechtigungs-nachweise über JNDI an den zugrunde liegenden Service-Provider übergeben. Diese Sicherheitsberechtigungs-nachweise 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 Sicherheitsberechtigungs-nachweise über JNDI an den zugrunde liegenden Service-Provider übergeben. Diese Sicherheitsberechtigungs-nachweise 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 Sicherheitsberechtigungs-nachweise 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](#)).

## Verwaltungstool starten

Das Verwaltungstool verfügt über eine Befehlszeilenschnittstelle, die Sie interaktiv verwenden können, oder um einen Stapelprozess zu starten.

## 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:

```
JMSAdmin [-t] [-v] [-cfg config_filename]
```

Dabei gilt:

### -t

Aktiviert den Trace (der Standardwert ist 'trace off').

Die Tracedatei wird in "%MQ\_JAVA\_DATA\_PATH%\errors (Windows) oder /var/mqm/trace (UNIX) generiert. Der Name der Tracedatei hat das folgende Format:

```
mjms_PID.trc
```

Dabei steht *PID* für die Prozess-ID der JVM.

### -v

Generiert eine ausführliche Ausgabe (Standardeinstellung ist 'terse output')

### -cfg Konfigurationsdateiname

Benennt eine alternative Konfigurationsdatei. Wenn dieser Parameter nicht angegeben wird, wird die Standardkonfigurationsdatei `JMSAdmin.config` verwendet. Weitere Informationen zur Konfigurationsdatei finden Sie in „[JMS-Verwaltungstool konfigurieren](#)“ auf Seite 608.

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 in „[JMS-Verwaltungstool konfigurieren](#)“ auf Seite 608.

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:

```
JMSAdmin test.scp
```

Hierbei steht `test.scp` für eine Scriptdatei, die Verwaltungsbefehle enthält. Weitere Informationen finden Sie im Abschnitt „[Verwaltungsbefehle verwenden](#)“ auf Seite 611. Der letzte Befehl in der Datei muss der Befehl END sein.

## Verwaltungsbefehle verwenden

Das Verwaltungstool akzeptiert Befehle, die aus einem Verwaltungsverb und seinen entsprechenden Parametern bestehen.

### Informationen zu diesem Vorgang

In der folgenden Tabelle sind die Verwaltungsverbs aufgelistet, die Sie beim Eingeben von Befehlen mit dem Verwaltungstool verwenden können.

Verb	Kurzform	Beschreibung
ALTER	ALT	Ändern Sie mindestens eine der Eigenschaften eines verwalteten Objekts.
DEFINIER	DEF	Ein verwaltetes Objekt erstellen und speichern oder einen Subkontext erstellen
ANZEIGEN	DIS	Zeigt die Eigenschaften eines oder mehrerer gespeicherter verwalteter Objekte oder den Inhalt des aktuellen Kontexts an.
LÖSCHEN	DEL	Entfernt ein oder mehrere verwaltete Objekte aus dem Namensbereich oder entfernt einen leeren Subkontext.
ÄNDERUNG	CHG	Ändern Sie den aktuellen Kontext, sodass der Benutzer den Verzeichnisnamensbereich an einer beliebigen Stelle unterhalb des Ausgangskontextes durchqueren kann (anstehende Sicherheitsfreigabe)
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 verwaltetes Objekt gespeichert wird
END		Verwaltungstool schließen

### Prozedur

- Wenn das Verwaltungstool noch nicht gestartet wurde, starten Sie es wie im Abschnitt „[Verwaltungstool starten](#)“ auf Seite 610 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 612 beschrieben.

- Geben Sie Befehle in das folgende Format ein:

```
verb [param]*
```

Dabei steht **Verb** für eines der Verwaltungsverben, die in Tabelle 34 auf Seite 611 aufgelistet sind. 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 Eingabetaste 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 das Verb **END**, um das Verwaltungstool zu schließen. Dieses Verb kann keine Parameter verwenden.

## Subkontexte konfigurieren

Sie können die Verben **CHANGE**, **DEFINE**, **DISPLAY** und **DELETE** verwenden, um Subkontexte für Verzeichnismensbereiche zu konfigurieren.

## Informationen zu diesem Vorgang

Die Verwendung dieser Verben ist in der folgenden Tabelle beschrieben.

<i>Tabelle 35. Syntax und Beschreibung der Befehle, die zum Bearbeiten von Subkontexten verwendet werden</i>	
<b>Befehlssyntax</b>	<b>Beschreibung</b>
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)	<p>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:</p> <p><b>= AKTIV</b> wird in den übergeordneten Kontext des aktuellen Kontexts verschoben</p> <p><b>= INIT</b> wird direkt in den Ausgangskontext verschoben</p> <p>Es ist fehlgeschlagen, wenn der angegebene Kontext nicht vorhanden ist oder wenn ein Sicherheitsverstoß vorliegt.</p>

Die Namen von JMS-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 in „JMS-Verwaltungstool konfigurieren“ auf Seite 608.

**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

Um JMS-Verbindungsfactory- und Zielobjekte zu erstellen und sie in einem JNDI-Namespaces zu speichern, verwenden Sie das Verb `DEFINE`. Um Ihre Objekte in einer LDAP-Umgebung 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 verwaltetes Objekt mit dem Typ, dem Namen und den Eigenschaften, die Sie angeben. Das neue Objekt wird im aktuellen Kontext gespeichert.

Die Namen von JMS-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 in „JMS-Verwaltungstool konfigurieren“ auf Seite 608.

**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 „[Verwaltungstool starten](#)“ auf Seite 610 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 612 beschrieben.

3. Verwenden Sie die folgende Befehlsyntax, 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 Befehlsyntax:

```
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.

```
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 bei der Suche nach verwalteten JNDI-Objekten aus einer JMS-Anwendung in Windows auftreten können.

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 36 auf Seite 615 fasst die Verwendung dieser Verben zusammen. Ersetzen Sie *TYPE* durch das Schlüsselwort, das für das erforderliche verwaltete Objekt steht, wie in „JMS-Objekte mit dem Verwaltungstool konfigurieren“ auf Seite 606 beschrieben.

Befehlsyntax	Beschreibung
ALTER <i>TYPE</i> (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 <i>TYPE</i> (Name) [ Merkmal] *	Es wird versucht, ein verwaltetes Objekt des Typs <i>TYPE</i> mit den angegebenen Eigenschaften zu erstellen und unter dem Namen <i>name</i> 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 36. Syntax und Beschreibung der Befehle, die zum Bearbeiten von verwalteten Objekten verwendet werden (Forts.)

Befehlssyntax	Beschreibung
DISPLAY <i>TYPE</i> (Name)	Zeigt die Eigenschaften des verwalteten Objekts des Typs <i>TYPE</i> an, das unter dem Namen <i>name</i> im aktuellen Kontext gebunden ist. Gibt an, ob das Objekt nicht vorhanden ist, oder wenn ein Sicherheitsverstoß vorliegt.
DELETE <i>TYPE</i> (Name)	Es wird versucht, das verwaltete Objekt des Typs <i>TYPE</i> mit dem Namen <i>name</i> aus dem aktuellen Kontext zu entfernen. Gibt an, ob das Objekt nicht vorhanden ist, oder wenn ein Sicherheitsverstoß vorliegt.
COPY <i>TYPE</i> (NameA) <i>TYPE</i> (NameB)	Erstellt eine Kopie des verwalteten Objekts vom Typ <i>TYPE</i> mit dem Namen <i>nameA</i> und benennt die Kopie <i>nameB</i> . Dies geschieht im Rahmen des aktuellen Kontexts. Fehlerhafte Objekte, wenn das zu kopierende Objekt nicht vorhanden ist, wenn ein Objekt mit dem Namen <i>nameB</i> vorhanden ist oder wenn ein Sicherheitsverstoß vorliegt.
MOVE <i>TYPE</i> (NameA) <i>TYPE</i> (NameB)	Verschiebt das verwaltete Objekt des Typs <i>TYPE</i> mit dem Namen <i>nameA</i> in <i>nameB</i> (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 <i>nameB</i> vorhanden ist oder wenn ein Sicherheitsverstoß vorliegt.

## JMS-Ressourcen in WebSphere Application Server konfigurieren

Um JMS-Ressourcen in WebSphere Application Server zu konfigurieren, können Sie entweder die Administrationskonsole oder wsadmin-Befehle verwenden.

### Informationen zu diesem Vorgang

Java Message Service (JMS)-Anwendungen stützen sich in der Regel auf extern konfigurierte Objekte, die beschreiben, wie die Anwendung eine Verbindung zu ihrem JMS-Provider und zu den Zielen herstellt, auf die sie zugreift. 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 (mit JMS 1.1, domänenunabhängige (einheitliche) Verbindungsfactorys werden gegenüber domänenspezifischen Warteschlangenverbindungsfactorys und Topic-Verbindungsfactorys bevorzugt.)
- 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 Server können die JMS-Messaging-Anwendungen (Java Message Service) Ihr IBM MQ-System als externen Anbieter 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.

## **Referenzinformationen für WebSphere Application Server Version 8.5.5**

[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](#)

## **Referenzinformationen für WebSphere Application Server 8.0**

[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](#)

## **Referenzinformationen für WebSphere Application Server 7.0**

[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-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. Links zu diesen Themen in der Produktdokumentation zu WebSphere Application Server 8.5.5, 8.0 und 7.0 finden Sie unter *Zugehörige Links*.

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:

- Wenn sie eine Aktivierungsspezifikation erstellen möchten, verwenden Sie den Assistenten zum Erstellen von IBM MQ JMS-Ressourcen.

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 Verbindungsnamensliste 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**

[„Definitionstabelle für den Clientkanal“ auf Seite 43](#)

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-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

[„Warteschlangenmanager mit mehreren Instanzen“ auf Seite 466](#)

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**

[„Publish/Subscribe-Messaging konfigurieren“ auf Seite 382](#)

Sie können den Status des eingereichten Publish/Subscribe starten, stoppen und anzeigen. Darüber hinaus können Sie Datenströme hinzufügen und entfernen sowie Warteschlangenmanager aus einer Brokerhierarchie hinzufügen und löschen.

### **Referenzinformationen zu WebSphere Application Server traditional 9.0**

[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](#)

### **Referenzinformationen zu WebSphere Application Server 8.5.5**

[Aktivierungsspezifikationen des IBM WebSphere MQ-Messaging-Providers](#)

[Aktivierungsspezifikation für den IBM WebSphere MQ-Messaging-Provider erstellen](#)

[Aktivierungsspezifikation für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

[Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider erstellen](#)

[Einheitliche Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

[Warteschlangenverbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

[Topic-Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

[Warteschlange für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

[Topic für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

### **Referenzinformationen zu WebSphere Application Server 8.0**

[Aktivierungsspezifikationen des IBM WebSphere MQ-Messaging-Providers](#)

[Aktivierungsspezifikation für den IBM WebSphere MQ-Messaging-Provider erstellen](#)

[Aktivierungsspezifikation für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider erstellen](#)  
[Einheitliche Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Warteschlangenverbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Topic-Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Warteschlange für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Topic für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

### **Referenzinformationen zu WebSphere Application Server 7.0**

[Aktivierungsspezifikationen des IBM WebSphere MQ-Messaging-Providers](#)  
[Aktivierungsspezifikation für den IBM WebSphere MQ-Messaging-Provider erstellen](#)  
[Aktivierungsspezifikation für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider erstellen](#)  
[Einheitliche Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Warteschlangenverbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Topic-Verbindungsfactory für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Warteschlange für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)  
[Topic für den IBM WebSphere MQ-Messaging-Provider konfigurieren](#)

## **JMS-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 bieten eine Übersicht über die Möglichkeiten, wie Sie WebSphere Application Server wsadmin-Befehle verwenden können, um JMS-Ressourcen für die Verwendung mit dem Messaging-Provider von IBM MQ konfigurieren können. Weitere Informationen zur Verwendung dieser Befehle finden Sie unter *Zugehörige Links* in der Produktdokumentation zu WebSphere Application Server 8.5.5, 8.0 und 7.0 .

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 **listWMQConnectionFactory**.
- 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 für den IBM MQ-Ressourcenadapter zu verwalten, der in einem bestimmten Bereich 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

„Definitionstabelle für den Clientkanal“ auf Seite 43

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-Plattformen wird automatisch eine CCDT erstellt. Anschließend müssen Sie sie für die Clientanwendung zur Verfügung stellen.

„Warteschlangenmanager mit mehreren Instanzen“ auf Seite 466

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

„Publish/Subscribe-Messaging konfigurieren“ auf Seite 382

Sie können den Status des eingereichten Publish/Subscribe starten, stoppen und anzeigen. Darüber hinaus können Sie Datenströme hinzufügen und entfernen sowie Warteschlangenmanager aus einer Brokerhierarchie hinzufügen und löschen.

### Referenzinformationen für WebSphere Application Server Version 8.5.5

**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](#)

#### **Referenzinformationen zu WebSphere Application Server 8.5.5**

[createWMQActivationSpec-Befehl](#)  
[createWMQConnectionFactory-Befehl](#)  
[createWMQQueue-Befehl](#)  
[createWMQTopic-Befehl](#)  
[deleteWMQActivationSpec-Befehl](#)  
[deleteWMQConnectionFactory-Befehl](#)  
[deleteWMQQueue-Befehl](#)  
[deleteWMQTopic-Befehl](#)  
[listWMQActivationSpecs-Befehl](#)  
[listWMQConnectionFactoryies-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](#)

#### **Referenzinformationen zu WebSphere Application Server 8.0**

[createWMQActivationSpec-Befehl](#)  
[createWMQConnectionFactory-Befehl](#)  
[createWMQQueue-Befehl](#)  
[createWMQTopic-Befehl](#)  
[deleteWMQActivationSpec-Befehl](#)  
[deleteWMQConnectionFactory-Befehl](#)  
[deleteWMQQueue-Befehl](#)  
[deleteWMQTopic-Befehl](#)  
[listWMQActivationSpecs-Befehl](#)  
[listWMQConnectionFactoryies-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](#)

## **Referenzinformationen zu WebSphere Application Server 7.0**

[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](#)

## **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-Ressourcen über die Administrationskonsole konfigurieren](#)“ auf Seite 617).
2. Erstellen Sie die Aktivierungsspezifikation (siehe „[JMS-Ressourcen über die Administrationskonsole konfigurieren](#)“ auf Seite 617).

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:

Eigenschaftsname	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 **Aktivierungsspezifikationsammlung** 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 Namen JNDI im Feld **Wert** verwenden, z. B. `WASSharedSubOne`.

### Typ

Der Typ der angepassten Eigenschaft. Wählen Sie den angepassten Eigenschaftstyp aus der Liste aus, der in diesem Fall `java.lang.String` sein muss.

- 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.

- 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 Informationen**

[Klone und gemeinsam genutzte Subskriptionen](#)

[Subskriptionspermanenz](#)

[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 JMS-Ressourcen des IBM MQ-Messaging-Providers konfigurieren](#)

**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 Aktivierungsspezifikationseigenschaften verwendet werden sollen.

**Informationen zu diesem Vorgang**

Die JMS 2.0-Spezifikation gibt zwei zusätzliche Eigenschaften in der verwendeten Aktivierungsspezifikation 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-Ressourcen über die Administrationskonsole konfigurieren“](#) auf Seite 617).
2. Erstellen Sie das Ziel (Destination) in JNDI wie üblich (siehe [„JMS-Ressourcen über die Administrationskonsole konfigurieren“](#) auf Seite 617).

Das Zielobjekt muss die korrekten Werte haben.

3. Erstellen Sie die Aktivierungsspezifikation mit allen erforderlichen Werten (siehe [„JMS-Ressourcen über die Administrationskonsole konfigurieren“](#) auf Seite 617).

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 Verbindungsfactory 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 627 beschrieben erstellt haben, oder verwenden Sie stattdessen Annotationen, wie in Schritt „5“ auf Seite 628 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 = "jms:
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 Informationen

[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 JMS-Ressourcen des IBM MQ-Messaging-Providers konfigurieren](#)

## Anwendungsserver für die Verwendung der neuesten Wartungsstufe des Ressourcenadapters 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:** Wenn Sie WebSphere Application Server 7.0, 8 oder 8.5 auf einer Plattform verwenden, installieren Sie den IBM MQ 8.0 -Ressourcenadapter nicht im Anwendungsserver. Der IBM MQ 8.0-Ressourcenadapter kann nur in einem Anwendungsserver implementiert werden, der JMS 2.0 unterstützt. WebSphere Application Server 7.0, 8 und 8.5 unterstützen jedoch nur JMS 1.1. Diese Versionen von WebSphere Application Server werden mit dem IBM WebSphere MQ 7.0-Ressourcenadapter, der für die Verbindung zu einem IBM MQ 8.0-Warteschlangenmanager verwendet werden kann, entweder mit dem BINDINGS- oder CLIENT-Transport verwendet.

## 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, 'getInfo', '')
```

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 Fixpack 15 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 IBM MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In IBM 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 den WebSphere Application Server (z. B. das Verzeichnis `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 Informationen**

[IBM MQ-Ressourcenadapter verwenden](#)

### **Referenzinformationen zu 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](#)

### **Referenzinformationen zu WebSphere Application Server 8.0**

[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](#)

### **Referenzinformationen zu WebSphere Application Server 7.0**

[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.

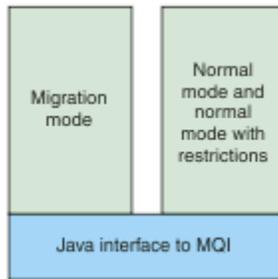


Abbildung 91. Messaging-Provider-Modi

## 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.

Falls eine Änderung der verwendeten Verbindungsfactory nicht möglich ist, können Sie Einstellungen der Verbindungsfactory mithilfe der Eigenschaft `com.ibm.msg.client.wmq.overrideProviderVersion` überschreiben. Diese Überschreibung gilt für alle Verbindungsfactorys in der Java Virtual Machine (JVM), die eigentlichen Verbindungsfactory-Objekte werden jedoch nicht geändert.

## Zugehörige Informationen

[PROVIDERVERSION](#)

[Fehlerbehebung bei JMS-Providerversionen](#)

[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

Dabei sind V, R, M und F Ganzzahlen größer oder gleich Null. 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-Objekte mit dem Verwaltungstool konfigurieren](#)“ auf Seite 606.

### Zugehörige Informationen

[IBM MQ-Klassen für JMS-Architektur](#)

### **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.

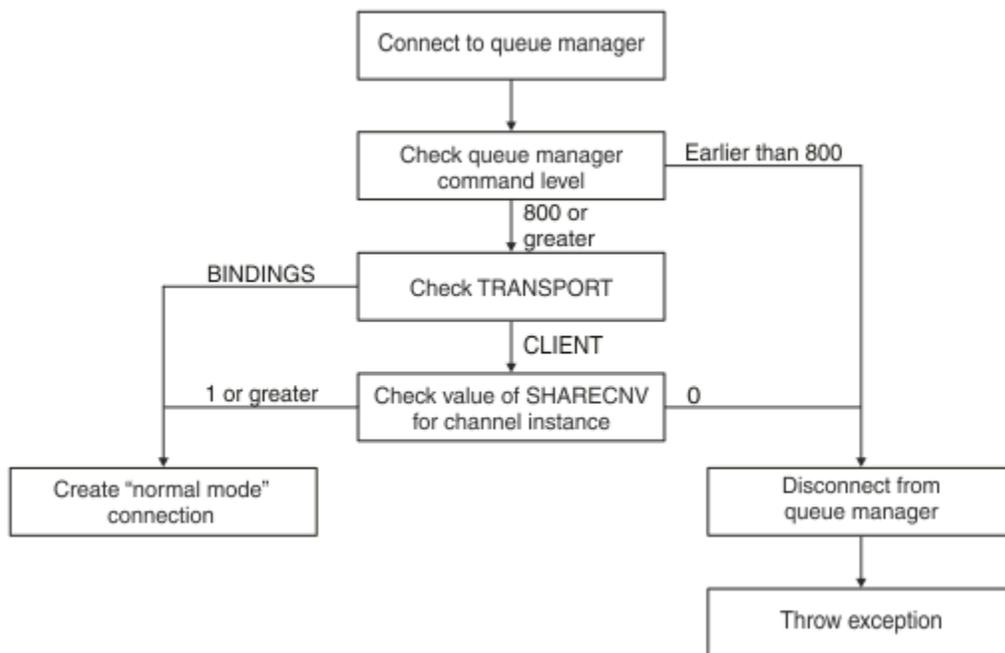


Abbildung 92. PROVIDERVERSION-Normalmodus

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 einer Ausnahmebedingung `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 Informationen

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 API von JMS 2.0, aber nicht die neuen Funktionen von IBM MQ 8.0 oder höher, wie z. B. gemeinsame Subskriptionen, verzögerte Zustellung oder asynchrone Sendebereitstellung.

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.

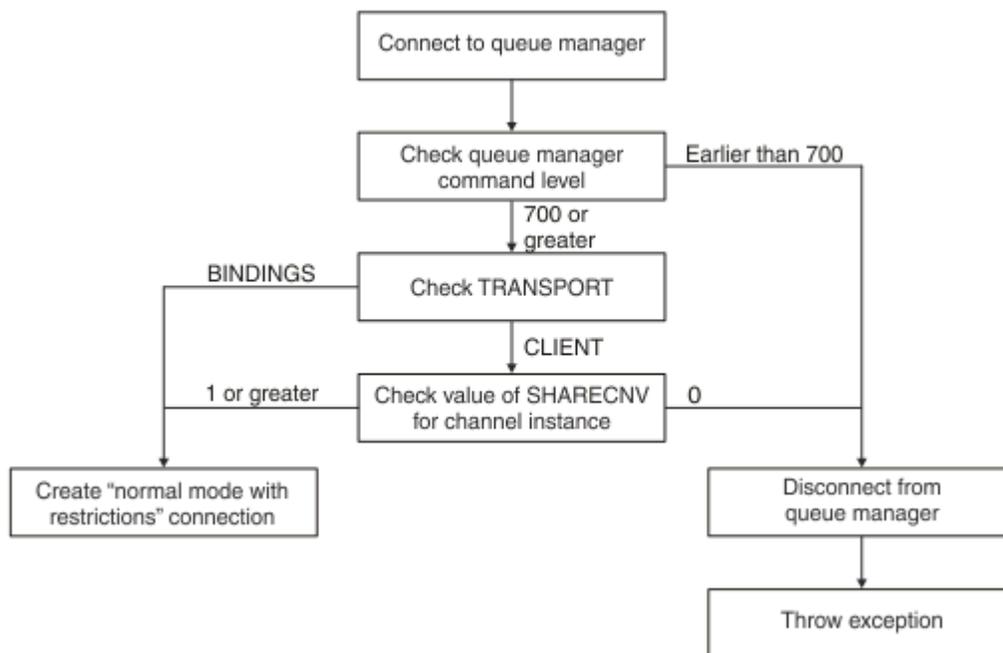


Abbildung 93. PROVIDERVERSION-normaler Modus mit Einschränkungen

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 Send-, 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 JMSSC5007 fehl.

### Zugehörige Informationen

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 eingereichtes Publish/Subscribe, auf der Clientseite implementierte Auswahl, Nicht-Multiplex-Kanäle und Polling zur Implementierung der Listener.

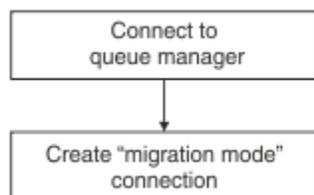


Abbildung 94. Migrationsmodus PROVIDERVERSION

Wenn Sie eine Verbindung zu WebSphere Message Broker 6.0 oder 6.1 über IBM MQ Enterprise Transport 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 Sie über einen Client von IBM MQ 8.0 oder höher verfügen, der eine Verbindung zu einem Warteschlangenmanager von IBM MQ 8.0 oder höher auf einer verteilten Plattform, **z/OS** oder einem Warteschlangenmanager von IBM MQ 8.0 oder höher unter z/OS 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 JM5CC5007 fehl.

### **Zugehörige Informationen**

Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten  
TRANSPORT

### **PROVIDERVERSION nicht angeben**

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 Verbindungseigenschaften, 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.

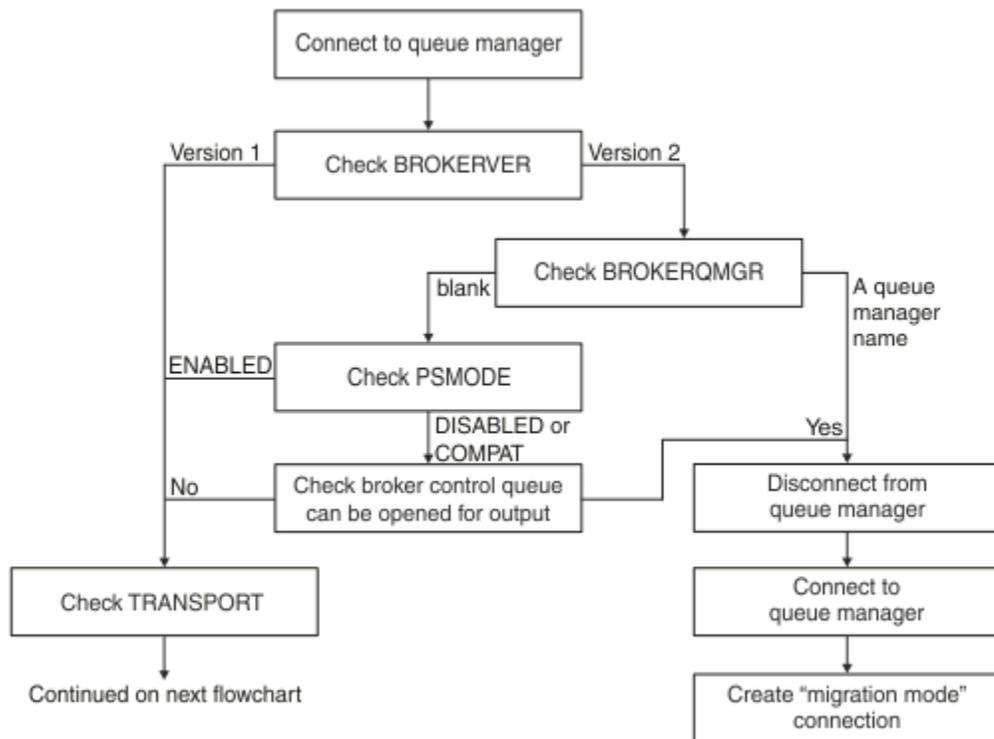


Abbildung 95. PROVIDERVERSION nicht angegeben

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.

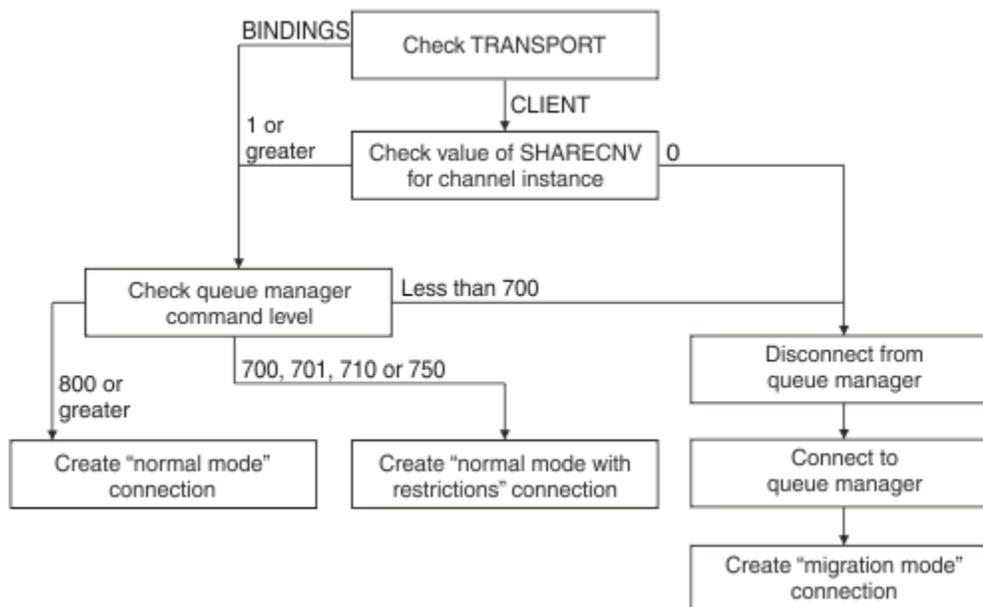


Abbildung 96. PROVIDERVERSION nicht angegeben (Fortsetzung)

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 über eine Befehlsebene von 700, 701, 710 oder 750 verfügt, 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 Informationen

[Abhängigkeiten zwischen Eigenschaften von IBM MQ classes for JMS-Objekten](#)

[ALTER QMGR \(Attribut PSMODE\)](#)

[BROKERCONQ](#)

[BROKERQMGR](#)

[BROKERVER](#)

[DEFINE CHANNEL \(Eigenschaft SHARECNV\)](#)

[TRANSPORT](#)

## Wann die Standardeinstellung für PROVIDERVERSION überschrieben werden soll

Wenn eine Verbindungsfactory, die im JNDI-Namensbereich mit einer Vorgängerversion von IBM MQ classes for JMS erstellt wurde, mit der neuen Version von IBM MQ classes for JMS verwendet wird, wird die Eigenschaft **PROVIDERVERSION** für die Verbindungsfactory auf den Standardwert `unspecified` gesetzt und mithilfe eines Algorithmus wird ermittelt, welche Betriebsart des IBM MQ-Messaging-Providers verwendet wird. Es gibt jedoch zwei Fälle, in denen Sie die Standardauswahl für die Eigenschaft **PROVIDERVERSION** außer Kraft setzen müssen, damit die IBM MQ classes for JMS ordnungsgemäß funktionieren können.

**Anmerkung:** Der in diesem Abschnitt beschriebene Migrationsmodus bezieht sich auf die Migration von IBM WebSphere MQ 6.0 auf 7.0. Er gilt nicht für die Migration von höheren Releases.

IBM WebSphere MQ 6.0, WebSphere Application Server 6.0.x und WebSphere Message Broker 6 werden nicht mehr unterstützt, daher ist dieser Abschnitt nur zu Referenzzwecken enthalten.

Wenn die Eigenschaft **PROVIDERVERSION** auf den Standardwert `unspecified` gesetzt ist, wird mithilfe eines Algorithmus festgestellt, welche Betriebsart verwendet werden soll, wie im Abschnitt „PROVIDERVERSION nicht angegeben“ auf Seite 636 beschrieben. Sie können diesen Algorithmus jedoch in den folgenden beiden Szenarios nicht verwenden.

1. Wenn sich WebSphere Message Broker und WebSphere Event Broker im Kompatibilitätsmodus befinden, müssen Sie einen Wert für die Eigenschaft **PROVIDERVERSION** angeben, damit WebSphere Message Broker und WebSphere Event Broker ordnungsgemäß funktionieren.
2. Wenn Sie WebSphere Application Server 6.0.1, 6.0.2 oder 6.1 verwenden, werden Verbindungsfactorys über die WebSphere Application Server -Administrationskonsole definiert.

In WebSphere Application Server ist der Standardwert der Eigenschaft **BROKERVER** in einer Verbindungsfactory V2. Der Standardwert für die Eigenschaft **BROKERVER** für Verbindungsfactorys, die mit dem JMS -Verwaltungstool **JMSAdmin** oder IBM MQ Explorer erstellt werden, ist V1. Diese Eigenschaft ist jetzt in IBM MQ auf `unspecified` (nicht angegeben) gesetzt.

Wenn die Eigenschaft **BROKERVER** auf V2 gesetzt ist, weil sie entweder von WebSphere Application Server erstellt wurde oder weil die Verbindungsfactory für Publish/Subscribe verwendet wurde und einen vorhandenen Warteschlangenmanager hat, für den eine Eigenschaft **BROKERCONQ** definiert ist (weil sie bereits für Publish/Subscribe-Messaging verwendet wurde), wird der Migrationsmodus des IBM MQ -Messaging-Providers verwendet.

Wenn Sie jedoch möchten, dass die Anwendung die Peer-to-Peer-Kommunikation verwendet und die Anwendung einen vorhandenen Warteschlangenmanager verwendet, der jemals für Publish/Subscribe verwendet wurde, und eine Verbindungsfactory mit **BROKERVER** auf 2 gesetzt ist (dies ist die Standardeinstellung, wenn die Verbindungsfactory in WebSphere Application Server erstellt wurde), wird der Migrationsmodus des IBM MQ -Messaging-Providers verwendet. In diesem Fall ist es nicht erforderlich, den Migrationsmodus des IBM MQ-Messaging-Providers zu verwenden. Verwenden Sie stattdessen den normalen Modus des IBM MQ-Messaging-Providers. Sie können eine der folgenden Methoden verwenden, um diese zu bearbeiten:

- Setzen Sie **BROKERVER** auf 1 oder nicht angegeben. Die Option, die Sie auswählen, hängt von Ihrer Anwendung ab.
- Setzen Sie **PROVIDERVERSION** auf 8 oder 7. Dies sind benutzerdefinierte Eigenschaften in WebSphere Application Server 6.1.

Alternativ können Sie die Clientkonfigurationseigenschaft verwenden oder den Warteschlangenmanager so ändern, dass er nicht über die **BROKERCONQ** -Eigenschaftengruppe verfügt, oder die Warteschlange unbrauchbar machen.

## 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

Befehl **createWMQConnectionFactory**

Einstellungen der Aktivierungsspezifikation des IBM MQ-Messaging-Providers

Befehl **createWMQActivationSpec**

**Referenzinformationen für WebSphere Application Server 8.0.0**

Einstellungen für die Verbindungsfactory des IBM MQ-Messaging-Providers

Befehl **createWMQConnectionFactory**

Einstellungen der IBM MQ-Aktivierungsspezifikation

Befehl **createWMQActivationSpec**

**Referenzinformationen für WebSphere Application Server 7.0.0**

Einstellungen für die Verbindungsfactory des IBM MQ-Messaging-Providers

Befehl **createWMQConnectionFactory**

Einstellungen der IBM MQ-Aktivierungsspezifikation

Befehl **createWMQActivationSpec**

## Permanente WebSphere Application Server-Subskriptionen entfernen

Wenn Sie den IBM MQ -Messaging-Provider mit WebSphere Application Server 7.0 und 8.0 verwenden, werden permanente Subskriptionen, die von an Aktivierungsspezifikationen gebundenen MDB-Anwendungen erstellt wurden, nicht entfernt. Permanente Subskriptionen können entweder mit dem IBM MQ Explorer oder einem IBM MQ-Befehlszeilendienstprogramm entfernt werden.

### Informationen zu dieser Task

Eine Message-driven Bean-Anwendung, die eine permanente Subskription entfernt, kann so konfiguriert werden, dass sie entweder einen Listener-Port oder eine Aktivierungsspezifikation verwendet, vorausgesetzt, die Anwendung wird in einer Instanz von WebSphere Application Server 7.0 oder 8.0 ausgeführt, die WebSphere MQ -Messaging-Provider im normalen Modus für die Verbindung zu IBM MQ verwendet.

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 "verwaissere 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.

### Bevor Sie beginnen

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 IBM MQ Explorer. Wenn IBM MQ Explorer mit einem Warteschlangenmanager verbunden wurde, der für Publish/Subscribe-Arbeit verwendet wird, können Sie die Liste der Subskribenten, die derzeit bei der Publish/Subscribe-Engine registriert sind, anzeigen, indem Sie im Navigationsfenster auf den Eintrag IBM WebSphere MQ ->queue manager name->Subscriptions klicken.
- 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

Sobald eine "verwaiste" permanente Subskription identifiziert wurde, kann sie mit IBM MQ Explorer oder mit dem IBM MQ Befehlszeilendienstprogramm **runmqsc** gelöscht werden.

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 *WebSphere 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`

## V 9.0.1 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

- [„Sicherheit konfigurieren“ auf Seite 643](#)
- [„Konfigurieren des HTTP-Host-Namens“ auf Seite 645](#)
- [„HTTP- und HTTPS-Ports konfigurieren“ auf Seite 646](#)
- [„Konfigurieren des Antwortzeitlimits“ auf Seite 647](#)
- [„Autostart konfigurieren“ auf Seite 649](#)
- [„Protokollierung konfigurieren“ auf Seite 650](#)
- [„Ablaufintervall für LTPA-Token“ auf Seite 653](#)
- [„messaging REST API konfigurieren“ auf Seite 655](#)
- [„CSRF-Schutz konfigurieren“ auf Seite 644](#)

## V 9.0.1 Sicherheit konfigurieren

Sie können die Sicherheit für die IBM MQ Console und die REST API konfigurieren, indem Sie die Datei `mqwebuser.xml` bearbeiten. Zur Konfiguration und Authentifizierung von Benutzern können Sie entweder eine Basisbenutzerregistry oder eine LDAP-Registry oder einen der anderen mit WebSphere Application Server Liberty bereitgestellten Registry-Typen konfigurieren. Anschließend können Sie diese Benutzer berechtigen, indem Sie Benutzer und Gruppen einer Rolle zuordnen. Bei IBM MQ 9.0.1 gibt es keine Sicherheitsfunktion für die REST API. Ab IBM MQ 9.0.2 können Sie die Sicherheit für die REST API konfigurieren.

### 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 sowie zur Authentifizierung und Autorisierung von Benutzern finden Sie im Abschnitt [Sicherheit der IBM MQ Console und REST API](#).

Wenn sich Benutzer bei der IBM MQ Console authentifizieren, wird ein LTPA-Token generiert. Wenn Sie die tokenbasierte Authentifizierung bei der REST API verwenden, wird ein anderes LTPA-Token generiert, wenn sich der Benutzer unter Verwendung der `/login`-REST API-Ressource mit der HTTP-POST-Methode anmeldet. Mit diesem Token kann der Benutzer die IBM MQ Console bis zum Ablauf des Tokens ohne erneute Authentifizierung verwenden. Sie können konfigurieren, wann das Token abläuft. Weitere Informationen finden Sie im Abschnitt [„Ablaufintervall für LTPA-Token“ auf Seite 653](#).

### Prozedur

- [Sicherheit der IBM MQ Console und REST API](#)
- [„Ablaufintervall für LTPA-Token“ auf Seite 653](#)

## V 9.0.4 CSRF-Schutz konfigurieren

Cross-Site Request Forgery (CSRF) ist eine Art von Angriff, der auftritt, wenn eine zerstörerische Website den Browser eines Benutzers dazu veranlasst, eine unerwünschte Aktion auf einer vertrauenswürdigen Site auszuführen, für die der Benutzer derzeit authentifiziert ist.

### Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

V 9.0.4 Sie können die aktuelle Konfiguration des CSRF-Schutzes mit dem folgenden Befehl anzeigen:

```
dspmweb properties -a
```

Das Feld `mqRestCsrftValidation` zeigt an, ob CSRF-Gültigkeitsprüfungen durchgeführt werden. Weitere Informationen finden Sie in [dspmweb](#).

**Anmerkung:** V 9.0.5 Das Feld `mqRestCsrftExpirationInMinutes`, das in IBM MQ 9.0.4 eingeführt wurde, um die CSRF-Ablaufzeit anzuzeigen, ist in IBM MQ 9.0.5 nicht mehr vorhanden.



**Achtung:** z/OS V 9.0.4

Bevor Sie den Befehl **setmqweb** oder **dspmweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre `mqweb`-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist `WLP_user_directory` der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

### Informationen zu diesem Vorgang

V 9.0.5 Vor IBM MQ 9.0.5 verwenden die IBM MQ Console und REST API ein Synchronisationstoken zum Schutz vor CSRF-Attacks. Nur in IBM MQ 9.0.4 werden die CSRF-Synchronisationstokens für die administrative REST API in regelmäßigen Abständen neu generiert. Ab IBM MQ 9.0.5 werden keine CSRF-Synchronisationstokens verwendet. Stattdessen muss ein angepasster HTTP-Header festgelegt werden, der einen äquivalenten Schutz für die Verwendung eines Synchronisationstokens bietet.

Sie können die Konfiguration des CSRF-Schutzes für REST API mit dem Befehl **setmqweb properties** ändern.

### Prozedur

- Verwenden Sie die folgende Methode, um die CSRF-Tokenvalidierung für die REST API zu konfigurieren:
- Nur IBM MQ 9.0.4: Verwenden Sie den Befehl **setmqweb properties**, um den Tokenablauf zu ändern:

```
setmqweb properties -k mqRestCsrftExpirationInMinutes -v time
```

Dabei gibt `time` die Zeit (in Minuten) an, bevor das CSRF-Token abläuft. Das Token bleibt für die nächste HTTP-POST-, PATCH- oder DELETE-Methode nach Ablauf des Ablaufs gültig, danach wird ein neues Token als Cookie zurückgegeben und der vorherige Tokenwert ungültig gemacht. Der

Zeitwert -1 inaktiviert die Verfallszeit des CSRF-Tokens, während der Wert 0 bewirkt, dass das Token bei jeder POST-, PATCH- oder DELETE-Anforderung geändert wird. Der Standardwert ist 30 Minuten.

- Verwenden Sie den Befehl **setmqweb properties**, um CSRF-Gültigkeitsprüfungen zu entfernen:

```
setmqweb properties -k mqRestCsrfValidation -v boolean
```

Dabei gibt *boolean* an, ob CSRF-Gültigkeitsprüfungen ausgeführt werden. Bei einem Wert von 'false' werden die Gültigkeitsprüfungen für CSRF-Token entfernt. Es wird empfohlen, Tokens zu validieren, insbesondere dann, wenn Benutzer Webbrowser für den Zugriff auf die REST API verwenden. Der Standardwert ist 'true' und CSRF-Tokens werden für alle HTTP-POST-, PATCH- und DELETE-Anforderungen über die REST API geprüft.

## V 9.0.1 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. Dies bedeutet, dass der Zugriff auf die IBM MQ Console und REST API nur auf dem System möglich ist, auf dem die IBM MQ Console und REST API installiert sind.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie den Hostnamen für das Zulassen von Fernverbindungen mit dem Befehl **setmqweb** konfigurieren. In IBM MQ 9.0.3 und früher können Sie den Hostnamen so konfigurieren, dass Fernverbindungen zugelassen werden, indem Sie die Datei `mqwebuser.xml` bearbeiten.

### Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie die aktuelle Konfiguration des HTTP-Hostnamens mit folgendem Befehl anzeigen:

```
dspmweb properties -a
```

Im Feld `httpHost` wird der HTTP-Hostname angezeigt. Weitere Informationen finden Sie in [dspmweb](#).



### Achtung: z/OS V 9.0.4

Bevor Sie den Befehl **setmqweb** oder **dspmweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist `WLP_user_directory` der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

### Prozedur

- **V 9.0.4**

Verwenden Sie eine der folgenden Methoden zum Konfigurieren des Hostnamens:

- Verwenden Sie im Verzeichnis IBM MQ 9.0.4 den Befehl **setmqweb properties**:

```
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.

- Bearbeiten Sie für IBM MQ 9.0.3 und früher die Datei `mqwebuser.xml`:

1. Öffnen Sie die Datei `mqwebuser.xml`.

Die Datei `mqwebuser.xml` befindet sich in einem der folgenden Verzeichnisse:

-  Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
-  Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

2. Konfigurieren Sie den `mqweb`-Server:

- Um Fernverbindungen mit dem `mqweb`-Server zuzulassen, fügen Sie die folgende Zeile innerhalb der Tags `<server>` zur Datei `mqwebuser.xml` hinzu:

```
<variable name="httpHost" value="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 (\*), um alle verfügbaren Netzschnittstellen anzugeben.

- Um nur lokale Verbindungen zum `mqweb`-Server zuzulassen, entfernen Sie entweder die folgende Zeile aus der Datei `mqwebuser.xml`, oder setzen Sie den Wert auf `localhost`:

```
<variable name="httpHost" value="hostName" />
```

## 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.  Ab IBM MQ 9.0.4 können Sie die Ports mit dem Befehl `setmqweb` konfigurieren. In IBM MQ 9.0.3 und früher können Sie die Ports konfigurieren, indem Sie die Datei `mqwebuser.xml` bearbeiten.

### Vorbereitende Schritte

Sie müssen ein privilegiertes Benutzer sein, um diese Prozedur ausführen zu können.

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 von einem Browser wiederverwendet werden. Sie können den `mqweb`-Server konfigurieren, um dieses Verhalten zu verhindern und die Umgebung sicherer zu machen, indem Sie die folgende Zeile in die Datei `mqwebuser.xml` hinzufügen:

```
<webAppSecurity ssoRequiresSSL="true"/>
```

 Ab IBM MQ 9.0.4 können Sie die aktuelle Konfiguration der HTTP- und HTTPS-Ports anzeigen, indem Sie den folgenden Befehl verwenden:

```
dspmweb properties -a
```

Im Feld `httpPort` wird der HTTP-Port angezeigt, und im Feld `httpsPort` wird der HTTPS-Port angezeigt. Weitere Informationen finden Sie in [dspmweb](#).



### Achtung:

Bevor Sie den Befehl `setmqweb` oder `dspmweb` unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre `mqweb`-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP\_user\_directory* der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

## Prozedur

### V 9.0.4

Verwenden Sie eine der folgenden Methoden, um die Ports zu konfigurieren:

- Verwenden Sie im Verzeichnis IBM MQ 9.0.4 den Befehl **setmqweb properties** :

- Verwenden Sie den folgenden Befehl, um den HTTP-Port zu aktivieren oder zu konfigurieren:

```
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.

- Verwenden Sie den folgenden Befehl, um den HTTPS-Port zu konfigurieren:

```
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.

- Bearbeiten Sie für IBM MQ 9.0.3 und früher die Datei `mqwebuser.xml`:

1. Öffnen Sie die Datei `mqwebuser.xml` .

Die Datei `mqwebuser.xml` befindet sich in einem der folgenden Verzeichnisse:

- **ULW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script **crtmqweb.sh** ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

2. Konfigurieren Sie die Ports:

- Um den HTTP-Port zu aktivieren oder zu konfigurieren, fügen Sie die folgende Zeile in der Datei `mqwebuser.xml` innerhalb der `<server>`-Tags hinzu oder bearbeiten Sie sie:

```
<variable name="httpPort" value="portNumber" />
```

Dabei gibt *portNumber* den Port an, den Sie für HTTP-Verbindungen verwenden möchten. Sie können den Port mit einem Wert von `-1` inaktivieren.

- Um den HTTPS-Port zu konfigurieren, fügen Sie die folgende Zeile in der Datei `mqwebuser.xml` in den `<server>`-Tags hinzu oder bearbeiten Sie sie:

```
<variable name="httpsPort" value="portNumber" />
```

Dabei gibt *portNumber* den Port an, den Sie für HTTPS-Verbindungen verwenden möchten. Sie können den Port mit einem Wert von `-1` inaktivieren.

### V 9.0.1 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. **V 9.0.4**

Ab IBM MQ 9.0.4 können Sie die IBM MQ Console und REST API für die Verwendung eines anderen

Zeitlimitwerts konfigurieren, indem Sie den Befehl **setmqweb** verwenden. In IBM MQ 9.0.3 und früher können Sie IBM MQ Console und REST API so konfigurieren, dass ein anderer Zeitlimitwert verwendet wird, indem Sie die Datei `mqwebuser.xml` bearbeiten.

## Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie die aktuelle Konfiguration des REST API-Antwortzeitlimits mit folgendem Befehl anzeigen:

```
dspmweb properties -a
```

Im Feld `mqRestRequestTimeout` wird der aktuelle Wert für das Antwortzeitlimit angezeigt. Weitere Informationen finden Sie in [dspmweb](#).



**Achtung:** **z/OS** **V 9.0.4**

Bevor Sie den Befehl **setmqweb** oder **dspmweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist `WLP_user_directory` der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

## Prozedur

**V 9.0.4**

Verwenden Sie eine der folgenden Methoden, um das Zeitlimit zu konfigurieren:

- Verwenden Sie im Verzeichnis IBM MQ 9.0.4 den Befehl **setmqweb properties**:

```
setmqweb properties -k mqRestRequestTimeout -v timeout
```

Dabei gibt `timeout` die Zeit (in Sekunden) vor dem Zeitlimit an.

- Bearbeiten Sie für IBM MQ 9.0.3 und früher die Datei `mqwebuser.xml`:

1. Öffnen Sie die Datei `mqwebuser.xml`.

Die Datei `mqwebuser.xml` befindet sich in einem der folgenden Verzeichnisse:

- **ULW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

- **z/OS** Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script **crtmqweb.sh** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

2. Konfigurieren Sie das Zeitlimit, indem Sie die folgende Zeile in der Datei `mqwebuser.xml` in den `<server>`-Tags hinzufügen oder bearbeiten:

```
<variable name="mqRestRequestTimeout" value="timeout" />
```

Dabei gibt `timeout` die Zeit (in Sekunden) vor dem Zeitlimit an.

## V 9.0.1 Autostart konfigurieren

Standardmäßig wird die IBM MQ Console beim Start des mqweb-Servers automatisch gestartet. In IBM MQ 9.0.1 wird die REST API nicht automatisch gestartet. Ab IBM MQ 9.0.2 wird die REST API beim Start des mqweb-Servers automatisch gestartet. **V 9.0.4** Ab IBM MQ 9.0.4 können Sie mit dem Befehl **setmqweb** konfigurieren, ob IBM MQ Console und REST API automatisch gestartet werden. In IBM MQ 9.0.3 und früheren Versionen können Sie konfigurieren, ob IBM MQ Console und REST API automatisch gestartet werden, indem Sie die Datei `mqwebuser.xml` bearbeiten.

### Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie die aktuelle Konfiguration des REST API-Autostarts mit dem folgenden Befehl anzeigen:

```
dspmweb properties -a
```

Im Feld `mqRestAutostart` wird angezeigt, ob die REST API automatisch gestartet wird, und im Feld `mqConsoleAutostart` wird angezeigt, ob die IBM MQ Console automatisch gestartet wird. Weitere Informationen finden Sie in [dspmweb](#).



#### Achtung: z/OS V 9.0.4

Bevor Sie den Befehl **setmqweb** oder **dspmweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist `WLP_user_directory` der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

### Prozedur

#### V 9.0.4

Verwenden Sie eine der folgenden Methoden, um zu konfigurieren, ob die IBM MQ Console und die REST API automatisch gestartet werden:

- Verwenden Sie im Verzeichnis IBM MQ 9.0.4 den Befehl **setmqweb properties** :
  - Konfigurieren Sie mit dem folgenden Befehl, ob die IBM MQ Console automatisch gestartet wird:  

```
setmqweb properties -k mqconsoleAutostart -v start
```

Dabei ist `start` der Wert `True`, wenn die IBM MQ Console automatisch gestartet werden soll, oder `False`, wenn dies nicht der Fall ist.
  - Konfigurieren Sie mit dem folgenden Befehl, ob die REST API einen manuellen Start erfordert:  

```
setmqweb properties -k mqRestAutostart -v start
```

Dabei ist `start` der Wert `True`, wenn die REST API automatisch gestartet werden soll, oder `False`, wenn dies nicht der Fall ist.
- Bearbeiten Sie für IBM MQ 9.0.3 und früher die Datei `mqwebuser.xml`:
  1. Öffnen Sie die Datei `mqwebuser.xml`.  
Die Datei `mqwebuser.xml` befindet sich in einem der folgenden Verzeichnisse:

- **ULW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

## 2. Automatischen Start konfigurieren:

- Konfigurieren Sie, ob die IBM MQ Console einen manuellen Start erfordert, indem Sie in der Datei `mqwebuser.xml` die folgende Zeile innerhalb der `<server>`-Tags hinzufügen oder aktualisieren:

```
<variable name="mqConsoleAutostart" value="start"/>
```

Dabei ist *start* der Wert `True`, wenn die IBM MQ Console automatisch gestartet werden soll, oder `False`, wenn dies nicht der Fall ist.

- Konfigurieren Sie, ob die REST API einen manuellen Start erfordert, indem Sie in der Datei `mqwebuser.xml` die folgende Zeile innerhalb der `<server>`-Tags hinzufügen oder aktualisieren:

```
<variable name="mqRestAutostart" value="start"/>
```

Dabei ist *start* der Wert `True`, wenn die REST API automatisch gestartet werden soll, oder `False`, wenn dies nicht der Fall ist.

## V 9.0.1 Protokollierung konfigurieren

Sie können die Protokollierungsstufen, die maximale Protokolldateigröße und die maximale Anzahl der Protokolldateien konfigurieren, die vom mqweb-Server verwendet werden, auf dem die IBM MQ Console und REST API gehostet werden. **V 9.0.4** Ab IBM MQ 9.0.4 können Sie die Protokollierung mit dem Befehl `setmqweb` konfigurieren. In IBM MQ 9.0.3 und früher können Sie die Protokollierung konfigurieren, indem Sie die Datei `mqwebuser.xml` bearbeiten.

### Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie die aktuelle Konfiguration der REST API-Protokollierung mit dem folgenden Befehl anzeigen:

```
dspmweb properties -a
```

Das Feld `maxTraceFileSize` zeigt die maximale Größe der Tracedatei an. Das Feld `maxTraceFiles` zeigt die maximale Anzahl Tracedateien an, und das Feld `traceSpec` zeigt die Tracestufe an, die verwendet wird. Weitere Informationen finden Sie in [dspmweb](#).



### Achtung: **z/OS** **V 9.0.4**

Bevor Sie den Befehl `setmqweb` oder `dspmweb` unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP\_user\_directory* der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

## Informationen zu diesem Vorgang

Die Protokolldateien für den mqweb-Server befinden sich in einem der folgenden Verzeichnisse:

- **ULW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb/logs`
- **z/OS** Unter z/OS: `WLP_user_directory/servers/mqweb/logs`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script **crtmqweb.sh** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

Weitere Informationen zum Aktivieren der Tracefunktion für die IBM MQ Console und REST API finden Sie unter [IBM MQ Console und REST API](#) nachverfolgen.

## Prozedur

### V 9.0.4

Verwenden Sie eine der folgenden Methoden, um die Protokollierung zu konfigurieren:

- Verwenden Sie im Verzeichnis IBM MQ 9.0.4 den Befehl **setmqweb properties** :
  - Verwenden Sie den folgenden Befehl, um die maximale Größe der Protokolldatei festzulegen:  
`setmqweb properties -k maxTraceFileSize -v size`  
Dabei gibt *size* die Größe (in MB) an, die jede Protokolldatei erreichen kann. Der Standardwert ist 20.
  - Verwenden Sie den folgenden Befehl, um die maximale Anzahl von Dateien festzulegen, die für die Protokollierung verwendet werden sollen:  
`setmqweb properties -k maxTraceFiles -v max`  
Dabei gibt *max* die maximale Anzahl Dateien an. Der Standardwert ist 2.
  - Verwenden Sie den folgenden Befehl, um die Protokollierungsstufe zu konfigurieren:  
`setmqweb properties -k traceSpec -v level`

Hierbei steht *level* für einen der in [Tabelle 37](#) auf [Seite 651](#) aufgelisteten Werte. In der Tabelle werden die Protokollierungsstufen in einer höheren Detaillierungsebene aufgeführt. 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** .

Der Standardwert ist **\*=info**. Ändern Sie diesen Wert, wenn IBM Service sie anfordert.

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 37. 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	Signifikantes Ereignis, das den Serverstatus oder die Server
* =Info	Allgemeine Informationen zur Gesamtaufgabenfortschritt
* =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

- Bearbeiten Sie für IBM MQ 9.0.3 und früher die Datei `mqwebuser.xml`:

1. Öffnen Sie die Datei `mqwebuser.xml`.

Die Datei `mqwebuser.xml` befindet sich in einem der folgenden Verzeichnisse:

-  Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
-  Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

2. Protokollierung konfigurieren:

- Um die maximale Größe der Protokolldatei festzulegen, fügen Sie die folgende Zeile in der Datei `mqwebuser.xml` in den `<server>`-Tags hinzu bzw. bearbeiten Sie sie entsprechend:

```
<variable name="maxTraceFileSize" value="size" />
```

Dabei gibt *size* die Größe (in MB) an, die jede Protokolldatei erreichen kann. Der Standardwert ist 20.

- Um die maximale Anzahl von Dateien festzulegen, die zum Protokollieren verwendet werden sollen, fügen Sie die folgende Zeile in der Datei `mqwebuser.xml` in den `<server>`-Tags hinzu bzw. bearbeiten Sie sie entsprechend:

```
<variable name="maxTraceFiles" value="max" />
```

Dabei gibt *max* die maximale Anzahl Dateien an. Der Standardwert ist 2.

- Um die verwendete Protokollierungsstufe zu konfigurieren, fügen Sie folgende Zeile in der Datei `mqwebuser.xml` in den `<server>`-Tags hinzu bzw. bearbeiten Sie sie entsprechend:

```
<variable name="traceSpec" value="level" />
```

Hierbei steht *level* für einen der Werte, die in der Tabelle [Tabelle 37 auf Seite 651](#) aufgelistet sind.

In der Tabelle werden die Protokollierungsstufen in einer höheren Detaillierungsebene aufgeführt. 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**.

Der Standardwert ist **\*=info**. Ändern Sie diesen Wert, wenn IBM Service sie anfordert.

## Ablaufintervall für LTPA-Token

Es können LTPA-Tokens verwendet werden, um zu vermeiden, dass ein Benutzer bei jeder Anforderung an WebSphere Application Server Liberty Berechtigungsnachweise in Form von Benutzername und Kennwort angeben muss. Sie können das Ablaufintervall für LTPA-Authentifizierungstoken konfigurieren.

### Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie die aktuelle Konfiguration des Tokenablaufs anzeigen, indem Sie den Befehl **dspmqweb properties** mit dem Flag `-a` verwenden. Weitere Informationen finden Sie in [dspmqweb](#). Sie können den Wert des Tokenablaufs mit dem Befehl **setmqweb properties** und den Flags `-k` und `-d` zurücksetzen. Weitere Informationen finden Sie in [setmqweb](#).

**V 9.0.2**

**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** **V 9.0.4**

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** unter ausgeben `z/OS`, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre `mqweb`-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist `WLP_user_directory` der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

### 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 API verwenden, wird ein LTPA-Token generiert, sobald sich der Benutzer unter Verwendung der REST API-Ressource `/login` mit der HTTP-Methode `POST` anmeldet. 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. **V 9.0.4** Ab IBM MQ 9.0.4 können Sie mit dem Befehl **setmqweb** konfigurieren, wann die Tokens ablaufen. In IBM MQ 9.0.3 und früher können Sie konfigurieren, wann die Tokens ablaufen, indem Sie die Datei `mqwebuser.xml` bearbeiten.

### Prozedur

• **V 9.0.4**

Verwenden Sie eine der folgenden Methoden, um den Tokenablauf zu konfigurieren:

- Verwenden Sie im Verzeichnis IBM MQ 9.0.4 den Befehl **setmqweb properties** :

```
setmqweb properties -k ltpaExpiration -v time
```

Dabei gibt *time* die Zeit in Minuten an, bevor das LTPA-Token abläuft und der Benutzer abgemeldet ist. Der Standardwert ist 120 Minuten.

- Bearbeiten Sie für IBM MQ 9.0.3 und früher die Datei `mqwebuser.xml`:

1. Öffnen Sie die Datei `mqwebuser.xml` .

Die Datei `mqwebuser.xml` befindet sich in einem der folgenden Verzeichnisse:

-  Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

-  Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script **crtmqweb.sh** ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

2. Konfigurieren Sie das Ablaufintervall für LTPA-Tokens, indem Sie die folgende Zeile in der Datei `mqwebuser.xml` in den `<server>`-Tags hinzufügen oder bearbeiten:

```
<variable name="ltpaExpiration" value="time" />
```

Dabei gibt *time* die Zeit in Minuten an, bevor das LTPA-Token abläuft und der Benutzer abgemeldet ist. Der Standardwert ist 120 Minuten.

## administrative REST API-Gateway konfigurieren

Das administrative REST API-Gateway ist standardmäßig aktiviert. 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 Standardgateway-Warteschlangenmanager verwendet wird, oder Sie können die Fernverwaltung verhindern, indem Sie das administrative REST API-Gateway mit dem Befehl **setmqweb properties** inaktivieren.

### Informationen zu diesem Vorgang

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

Sie können die aktuelle Konfiguration des administrative REST API-Gateways mit dem folgenden Befehl anzeigen:

```
dspmweb properties -a
```

Das Feld `mqRestGatewayEnabled` zeigt an, ob das Gateway aktiviert ist, und das Feld `mqRestGatewayQmgr` zeigt den Namen des Standard-Gateway-Warteschlangenmanagers an. Weitere Informationen finden Sie in [dspmweb](#) .

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 API finden Sie im Abschnitt [Fernverwaltung mit der REST API](#).



**Achtung:**  

Bevor Sie den Befehl **setmqweb** oder **dspmweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre `mqweb`-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP\_user\_directory* der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

## Prozedur

- 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
```

## V 9.0.4 messaging REST API konfigurieren

Standardmäßig ist auf dem mqweb-Server, auf dem sich die IBM MQ Console und die REST API befinden, die messaging REST API aktiviert. Mit dem Befehl **setmqweb properties** können Sie konfigurieren, ob das Messaging aktiviert oder inaktiviert werden soll.

### Vorbereitende Schritte

Sie müssen ein [privilegierter Benutzer](#) sein, um diese Prozedur ausführen zu können.

Sie können die aktuelle Konfiguration der messaging REST API mit dem folgenden Befehl anzeigen:

```
dspmweb properties -a
```

Im Feld `mqRestMessagingEnabled` ist angegeben, ob die messaging REST API aktiviert oder inaktiviert ist. Weitere Informationen finden Sie in [dspmweb](#).

Um die messaging REST API verwenden zu können, muss der Aufrufende beim mqweb-Server authentifiziert werden und Mitglied der Rolle `MQWebUser` sein. Die Rollen `MQWebAdmin` und `MQWebAdminRO` gelten nicht für messaging REST API. Der Aufrufende muss auch über OAM/RACF berechtigt werden. Weitere Informationen zur Sicherheit für die REST API finden Sie im Abschnitt [Sicherheit der IBM MQ Console und der REST API](#).



**Achtung:** z/OS V 9.0.4

Bevor Sie den Befehl **setmqweb** oder **dspmweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP\_user\_directory* der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

## Prozedur

### V 9.0.4

Gehen Sie zur Konfiguration der messaging REST API wie folgt vor:

- Verwenden Sie den Befehl **setmqweb properties** :

- Konfigurieren Sie mit dem folgenden Befehl, ob die messaging REST API aktiviert ist:

```
setmqweb properties -k mqRestMessagingEnabled -v enabled
```

Für *enabled* lautet der Wert dabei `true`, wenn die messaging REST API aktiviert sein soll, oder andernfalls `false`.

### V 9.0.5 REST API für MFT konfigurieren

Standardmäßig ist auf dem mqweb-Server, auf dem sich die IBM MQ Console und die REST API befinden, die MFT-REST API inaktiviert. Sie können REST API für MFT aktivieren oder inaktivieren, den Koordinationswarteschlangenmanager festlegen und das MFT -Zeitlimit für die Verbindungswiederherstellung mit dem Befehl **setmqweb properties** angeben.

## Vorbereitende Schritte

Sie müssen ein privilegierter Benutzer sein, um diese Prozedur ausführen zu können.

Mit dem folgenden Befehl können Sie die aktuelle Konfiguration der REST API für MFT anzeigen:

```
dspmqweb properties -a
```

Im Feld `mqRestMftEnabled` wird angezeigt, ob REST API for MFT aktiviert oder inaktiviert ist. Im Feld `mqRestMftCoordinationQmgr` ist der Name des Koordinationswarteschlangenmanagers und im Feld `mqRestMftReconnectTimeoutInMinutes` der Zeitlimitwert für MFT-Anforderungen angegeben. Weitere Informationen finden Sie in [dspmqweb](#).

Um REST API for MFT verwenden zu können, muss der Aufrufende beim mqweb-Server authentifiziert werden und mindestens einer der Rollen `MFTWebAdmin` oder `MFTWebAdminRO` angehören.



### Achtung: z/OS V 9.0.4

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP\_user\_directory* der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

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

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

## Informationen zu diesem Vorgang

Bei der Konfiguration der REST API für MFT können Sie drei Eigenschaften konfigurieren:

- Ob die REST API für MFT aktiviert ist. Standardmäßig ist sie inaktiviert.
- Den Namen des Koordinationswarteschlangenmanagers, von dem Informationen abgerufen werden, wenn Sie die Ressourcen der MFT-REST API verwenden. Dieser Warteschlangenmanager muss sich auf demselben System wie der mqweb-Server befinden. Beim Start des mqweb-Servers richtet die REST API für MFT eine Bindungsverbindung zu diesem Warteschlangenmanager her.

Standardmäßig ist dieser WS-Manager-Name leer. Wenn kein Wert festgelegt ist und die MFT-REST API aufgerufen wird, wird ein Fehler HTTP 400 zurückgegeben.

- Das Zeitlimit in Minuten, nach dem die REST API für MFT nicht mehr versucht, eine Verbindung zum Koordinationswarteschlangenmanager herzustellen. Der erste Versuch, die Verbindung erneut herzustellen, wird unmittelbar nach dem Aufbrechen der Verbindung zum Koordinations-WS-Manager hergestellt. Wenn dies fehlschlägt, gibt es ein Intervall von fünf Minuten zwischen jedem Versuch der Verbindungswiederherstellung.

Nach der Zeitlimitüberschreitung bei der Verbindungswiederholung wird der nächste Verbindungsversuch beim Aufruf der REST API-Ressourcen `/transfer` oder `/agent` unternommen. Wenn dieser Verbindungswiederholungsversuch fehlschlägt, versucht MFT alle fünf Minuten erneut, die Verbindung herzustellen, bis das Zeitlimit für die Verbindungswiederholung überschritten wurde.

Standardmäßig beträgt der Zeitlimitwert 30 Minuten. Falls die MFT-REST API aufgerufen wird, wenn der Koordinationswarteschlangenmanager nicht gestartet ist, wird ein Fehler HTTP 503 zurückgegeben.

## Vorgehensweise

### 1. Passen Sie die Konfiguration der REST API für MFT an:

- Konfigurieren Sie mit dem folgenden Befehl, ob die REST API für MFT aktiviert ist:

```
setmqweb properties -k mqRestMftEnabled -v value
```

Dabei ist *Wert true* , wenn REST API für MFT aktiviert werden soll, oder *false* , wenn dies nicht der Fall ist.

- Konfigurieren Sie mit dem folgenden Befehl den Koordinationswarteschlangenmanager, aus dem Übertragungsdetails abgerufen werden:

```
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.

- Konfigurieren Sie mit dem folgenden Befehl das Zeitlimit in Minuten, nach dem die REST API für MFT nicht mehr versucht, eine Verbindung zum Koordinationswarteschlangenmanager herzustellen:

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
```

Dabei gibt *time* die Zeit (in Minuten) vor dem Auftreten des Zeitlimits an.

- Ein Wert zwischen 0-5 gibt an, dass REST API for MFT nur einmal versucht, die Verbindung zum Koordinationswarteschlangenmanager wiederherzustellen. Wenn die Verbindung fehlschlägt, werden keine Versuche unternommen, die Verbindung erneut herzustellen, bis die REST API aufgerufen wird.
- Der Wert -1 gibt an, dass REST API for MFT versucht, die Verbindung wiederherzustellen, bis die Verbindung erfolgreich hergestellt wurde.

### 2. Starten Sie den mqweb-Server erneut, indem Sie die folgenden Befehle eingeben:

```
endmqweb
startmqweb
```

## V 9.0.2 Die JVM des mqweb-Servers optimieren

Standardmäßig verwendet die Java Virtual Machine (JVM) des mqweb-Web-Servers plattformspezifische Standardwerte für die minimale und die maximale Größe des Heapspeichers. Möglicherweise müssen Sie die Standardwerte ändern. Wenn z. B. ein `java.lang.OutOfMemoryError` vom mqweb-Server ausgelöst wird, müssen Sie die maximale Größe des Heapspeichers erhöhen. 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:

-  Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
-  Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

2. Optional: Legen Sie die maximale Heapspeichergröße fest, indem Sie der Datei die folgende Zeile hinzufügen:

```
-XmxMaxSizem
```

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:

```
-XmsMinSizem
```

Dabei gibt *MinSize* die Mindestgröße des Heapspeichers in Megabyte an.

In der folgenden Zeile wird beispielsweise die Mindestgröße des Heapspeichers auf 512 MB festgelegt:

```
-Xms512m
```

4. Starten Sie den mqweb-Server erneut, indem Sie die folgenden Befehle in der Befehlszeile eingeben:

```
endmqweb  
startmqweb
```

## 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 editierbaren Dateien hängt vom Betriebssystem ab:

- **U/LW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/`
- **z/OS** Unter z/OS: `WLP_user_directory`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script **crtmqweb.sh** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

Unter diesem Ausgangsverzeichnis sind die folgenden Verzeichnisse und Dateien vorhanden:

Verzeichnisse und Dateien	Beschreibung
<code>angular.persistence/</code>	Verzeichnis, in dem die Konfiguration des IBM MQ Console-Dashboards gespeichert ist.
<code>servers/</code>	Verzeichnis für WebSphere Liberty Profile-Server.
<code>servers/mqweb</code>	Verzeichnis, das die Verzeichnisstruktur des mqweb-Servers enthält.
<code>servers/mqweb/logs</code>	Verzeichnis, das Protokolle für den mqweb-Server enthält.
<code>servers/mqweb/logs/console.log</code>	Protokoll der grundlegenden Serverstatus- und Operationsnachrichten.
<code>servers/mqweb/logs/ffdc</code>	FFDC-Ausgabeverzeichnis (FFDC-First Failure Data Capture).
<code>servers/mqweb/logs/messages.log</code>	Protokoll der Laufzeitnachrichten vom mqweb-Server, einschließlich IBM MQ Console und REST API. Ältere Nachrichten werden in Dateien gespeichert, die als <code>messages_timestamp.log</code> bezeichnet werden.
<code>servers/mqweb/logs/trace.log</code>	Protokoll des Trace vom mqweb-Server, einschließlich IBM MQ Console und REST API. Ältere Traces werden in Dateien gespeichert, die als <code>trace_timestamp.log</code> bezeichnet werden. Diese Dateien sind nur vorhanden, wenn die Tracefunktion aktiviert ist.
<code>servers/mqweb/logs/state</code>	Serverspezifischer Status.
<code>servers/mqweb/server.xml</code>	Hauptserverkonfigurationsdatei. Diese Datei ist schreibgeschützt. Bearbeiten Sie die <code>mqwebuser.xml</code> -Datei, um die Standardkonfiguration zu überschreiben.
<code>servers/mqweb/mqwebuser.xml</code>	Konfigurationsdatei für IBM MQ Console und REST API. Einstellungen, die in dieser Datei konfiguriert sind, überschreiben die Standardkonfiguration. Sie müssen ein <u>privilegierter Benutzer</u> sein, um diese Datei zu bearbeiten.
<code>servers/mqweb/resources</code>	Verzeichnis, das verschiedene Serverressourcen, wie z. B. Keystores, enthält.
<code>servers/mqweb/workarea</code>	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 editierbaren Dateien hängt vom Betriebssystem ab:

-  Unter UNIX, Linux, and Windows: `MQ_INSTALLATION_PATH/web`
-  Unter z/OS: `installation_directory/web/`

Hierbei steht `installation_directory` für den Installationspfad von IBM MQ UNIX System Services Components.

Die folgende Verzeichnisstruktur und die folgenden Dateien sind an dieser Position vorhanden:

Verzeichnisse und Dateien	Beschreibung
bin/	Verzeichnis, das Liberty-Befehle enthält. Sie müssen ein <u>privilegierter Benutzer</u> 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. Bearbeiten Sie die Datei <code>mqwebuser.xml</code> , um Konfigurationsänderungen vorzunehmen.
mq/libs	Verzeichnis, das gemeinsam genutzte Bibliotheken für die Verwendung durch IBM MQ Console und REST API enthält.
mq/samp	Verzeichnis, das Muster enthält.
mq/samp/configuration	Verzeichnis, das Beispielkonfigurationsdateien enthält, die in die <code>mqwebuser.xml</code> -Datei kopiert werden können.

## IBM MQ in Docker konfigurieren

Verwenden Sie diese Informationen, um IBM MQ mithilfe von Docker zu konfigurieren.

### Informationen zu diesem Vorgang

Mithilfe von Docker können Sie einen IBM MQ-Warteschlangenmanager oder eine IBM MQ-Clientanwendung mit allen Abhängigkeiten in eine standardisierte Einheit für die Softwareentwicklung packen.

Änderungen an Ihrer Anwendung können schnell und einfach in Test- und Bereitstellungssysteme implementiert werden. Diese Funktion kann ein großer Vorteil für die kontinuierliche Bereitstellung in Ihrem Unternehmen sein.

## Prozedur

- Informationen zur Konfiguration von IBM MQ mithilfe von Docker finden Sie in folgenden Unterabschnitten:
  - **Linux** „[Docker-Unterstützung auf Linux-Systemen](#)“ auf Seite 661
  - „[Eigenes Image des IBM MQ-Warteschlangenmanagers mit Docker planen](#)“ auf Seite 661
  - „[Beispiel für ein IBM MQ-Warteschlangenmanagerimage mit Docker erstellen](#)“ auf Seite 662
  - „[Lokale Bindungsanwendungen in separaten Containern ausführen](#)“ auf Seite 666

## **Linux** Docker-Unterstützung auf Linux-Systemen

Informationen, die Sie bei der Verwendung von Docker auf einem Linux-System beachten müssen.

- Das vom Docker-Image verwendete Basisimage muss ein unterstütztes Linux-Betriebssystem verwenden.
- Sie müssen die IBM MQ-Installationsprogramme verwenden, um das Produkt innerhalb des Docker-Image zu installieren.
- Eine Liste der unterstützten Pakete finden Sie unter [IBM MQ-RPM-Komponenten für Linux-Systeme](#).
- **V 9.0.4** Die folgenden Pakete werden nicht unterstützt:
  - MQSeriesBCBridge
  - MQSeriesRDQM
- Das Datenverzeichnis des Warteschlangenmanagers (standardmäßig `/var/mqm`) muss auf einem Docker-Datenträger gespeichert werden, der den persistenten Status behält.

**Wichtig:** Das Union-Dateisystem kann nicht verwendet werden.

Sie müssen entweder ein Hostverzeichnis als Datenträger bereitstellen oder einen Datenträgercontainer verwenden. Weitere Informationen finden Sie unter [Manage data in containers](#) (Daten in Containern verwalten).

- Sie müssen IBM MQ-Steuerbefehle, z. B. **endmqm**, innerhalb des Containers ausführen können.
- Sie müssen zu Diagnosezwecken Dateien und Verzeichnisse aus dem Container heraus abrufen können.
- **V 9.0.3** Sie können Namensbereiche verwenden, um die Namensbereiche des Containers für den Warteschlangenmanager mit anderen Containern gemeinsam zu nutzen, um Anwendungen lokal an einen Warteschlangenmanager binden zu können, der in separaten Containern ausgeführt wird. Weitere Informationen finden Sie im Abschnitt „[Lokale Bindungsanwendungen in separaten Containern ausführen](#)“ auf Seite 666.

## Eigenes Image des IBM MQ-Warteschlangenmanagers mit Docker planen

Verwenden Sie diese Informationen, um IBM MQ mithilfe von Docker zu konfigurieren. Bei der Ausführung eines IBM MQ-Warteschlangenmanagers in Docker sind mehrere Anforderungen zu berücksichtigen. Das Docker-Beispielimage bietet eine Möglichkeit, diesen Anforderungen gerecht zu werden. Wenn Sie jedoch ein eigenes Image verwenden möchten, müssen Sie sich klar machen, wie diese Anforderungen erfüllt werden.

### Prozessüberwachung

Wenn Sie einen Docker-Container ausführen, führen Sie im Prinzip einen einzelnen Prozess (PID 1 innerhalb des Containers) aus, der später untergeordnete Prozesse generieren kann.

Wenn der Hauptprozess beendet wird, stoppt Docker den Container. Für einen IBM MQ-Warteschlangenmanager müssen mehrere Prozesse im Hintergrund ausgeführt werden.

Aus diesem Grund müssen Sie sicherstellen, dass Ihr Hauptprozess aktiv bleibt, solange der Warteschlangenmanager aktiv ist. Es ist sinnvoll, zu überprüfen, ob der Warteschlangenmanager von diesem Prozess aus aktiv ist, z. B. durch Ausführen von Verwaltungsabfragen.

## **/var/mqm füllen**

Docker-Container müssen mit /var/mqm als Docker-Datenträger konfiguriert werden.

Dabei ist das Verzeichnis des Datenträgers leer, wenn der Container zuerst gestartet wird. Dieses Verzeichnis wird in der Regel zur Installationszeit belegt, bei Verwendung von Docker sind Installations- und Laufzeitumgebung jedoch separat.

**V 9.0.3** Um dieses Problem zu lösen, können Sie beim Start Ihres Containers den Befehl **crtmqdir** verwenden, um /var/mqm bei der ersten Ausführung zu füllen.

## **Beispiel für ein IBM MQ-Warteschlangenmanagerimage mit Docker erstellen**

Verwenden Sie diese Informationen, um ein Docker-Beispielimage für die Ausführung eines IBM MQ-Warteschlangenmanagers in einem Docker-Container zu erstellen.

### **Informationen zu diesem Vorgang**

Zunächst erstellen Sie ein Basisimage, das ein Ubuntu Linux-Dateisystem und eine bereinigende Installation von IBM MQ enthält.

Dann erstellen Sie eine weitere Docker-Imageebene auf der Basis, die einige IBM MQ-Konfigurationen hinzufügt, um grundlegende Sicherheit für Benutzer-IDs und Kennwörter zu ermöglichen.

Schließlich führen Sie einen Docker-Container unter Verwendung dieses Image als Dateisystem aus, wobei der Inhalt von /var/mqm von einem containerspezifischen Docker-Datenträger auf dem Hostdateisystem von Docker bereitgestellt wird.

### **Prozedur**

- Weitere Informationen zum Erstellen eines Docker-Beispielimage für die Ausführung eines IBM MQ-Warteschlangenmanagers in einem Docker-Container finden Sie in den folgenden Unterabschnitten:
  - [„Beispielbasisimage eines IBM MQ-Warteschlangenmanagers erstellen“](#) auf Seite 662
  - [„Beispiel für ein konfiguriertes IBM MQ-Warteschlangenmanagerimage erstellen“](#) auf Seite 664

## **Beispielbasisimage eines IBM MQ-Warteschlangenmanagers erstellen**

Damit IBM MQ in Docker verwendet werden kann, müssen Sie zunächst ein Basisimage mit einer sauberen IBM MQ-Installation erstellen. In den folgenden Schritten wird gezeigt, wie ein Beispiel für ein Basisimage erstellt wird. Dabei wird der Code verwendet, der auf GitHub gehostet wird

### **Informationen zu diesem Vorgang**

#### **Verwenden von Make zum Erstellen des Docker-Image**

Wenn Sie die make-Dateien verwenden möchten, die im [mq-container-GitHub-Repository](#) bereitgestellt werden, um Ihr Docker-Produktionsimage zu erstellen, befolgen Sie die Anweisungen im Abschnitt [Docker-Image erstellen in GitHub](#).

#### **Docker-Image manuell mithilfe von Docker erstellen**

Wenn Sie das Image manuell mit Hilfe von Docker erstellen möchten, führen Sie die folgenden Schritte aus.

### **Vorgehensweise**

1. Installieren Sie die vorausgesetzten Pakete.

In diesen Anweisungen werden einige Linux-Pakete verwendet, die Sie installieren müssen.

- Unter Ubuntu:

```
sudo apt-get install python git
```

- Unter Red Hat Enterprise Linux:

```
sudo yum install python git
```

2. Erstellen Sie ein Verzeichnis `downloads`, indem Sie den Befehl `mkdir downloads` ausgeben.
3. Laden Sie den IBM MQ-Server für das Linux-Image unter Verwendung von Passport Advantage herunter.

Weitere Informationen finden Sie unter [Installation mit Electronic Software Download](#).

Wählen Sie beispielsweise die Datei `WS_MQ_V9.0.5.0_LINUX_ON_X86_64_IM.tar.gz` aus, und legen Sie die Datei in das von Ihnen erstellte Verzeichnis `downloads` ab.

**Anmerkung:**  Sie müssen sicherstellen, dass Sie die Debian-Installation herunterladen, wenn Sie Ubuntu als Basis-Image verwenden wollen.

4. Erstellen Sie den IBM MQ-Server für die Linux-Imagedatei (`tar.gz`) auf einem HTTP- oder FTP-Server.

Der Grund dafür ist, dass der Speicherplatz in den Docker-Image-Ebenen gespeichert wird. Jede Anweisung in einer Docker-Datei bewirkt, dass eine neue Image-Ebene erstellt wird.

Wenn Sie die Anweisungen **ADD** oder **COPY** gefolgt von einer **RUN**-Installationsanweisung verwenden, werden die hinzugefügten oder kopierten Dateien auf einer neuen Imageebene festgeschrieben.

Auch wenn Sie die Datei in nachfolgenden Layern löschen, ist die Datei noch in der vorherigen Ebene vorhanden. Aus diesem Grund empfiehlt es sich, die Dateien mit einem einzigen **RUN**-Befehl herunterzuladen und zu installieren. Dies bedeutet, dass die Dateien im Netz verfügbar sein müssen.

Sie können z. B. Python verwenden, um einen HTTP-Server auszuführen und alle Dateien in Ihrem aktuellen Verzeichnis zu bedienen:

```
pushd downloads
nohup python -m SimpleHTTPServer 8000 &
popd
```

5. Extrahieren Sie die Beispieldateien, um ein unterstütztes Docker-Image von GitHub zu erstellen:

-  Geben Sie den folgenden Befehl ein:

```
git clone -b mq-9-lts https://github.com/ibm-messaging/mq-docker mq-docker
```

-  Geben Sie den folgenden Befehl ein:

```
git clone https://github.com/ibm-messaging/mq-container mq-container
```

6. Identifizieren Sie Ihre lokale IP-Adresse.

Ihre Adresse ist spezifisch für Ihre lokale Umgebung, sollte aber verfügbar sein, wenn Sie den folgenden Befehl ausführen:

```
ip addr show
```

Beachten Sie, dass `localhost` nicht funktioniert.

7. Erstellen Sie das IBM MQ-Basisimage, indem Sie den folgenden Befehl ausgeben, indem Sie die IP-Adresse und den Dateinamen in der `MQ_URL` für die Werte, die Sie gerade angegeben haben, ersetzen:  
Beispiel:

## • LTS

```
sudo docker build --tag mq --build-arg MQ_URL=http://10.0.2.15:8000/WS_MQ_V9.0.0.0_LI□  
NUX_ON_X86_64_IM.tar.gz mq-docker
```

## • CD

```
sudo docker build --tag mq --build-arg MQ_URL=http://10.0.2.15:8000/WS_MQ_V9.0.0.0_LI□  
NUX_ON_X86_64_IM.tar.gz mq-container/Dockerfile-server mq-container
```

## Ergebnisse

Sie verfügen nun über ein Docker-Image mit installiertem IBM MQ.

## Beispiel für ein konfiguriertes IBM MQ-Warteschlangenmanagerimage erstellen

Nachdem Sie Ihr generisches Basisimage für IBM MQ Docker erstellt haben, müssen Sie Ihre eigene Konfiguration anwenden, um einen sicheren Zugriff zu ermöglichen. Erstellen Sie dazu ein eigenes Docker-Image, wobei Sie das generische Image als übergeordnetes Element verwenden. In den folgenden Schritten wird gezeigt, wie ein Beispielbild mit einer minimalen Sicherheitskonfiguration erstellt wird.

## Vorgehensweise

1. Erstellen Sie ein neues Verzeichnis und fügen Sie eine Datei mit der Bezeichnung `config.mqsc` mit den folgenden Inhalten hinzu:

```
DEFINE CHANNEL(PASSWORD.SVRCONN) CHLTYPE(SVRCONN)  
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody') +  
DESCR('Allow privileged users on this channel')  
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) DESCR('BackStop rule')  
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(CHANNEL) CHCKCLNT(REQUIRED)  
ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) ADOPTCTX(YES)  
REFRESH SECURITY TYPE(CONNAUTH)
```

Beachten Sie, dass im vorigen Beispiel eine einfache Benutzer-ID und Kennwortauthentifizierung verwendet wird. Sie können allerdings auch die für Ihr Unternehmen erforderliche Sicherheitskonfiguration anwenden.

2. Erstellen Sie eine Datei mit der Bezeichnung `Dockerfile` mit den folgenden Inhalten:

```
FROM mq  
RUN useradd johndoe -G mqm && \  
    echo johndoe:passw0rd | chpasswd  
COPY config.mqsc /etc/mqm/
```

Dabei gilt:

- `johndoe` ist die Benutzer-ID, die Sie hinzufügen möchten.
- `passw0rd` ist das ursprüngliche Kennwort.

3. Erstellen Sie Ihr angepasstes Docker-Image mit dem folgenden Befehl:

```
sudo docker build -t mymq .
```

Dabei steht `"."` für das Verzeichnis, das die beiden gerade erstellten Dateien enthält.

Docker erstellt anschließend einen temporären Container mithilfe des Images und führt die verbleibenden Befehle aus.

Der Befehl **RUN** fügt einen Benutzer namens `johndoe` mit Kennwort `passw0rd` hinzu und der Befehl **COPY** fügt die Datei `config.mqsc` an einer bestimmten Position hinzu, die dem übergeordneten Image bekannt ist.

4. Führen Sie das neue angepasste Image aus, um einen neuen Container mit dem soeben erstellten Plattenimage zu erstellen.

In Ihrer neuen Imageebene wurde kein bestimmter Befehl für die Ausführung angegeben, so dass er von dem übergeordneten Image übernommen wurde. Der Eingangspunkt des übergeordneten Elements (der Code ist auf GitHub verfügbar):

- Erstellen einen Warteschlangenmanager
- Startet den Warteschlangenmanager
- Erstellt einen Standardlistener
- Anschließend werden alle MQSC-Befehle aus `/etc/mqm/config.mqsc` ausgeführt.

Geben Sie die folgenden Befehle aus, um Ihr neu angepasstes Image auszuführen:

```
sudo docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

Dabei gilt Folgendes:

#### Erster Parameter `env`

Übergibt eine Umgebungsvariable an den Container, der bestätigt, dass die Lizenz für IBM IBM WebSphere MQ akzeptiert wird. Sie können auch die Variable `LICENSE` für die Anzeige der Lizenz festlegen.

Weitere Informationen zu IBM MQ -Lizenzen finden Sie unter [IBM MQ -Lizenzinformationen](#).

#### Zweiter Parameter `env`

Legt den Namen des Warteschlangenmanagers fest, den Sie verwenden.

#### Parameter 'Volume'

Dem Container wird mitgeteilt, dass alle Informationen, die von MQ in `/var/mqm` geschrieben werden, tatsächlich im Host in `/var/example` geschrieben werden sollten.

Mit dieser Option können Sie den Container später auf einfache Weise löschen und trotzdem alle persistenten Daten beibehalten. Sie vereinfacht auch die Anzeige von Protokolldateien.

#### Parameter 'Publish'

Ports im Hostsystem werden Ports im Container zugeordnet. Der Container wird standardmäßig mit seiner eigenen internen IP-Adresse ausgeführt, d. h. Sie müssen alle Ports, die Sie zugänglich machen möchten, gezielt zuordnen.

In diesem Beispiel muss Port 1414 auf dem Host Port 1414 im Container zugeordnet werden.

#### Parameter 'Detach'

Führt den Container im Hintergrund aus.

## Ergebnisse

Sie haben ein konfiguriertes Docker-Image erstellt und können die aktiven Container mit dem Befehl "docker **ps**" anzeigen. Sie können die IBM MQ-Prozesse anzeigen, die in Ihrem Container ausgeführt werden, indem Sie den Befehl "docker **top**" verwenden.



**Achtung:** Wenn Ihr Container bei Verwendung des Docker-Befehls **ps** nicht angezeigt wird, ist der Container möglicherweise fehlgeschlagen. Sie können fehlgeschlagene Container mit dem Befehl 'docker **ps -a**' anzeigen.

Die Container-ID wird mit dem Docker-Befehl **ps -a** angezeigt und wurde auch ausgegeben, als Sie den Docker-Befehl **run** ausgegeben haben.

Sie können die Protokolle eines Containers mit dem Docker-Befehl **logs \${CONTAINER\_ID}** anzeigen.

Ein häufiges Problem ist, dass **mqconfig** anzeigt, dass bestimmte Kerneinstellungen auf dem Docker -Host nicht korrekt sind. Die Kerneinstellungen werden vom Docker-Host und den Containern gemeinsam genutzt und müssen ordnungsgemäß festgelegt werden (siehe [Hardware- und Softwarevoraussetzungen auf UNIX and Linux-Systemen](#)).

Beispielsweise kann die maximale Anzahl offener Dateien mit dem Befehl **sysctl fs.file-max=524288** festgelegt werden.

## V 9.0.3 Lokale Bindungsanwendungen in separaten Containern ausführen

Mit der gemeinsamen Nutzung von Prozessnamensbereichen zwischen Containern in Docker. Sie können jetzt Anwendungen ausführen, die eine lokale Bindungsverbindung zu IBM MQ in separaten Containern aus dem IBM MQ-Warteschlangenmanager benötigen. Diese Funktion wird in IBM MQ 9.0.3-Warteschlangenmanagern und späteren Warteschlangenmanagern unterstützt.

### Informationen zu diesem Vorgang

Sie müssen die folgenden Einschränkungen beachten:

- Docker Version 1.12 oder höher muss verwendet werden.
- Sie müssen den PID-Namensbereich der Container mit dem Argument `--pid` gemeinsam nutzen.
- Sie müssen den IPC-Namensbereich der Container mit dem Argument `--ipc` gemeinsam nutzen.
- Sie müssen eine der folgenden Schritte ausführen:
  1. Nutzen Sie den UTS-Namensbereich der Container gemeinsam mit dem Host, indem Sie das Argument `--uts` verwenden, oder
  2. Stellen Sie sicher, dass die Container denselben Hostnamen verwenden, indem Sie das Argument `-h` oder `--hostname` verwenden.
- Sie müssen das IBM MQ-Datenverzeichnis in einem Volume anhängen, das für alle Container unter dem Verzeichnis `/var/mqm` verfügbar ist.

Sie können diese Funktionalität ausprobieren, indem Sie die folgenden Schritte auf einem Linux-System ausführen, auf dem bereits Docker 1.12 oder höher installiert ist.

Im folgenden Beispiel wird das Beispielcontainerimage von IBM MQ Docker verwendet. Einzelheiten zu diesem Image finden Sie unter [Github](#).

### Vorgehensweise

1. Erstellen Sie mit folgendem Befehl ein temporäres Verzeichnis, das als Datenträger verwendet werden soll:

```
mkdir /tmp/dockerVolume
```

2. Erstellen Sie mit folgendem Befehl einen Warteschlangenmanager (QM1) in einem Container mit der mit der Bezeichnung `sharedNamespace`:

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm --uts host --name sharedNamespace ibmcom/mq
```

3. Starten Sie einen zweiten Container mit dem Namen `secondaryContainer`, der auf `ibmcom/mq` basiert, aber erstellen Sie keinen Warteschlangenmanager, indem Sie folgenden Befehl ausgeben:

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid container:sharedNamespace --ipc container:sharedNamespace --uts host --name secondaryContainer -it --detach ibmcom/mq
```

4. Führen Sie den Befehl **dspsmq** im zweiten Container aus, um den Status der beiden Warteschlangenmanager anzuzeigen:

```
docker exec secondaryContainer dspmq
```

5. Führen Sie den folgenden Befehl aus, um die MQSC-Befehle für den Warteschlangenmanager zu verarbeiten, der im anderen Container aktiv ist:

```
docker exec -it secondaryContainer runmqsc QM1
```

## Ergebnisse

Sie verfügen jetzt über lokale Anwendungen, die in separaten Containern ausgeführt werden, und können jetzt Befehle wie **dspmq**, **amqspmt**, **amqsget** und **runmqsc** erfolgreich als lokale Bindungen zum QM1-Warteschlangenmanager aus dem sekundären Container ausführen.

Wenn nicht das erwartete Ergebnis angezeigt wird, finden Sie unter [„Fehlerbehebung für Ihre Namensbereichsanwendungen“](#) auf Seite 667 weitere Informationen.

### V 9.0.3 Fehlerbehebung für Ihre Namensbereichsanwendungen

Wenn Sie gemeinsam genutzte Namensbereiche verwenden, müssen Sie sicherstellen, dass alle Namensbereiche (IPC, PID und UTS/Hostname) und die angehäng. Datenträger gemeinsam genutzt werden, da Ihre Anwendungen andernfalls nicht funktionieren.

Im Abschnitt [„Lokale Bindungsanwendungen in separaten Containern ausführen“](#) auf Seite 666 finden Sie eine Liste der Einschränkungen, die Sie beachten müssen.

Wenn Ihre Anwendung nicht alle aufgeführten Einschränkungen erfüllt, kann es zu Problemen kommen, bei denen der Container startet, aber die erwartete Funktion nicht ausgeführt wird.

Die folgende Liste enthält einige häufige Ursachen und das Verhalten, das Sie wahrscheinlich sehen, wenn Sie eine der Einschränkungen vergessen haben.

- Wenn Sie vergessen haben, einen Namensbereich (UTS/PID/IPC) gemeinsam zu nutzen oder den Hostnamen der Container als denselben festzulegen, aber den Datenträger zu mounten, kann Ihr Container den Warteschlangenmanager zwar anzeigen, aber nicht mit dem Warteschlangenmanager interagieren.
  - Für **dspmq**-Befehle wird Folgendes angezeigt:

```
docker exec container dspmq
```

```
QMNAME(QM1)                STATUS(Status not available)
```

- Für Befehle des Typs **runmqsc** oder andere Befehle, die versuchen, eine Verbindung zum Warteschlangenmanager herzustellen, wird wahrscheinlich die Fehlermeldung AMQ8146 angezeigt:

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2023.
Starting MQSC for queue manager QM1.
AMQ8146: IBM MQ queue manager not available
```

- Wenn Sie den gesamten erforderlichen Namensbereich gemeinsam nutzen, aber keinen gemeinsam genutzten Datenträger in das Verzeichnis `/var/mqm` aufnehmen und einen gültigen IBM MQ-Datenpfad haben, empfangen Ihre Befehle auch AMQ8146-Fehlermeldungen.

**dspmq** ist jedoch nicht in der Lage, Ihren Warteschlangenmanager zu sehen, und gibt stattdessen eine leere Antwort zurück:

```
docker exec container dspmq
```

- Wenn Sie den gesamten erforderlichen Namensbereich gemeinsam nutzen, aber keinen gemeinsam genutzten Datenträger in das Verzeichnis `/var/mqm` aufnehmen, und Sie haben keinen gültigen IBM MQ-Datenpfad (oder gar keinen IBM MQ-Datenpfad), sehen Sie verschiedene Fehler, da der Datenpfad eine Schlüsselkomponente einer IBM MQ-Installation ist. Ohne den Datenpfad kann IBM MQ nicht ausgeführt werden.

Wenn Sie einen der folgenden Befehle ausführen und Antworten ähnlich den Antworten finden, die in diesen Beispielen aufgeführt sind, sollten Sie überprüfen, ob Sie das Verzeichnis angehängt oder ein IBM MQ-Datenverzeichnis erstellt haben:

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff

docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.

docker exec container mqrc
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715

docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.

docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container dltmqm QM1
AMQ7002: An error occurred manipulating a file.

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715
```

## **Windows** > **Linux** > **V 9.0.2** **IBM MQ für die Verwendung mit dem IBM Cloud Product Insights-Service in IBM Cloud konfigurieren**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im [Blogbeitrag Service Deprecation: IBM Cloud Product Insights](#).

## **Windows** > **Linux** > **V 9.0.2** **IBM Cloud Product Insights-Serviceinstanz in IBM Cloud (formerly Bluemix) erstellen**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im [Blogbeitrag Service Deprecation: IBM Cloud Product Insights](#).

## **Windows** > **Linux** > **V 9.0.2** **Warteschlangenmanager für die Verwendung mit der IBM Cloud Product Insights-Serviceinstanz in IBM Cloud (formerly Bluemix) konfigurieren**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im [Blogbeitrag Service Deprecation: IBM Cloud Product Insights](#).

## **V 9.0.4** **Verbindung zu IBM Cloud Product Insights in IBM Cloud über einen HTTP-Proxy herstellen**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im [Blogbeitrag Service Deprecation: IBM Cloud Product Insights](#).

## **V 9.0.4** **Behebung von Fehlern bei der Verbindung zu Product Insights**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im [Blogbeitrag Service Deprecation: IBM Cloud Product Insights](#).

## 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

- IBM MQ Bridge to Salesforce ist unter **Linux** Linux for System x (64 Bit) verfügbar. Für die Verbindung zu Warteschlangenmanagern, die unter IBM WebSphere MQ 6.0 und früher ausgeführt werden, wird die Bridge nicht unterstützt.
- Installieren Sie das **MQSeriesSFBridge** -Paket. Weitere Informationen finden Sie im Abschnitt [IBM MQ-Server unter Linux installieren](#).

### Informationen zu diesem Vorgang

Salesforce ist eine Cloud-basierte Customer-Relationship-Management-Plattform. Wenn Sie Salesforce für die Verwaltung von Kundendaten und Interaktionen verwenden, können Sie in IBM MQ 9.0.2 über die IBM MQ Bridge to Salesforce Salesforce-Push-Topics und -Plattformereignisse abonnieren, die dann auf 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.

**V 9.0.4**

Ab IBM MQ 9.0.4 können Sie die Bridge auch zum Erstellen von Ereignisnachrichten für Plattformereignisse in Salesforce verwenden.

Eine Übersicht über die IBM MQ Bridge to Salesforce finden Sie in dem Diagramm in [Abbildung 1](#).

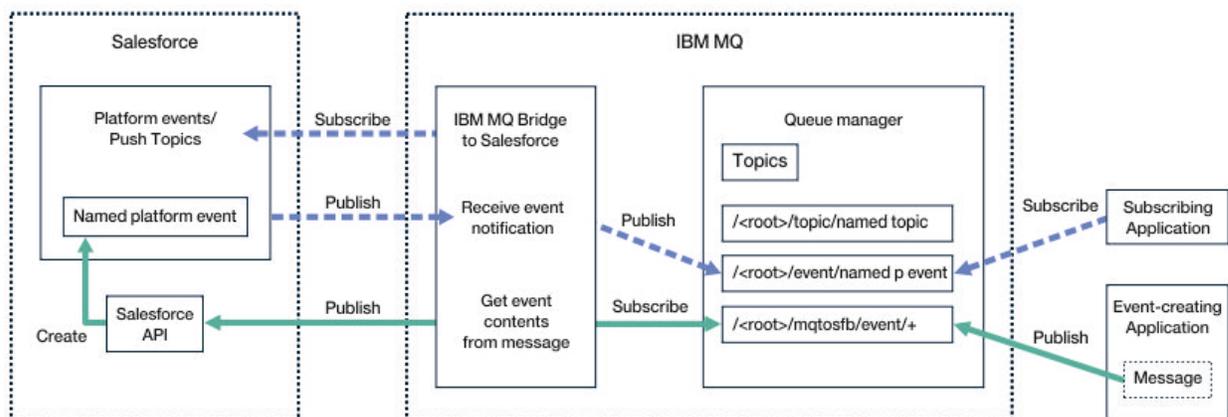


Abbildung 97. IBM MQ Bridge to Salesforce

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 Salesforce zu 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 ermitteln, die von der Plattform Force.com erstellt oder verarbeitet 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 670.

- **V 9.0.4** Informationen zum Erstellen der Konfiguration zum Erstellen von Ereignisnachrichten für Salesforce-Plattformereignisse finden Sie in „[Ereignisnachrichten für Salesforce-Plattformereignisse erstellen](#)“ auf Seite 675.

Sie können die Daten von der Bridge auf zwei Arten überwachen: über IBM MQ Console und mithilfe des Parameters **-p** mit dem Befehl **amqsrua**. Eine Gruppe von Daten wird für den Gesamtüberbrückungsstatus veröffentlicht:

- Die Gesamtzahl der Push-Topic-Nachrichten, die in einem Intervall verarbeitet werden (unter der STATUS/PUSHTOPIC -Baumstruktur).
- Anzahl der in diesem Intervall sichtbaren Push-Themen.
- Die Gesamtzahl der Plattformereignisse, die in einem Intervall verarbeitet werden (unter der Baumstruktur STATUS/PLATFORM).
- Die Anzahl der Plattformereignisse, die in diesem Intervall angezeigt werden.
- **V 9.0.4** Gesamtzahl der von IBM MQ erstellten Plattformereignisse, die in einem Intervall verarbeitet werden (in der Struktur STATUS/MQPE).
- **V 9.0.4** Eindeutige Anzahl der von IBM MQ erstellten Plattformereignisse, die in diesem Intervall angezeigt werden.
- **V 9.0.4** 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-Topic verwendet den vollständigen Salesforce-Topic-Namen sowie die Angabe `/event` oder `/topic` im Objekt-namen:

- Die Anzahl der Nachrichten, die in einem Intervall verarbeitet werden.

Wenn Sie die IBM MQ Console für die Überwachung der Bridge-Daten konfigurieren möchten, beachten Sie die Schritte 9 und 10 in der nächsten Task, [IBM MQ Bridge to Salesforce konfigurieren](#). Informationen zur Verwendung des Befehls **amqsrua** finden Sie im Abschnitt [IBM MQ Bridge to Salesforceüberwachen](#).

Führen Sie die Schritte in den folgenden Tasks aus, um die IBM MQ Bridge to Salesforce zu konfigurieren und auszuführen:

## Vorgehensweise

1. Konfigurieren Sie die IBM MQ Bridge to Salesforce.
2. **V 9.0.4**  
Erstellen Sie Ereignisnachrichten für Salesforce-Plattformereignisse.
3. Führen Sie die IBM MQ Bridge to Salesforce aus.

## Zugehörige Informationen

[runmqsfb \(IBM MQ Bridge to Salesforce ausführen\)](#)

[Tracefunktion für IBM MQ Bridge to Salesforce](#)

Linux

V 9.0.2

## 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

- 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 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.

**V 9.0.4** Informationen zum Erstellen der Konfiguration zum Erstellen von Ereignisnachrichten für Salesforce-Plattformereignisse finden Sie in „[Ereignisnachrichten für Salesforce-Plattformereignisse erstellen](#)“ auf Seite 675.

### 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 Überwachungsthema und die Anwendungen, die eine Verbindung zu IBM MQ herstellen, können unter Umständen nach Push-Themen unter `/topic/EscalatedCases` und Plattformereignissen unter `/event/NewCustomer__e` suchen.

Die veröffentlichte Nachricht enthält Steuerinformationen und die Datenstruktur, die die angeforderten Datenfelder enthält. Bei Push-Themen ist die Datenstruktur ein **subject** 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 persistierte *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 regelmä-

ßigen 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.

**V 9.0.4** Informationen zum Erstellen der Konfiguration zum Erstellen von Ereignisnachrichten für Plattformereignisse finden Sie in „[Ereignisnachrichten für Salesforce-Plattformereignisse erstellen](#)“ auf Seite 675.

## 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 672 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 einen Namen für das Zertifikat in das Feld **Bezeichnung** ein, drücken Sie die Tabulatortaste und 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. Öffnen Sie den aus Salesforce exportierten Keystore in der Key Management GUI von IBM und füllen Sie die Unterzeichnerzertifikate aus.
    - a) Führen Sie zum Öffnen der IBM Key Management GUI den Befehl `strmqikm` aus.  
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 **Connected App Name** (Name der verbundenen Anwendung) einen Namen für Ihre IBM MQ Bridge to Salesforce ein, zum Beispiel **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.

```
runmqsf -o new_config.cfg
```

Die vorhandenen Werte werden in den eckigen Klammern angezeigt. Drücken Sie die Eingabetaste , um vorhandene Werte zu akzeptieren, drücken Sie die Leertaste und anschließend die

Eingabetaste , um die Werte zu löschen, und geben Sie anschließend 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              : []
V9.0.4 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 Schlüsselspeicherkenwort. 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]
```

```
V 9.0.4 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 Beispieldatei `mqsfbService.mqsc` 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. **V 9.0.1**  
Optional: Befolgen Sie die Anweisungen im Abschnitt [Einführung in die IBM MQ Console](#), um die IBM MQ Console einzurichten.
10. **Anmerkung:** Bevor Sie Daten über die Bridge in MQ Console sehen können, müssen Sie die Bridge mindestens einmal ausführen, damit die Verbindungen zu Salesforce und IBM MQ hergestellt werden können, wenn sie gestartet wird. Die Meta-Themen für die Brücke werden bei der Bridge-Start-up-Brücke veröffentlicht.

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** aus dem Dropdown-Menü **Zu überwachender Ressource , Quelle**.

- f) Klicken Sie auf **Speichern**.

## 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 682 beschriebenen Schritte aus.

### Zugehörige Informationen

[runmqsfb \(IBM MQ Bridge to Salesforce ausführen\)](#)

[Tracefunktion für IBM MQ Bridge to Salesforce](#)

[IBM MQ Bridge to Salesforce überwachen](#)

Linux

V 9.0.4

## Ereignisnachrichten für Salesforce-Plattformereignisse erstellen

Sie können IBM MQ konfigurieren und die IBM MQ Bridge to Salesforce-Parameter eingeben, um die Konfigurationsdatei zu erstellen und die Bridge zum Erstellen von Ereignisnachrichten für Salesforce-Plattformereignisse zu verwenden.

### 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 670.

### Ereignisnachrichten für Salesforce-Plattformereignisse erstellen

Von IBM MQ 9.0.4 aus können Sie eine IBM MQ-Anwendung verwenden, um Nachrichten zu erstellen, die in einem Warteschlangenmanager-Thema `/root/mqtosfb/event/+` enthalten sind. 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 zusätzlich zu den Attributen in IBM MQ 9.0.2 bereitstellen, die zum Abonnieren von Push-Themen und Plattformereignissen verwendet wurden:

- Erstellen und fügen Sie den Namen des **MQ Publication Error Queue** s in den Brückenkonfigurationsattributen für **Verbindung zum Warteschlangenmanager** hinzu.
- Setzen Sie die Option **Subscribe to MQ publications for platform events** auf `Y` in den Brückenkonfigurationsattributen für die Definition des **Verhaltens des Bridge-Programms**.

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 679, 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 herstellt, nicht permanent. Alle Veröffentlichungen, die von IBM MQ-Anwendungen ausgeführt werden, während die Brücke nicht aktiv ist, werden nicht verarbeitet.
- Wenn Sie die **At-least-once** -Servicequalität 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 den 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, wo die Themenzeichenfolge `/sƒ/mqtosƒb/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 den Subskriptionsnamen **SUB**, der in `:D_SUB_RUNMQSFB` endet. Setzen Sie den Unterbefehl `delete` ab und schließen Sie die **SUBID** der Subskription ein, die Sie löschen möchten. Beispiel: `DELETE SUB SUBID(414D5120514D3120202020202020205C589459987E8620)`

### Stoppen Sie die Brücke, und starten Sie die Brücke 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.

```
stirmqm PEQM1
```

2. **Anmerkung:** Wenn Sie für die Anmeldung und die Sicherheit bestehende Salesforce-Berechtigungs-nachweise 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 einen Namen für das Zertifikat in das Feld **Bezeichnung** ein, drücken Sie die Tabulatortaste und 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. Öffnen Sie den aus Salesforce exportierten Keystore in der Key Management GUI von IBM und füllen Sie die Unterzeichnerzertifikate aus.
- Führen Sie zum Öffnen der IBM Key Management GUI den Befehl **strmqikm** aus. Weitere Informationen finden Sie unter [Using runmqckm, runmqakm und strmqikm to manage digital certificates](#).
  - Klicken Sie auf **Open a key database file** (Schlüsseldatenbankdatei öffnen) und navigieren Sie zum Speicherort des Salesforce-Keystore.
  - Klicken Sie auf **Öffnen**, stellen Sie sicher, dass Sie **JKS** aus den Optionen für **Schlüsseldatenbanktyp** auswählen, und klicken Sie dann auf **OK**.
  - Geben Sie das Kennwort ein, das Sie für den Keystore in Schritt 3f erstellt haben, und klicken Sie dann auf **OK**.
  - Wählen Sie in den Optionen für **Key database content** die Option **Signer Certificates** aus.
  - Klicken Sie auf **Populate**.
  - 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 Anwendungsverbinding 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.
- Wählen Sie **Erstellen** und dann **Apps** im Menü **Build** Ihrer **Force.com Home**-Seite aus.  
Die Seite **Apps** wird geöffnet.
  - Klicken Sie im Abschnitt **Connected Apps** auf **Neu**.  
Die Seite **Neue verbundene Anwendung** wird geöffnet.
  - Geben Sie im Feld **Connected App Name** (Name der verbundenen Anwendung) einen Namen für Ihre IBM MQ Bridge to Salesforce ein, zum Beispiel **MQBridgeToSalesforce**.
  - 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.
  - Geben Sie Ihre **Kontaktmail-Adresse** ein.
  - Wählen Sie die Option **OAuth-Einstellungen aktivieren** im Abschnitt **API (Enable OAuth Settings)** aus.  
Weitere Optionen in diesem Abschnitt werden dann angezeigt.
  - Fügen Sie Ihre **Callback-URL** hinzu, z. B. `https://www.ibm.com`.
  - 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.
  - Klicken Sie auf **Speichern**.
  - Klicken Sie auf **Weiter**.
  - 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/mqsfb/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

einzigste 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 Brückenkonfigurationsparameters hinzufügen, wie in Schritt „8.a“ auf Seite 679 gezeigt.

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 z. B. das Beispiel **AMQSPUBA** verwenden, um die folgende JSON-formatierte Nachricht 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 nach dem Starten der Bridge zu erstellen. Geben Sie im Verzeichnis *MQ installation location/samp/bin* den folgenden Befehl 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 die Eingabetaste , um vorhandene Werte zu akzeptieren, drücken Sie die Leertaste und anschließend die Eingabetaste , um die Werte zu löschen, und geben Sie anschließend 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 Anmeldungsendpoint 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** von der Standardeinstellung *N* in *Y* ändern, um die Brücke zum Erstellen von Ereignisnachrichten verwenden zu können. 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 Beispieldatei `mqsfbService.mqsc` 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. **V 9.0.1**

- 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 Thema `/sf/mqtosfb/event/Salesforce Platform Object Name/API name` beginnen, zeigt das Widget die Anzahl der gesamten Nachrichtenergebnisse an, die von der Bridge erstellt wurden.

## V 9.0.4 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 reiht eine Nachricht in ein bestimmtes WS-Manager-Thema ein, z. B. `/root/mqtosfb/event/MQPlatformEvent1__e`. 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 Eingabennachricht 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 Eingabennachricht 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 Plattformereignisinhalte, Status 400 Text

```
[{"message": "No such column 'Name__c' on subject of type MQPlatformEvent2__e", "errorCode": "INVALID_ID_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

V 9.0.2

## 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. **V 9.0.4** Ab IBM MQ 9.0.4 kann die Bridge auch Ereignisnachrichten für Salesforce-Plattformereignisse erstellen.

### Vorbereitende Schritte

Sie haben Konfigurationsschritte in der Task abgeschlossen:

- „IBM MQ Bridge to Salesforce konfigurieren“ auf Seite 670
- **V 9.0.4** „Ereignisnachrichten für Salesforce-Plattformereignisse erstellen“ auf Seite 675

### 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 zu subscribierenden Push-Themen oder Plattformereignisse in Salesforce **V 9.0.4** oder das Plattformereignis, 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:

In IBM MQ 9.0.2

```
Successful connection to queue manager QM1  
Successful login to Salesforce at https://eu11.salesforce.com  
Ready to process events.
```

- Wenn Sie die Bridge verwenden, um Salesforce-Push-Themen und -Plattformereignisse zu abonnieren, 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 Informationen

[runmqsfb \(IBM MQ Bridge to Salesforce ausführen\)](#)

[IBM MQ Bridge to Salesforce überwachen](#)

Richten Sie IBM MQ Bridge to blockchain ein und führen Sie es aus, um eine sichere Verbindung zwischen einem IBM MQ Advanced -Warteschlangenmanager und IBM Blockchainherzustellen. Verwenden Sie die Bridge, um den Status einer Ressource in Ihrer Blockchain asynchron zu verbinden, zu suchen und zu aktualisieren, indem Sie eine Messaging-Anwendung verwenden, die eine Verbindung zu Ihrem IBM MQ Advanced-Warteschlangenmanager herstellt.

## Vorbereitende Schritte

- IBM MQ Bridge to blockchain ist nur für die Verbindung mit IBM MQ Advanced-Warteschlangenmanagern verfügbar.
- Der Warteschlangenmanager muss sich auf derselben Befehlsebene wie die Bridge befinden, z. B. IBM MQ 9.0.4.
- IBM MQ Bridge to blockchain wird für die Verwendung mit Ihrem Blockchain-Netz unterstützt, das auf Hyperledger Fabric 1.0 architecture 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 (formerly Bluemix) 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 Projectgemeinschaft entwickelt wird, einschließlich IBM als ursprünglicher Codeentwickler. Hyperledger Project oder Hyperledger ist 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 bereits IBM MQ Advanced und IBM Blockchain verwenden, können Sie die IBM MQ Bridge to blockchain verwenden, um einfache Abfragen, Aktualisierungen zu senden und Antworten von Ihrem Blockchain-Netz zu empfangen. Auf diese Weise können Sie Ihre lokale IBM-Software in einen Blockchain-Service der Cloud 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 IBM MQ Advanced-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.

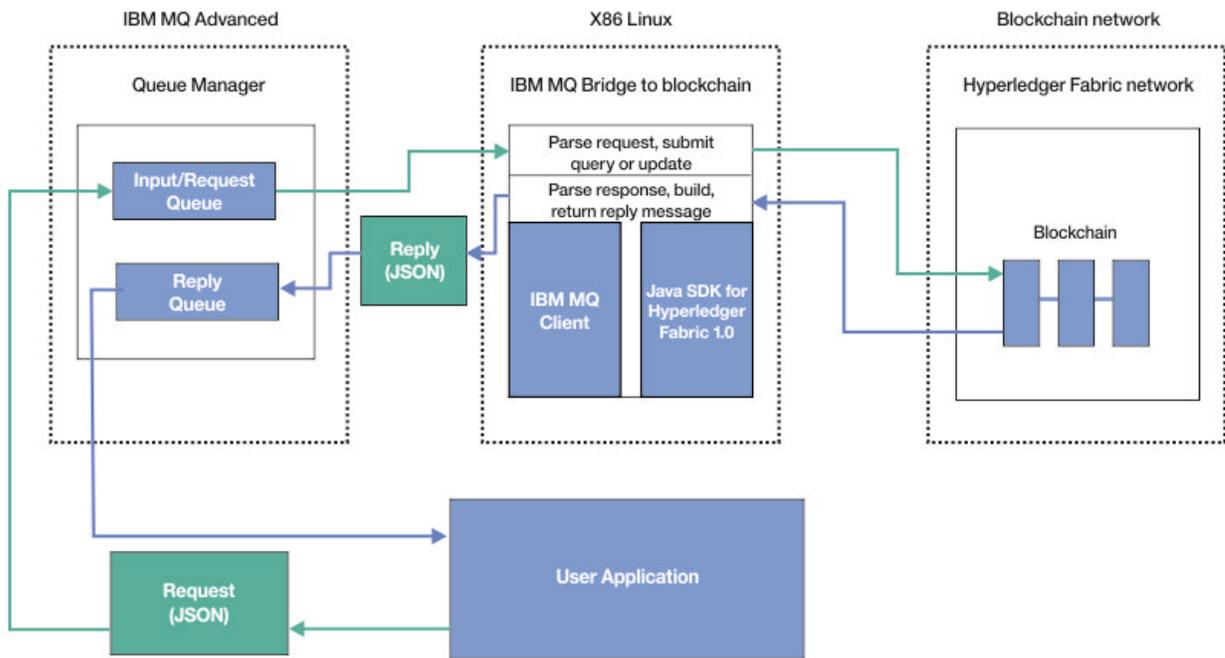


Abbildung 98. IBM MQ Bridge to blockchain

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:

- Benutzerberechtigungs-nachweise, z. B. SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE
- Nachrichteneingabe an der Brücke, z. B. APPL1.BLOCKCHAIN.INPUT.QUEUE
- Antworten von Blockketten, z. B. APPL1.BLOCKCHAIN.REPLY.QUEUE

Geben Sie im Verzeichnis `/opt/mqm/mqbc/samp` den folgenden Befehl 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 Ihre IBM MQ Advanced-Warteschlangenmanagerinformationen und die Berechtigungsnachweise aus Ihrem Blockchain-Netz, um eine Konfigurationsdatei für die IBM MQ Bridge to blockchain zu erstellen.

### V 9.0.4 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.
- 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 1.0 architecture basieren. Um die Brücke verwenden zu können, benötigen Sie Konfigurationen 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 688.
- Hyperledger Fabric-Netz, das in einem Kubernetes-Cluster in IBM Cloud (formerly Bluemix) ausgeführt wird. Weitere Informationen finden Sie unter [Entwickeln in einer Cloud-Sandbox auf IBM Blockchain Platform](#) und [„Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters“](#) auf Seite 690.

Weitere Informationen zu den Bedeutungen und Optionen für alle IBM MQ Bridge to blockchain-Parameter finden Sie im Befehl `runmqbcb`. Sie müssen Ihre eigenen Sicherheitsanforderungen berücksichtigen und die Parameter anpassen, die für Ihre Implementierung geeignet sind.

## 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 „6“ auf Seite 688 verwenden möchten, müssen Sie JNDI oder CCDT verwenden und **MQ CCDT URL** oder **JNDI implementation class** und **JNDI provider URL** entsprechend angeben.

```
Connection to Queue Manager
-----
Queue Manager                : [adv_qmgr_name]
Bridge Input Queue           : [APP1.BLOCKCHAIN.INPUT.QUEUE]
Bridge User Identity Queue   : [SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE]
MQ Channel                    : []
MQ Conname                    : []
MQ CCDT URL                   : []
JNDI implementation class     : []
JNDI provider URL            : []
MQ Userid                     : []
MQ Password                   : []
```

3. Geben Sie die Anmeldedetails für die Zertifizierungsstelle für Ihr Blockkettennetz ein.

Die Standardwerte für Ihre lokalen Hyperledger Fabric- und Kubernetes-Cluster-Beispiele sind *admin* für **Userid** und *adminpw* für **Enrollment Secret**. Wenn Sie diese Werte für Ihr Blockkettennetz geändert haben, stellen Sie sicher, dass Sie die richtigen Werte für die Konfiguration der Bridge verwenden.

```
Blockchain - User Identification
-----
Blockchain Userid             : []admin
Enrollment Secret            : []*****
```

4. Geben Sie die ID des Mitgliedschaftsservice-Providers (**MSPid**) ein, der die Mitgliedschaften und Identitätsregeln für Ihr Blockkettennetz regelt.

Geben Sie in Ihrer Berechtigungsnachweisdatei den Parameter **msp\_id** für den **Organisation Name** und den **Organisation MSPid** an. Verwenden Sie in „[Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungsdatei](#)“ auf Seite 688 den Wert **CORE\_PEER\_LOCALMSPID** aus dem Abschnitt **peer** der Datei. Verwenden Sie in der „[Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters](#)“ auf Seite 690 den Wert **mSPID**.

```
Blockchain - Organisation Identification
-----
Organisation Name             : []Org1MSP
Organisation MSPID            : []Org1MSP
```

5. Geben Sie die Standortwerte für die Blockkettennetzserver ein:

Geben Sie in „[Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungsdatei](#)“ auf Seite 688 die Namen und Server: Portpositionen für Zertifizierungsstellen-, Peer- und Anordnungsknoten-elemente an.

```
Blockchain server locations
-----
Certificate Authority servers : [ca.example.com Docker_container_host:7054] (for example ca.example.com localhost:7054)
Peer servers                  : [peer0 localhost:7051]
Orderer servers               : [orderer0 localhost:7050]
```

```
Peer Event servers : [peer0 localhost:7053]
Location of PEM file for Blockchain certificate : []
```

Geben Sie in „[Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters](#)“ auf Seite 690 die Namen und Server: Portpositionen für Zertifizierungsstellen-, Peer- und Anordnungsknoten-Elemente an.

```
Blockchain server locations
-----
Certificate Authority servers : [CA1 your_blockchain_network_pub
lic_ip_address:30000] (for example CA1 123.456.789.10:30000)
Peer servers : [blockchain-org1peer1 your_blockchain_network_pub
lic_ip_address:30110]
Orderer servers : [blockchain-orderer your_blockchain_network_pub
lic_ip_address:31010]
Peer Event servers : [blockchain-org1peer1 your_blockchain_network_pub
lic_ip_address:30111]
Location of PEM file for Blockchain certificate : []
```

#### 6. Geben Sie Zertifikate für TLS-Verbindungen ein.

Die Brücke fungiert als IBM MQ Java-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 Java-Clients. Die Konfiguration von TLS-Verbindungsdetails wird nur dann bereitgestellt, wenn Sie in Schritt „2“ auf Seite 687 JNDI- oder CCDT-Informationen angegeben haben.

```
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]
```

#### 7. Geben Sie die Position für die Protokolldatei für den IBM MQ Bridge to blockchain ein.

Sie müssen 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-Warteschlangenmanager herzustellen.

## Nächste Schritte

Führen Sie die in „[IBM MQ Bridge to blockchain ausführen](#)“ auf Seite 691 beschriebenen Schritte aus.

## Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungs nachweisen

Inhalt der .yaml-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.

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 686 die Konfigurationsdetails aus dieser Datei verwenden.

```
#
# Copyright IBM Corp All Rights Reserved
#
# SPDX-License-Identifier: Apache-2.0
#
version: '2'

networks:
  basic:

services:
  ca.example.com:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca.example.com
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-con
fig/ca.org1.example.com-cert.pem --ca.keyfile
/etc/hyperledger/fabric-ca-server-con
fig/f329434b83a06f32f17a300fef841cfd16ff58f3185fb744aae047207b01a9e_sk -b admin:adminpw -d'
    volumes:
      - ./crypto-config/peerOrganizations/org1.example.com/ca/:/etc/hyper
ledger/fabric-ca-server-config
      container_name: ca.example.com
    networks:
      - basic

  orderer.example.com:
    container_name: orderer.example.com
    image: hyperledger/fabric-orderer
    environment:
      - ORDERER_GENERAL_LOGLEVEL=debug
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_GENESIMETHOD=file
      - ORDERER_GENERAL_GENESISFILE=/etc/hyperledger/configtx/genesis.block
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/etc/hyperledger/msp/orderer/msp
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric/orderer
    command: orderer
    ports:
      - 7050:7050
    volumes:
      - ./config/:/etc/hyperledger/configtx
      - ./crypto-config/ordererOrganizations/example.com/orderers/orderer.exam
ple.com/:/etc/hyperledger/msp/orderer
      - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.exam
ple.com/:/etc/hyperledger/msp/peerOrg1
    networks:
      - basic

  peer0.org1.example.com:
    container_name: peer0.org1.example.com
    image: hyperledger/fabric-peer
    environment:
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_PEER_ID=peer0.org1.example.com
      - CORE_LOGGING_PEER=debug
      - CORE_CHAINCODE_LOGGING_LEVEL=DEBUG
      - CORE_PEER_LOCALMSPID=Org1MSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/peer/
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
      # the following setting starts chaincode containers on the same
      # bridge network as the peers
      # https://docs.docker.com/compose/networking/
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_basic
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb:5984
      # The CORE_LEDGER_STATE_COUCHDBCONFIG_
CONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
      # provide the
      credentials for ledger to connect to CouchDB. The username and password must
      # match the username and password set for the associated CouchDB.
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
```

```

- CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
working_dir: /opt/gopath/src/github.com/hyperledger/fabric
command: peer node start
# command: peer node start --peer-chaincodedev=true
ports:
- 7051:7051
- 7053:7053
volumes:
- /var/run/:/host/var/run/
- ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/hyperledger/msp/peer
- ./crypto-config/peerOrganizations/org1.example.com/users:/etc/hyperledger/msp/users
- ./config:/etc/hyperledger/configtx
depends_on:
- orderer.example.com
- couchdb
networks:
- basic

couchdb:
container_name: couchdb
image: hyperledger/fabric-couchdb
# Populate the COUCHDB_USER and COUCHDB_PASSWORD to set an admin user and password
# for CouchDB. This will prevent CouchDB from operating in an "Admin Party" mode.
environment:
- COUCHDB_USER=
- COUCHDB_PASSWORD=
ports:
- 5984:5984
networks:
- basic

cli:
container_name: cli
image: hyperledger/fabric-tools
tty: true
environment:
- GOPATH=/opt/gopath
- CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
- CORE_LOGGING_LEVEL=DEBUG
- CORE_PEER_ID=cli
- CORE_PEER_ADDRESS=peer0.org1.example.com:7051
- CORE_PEER_LOCALMSPID=Org1MSP
-
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
- CORE_CHAINCODE_KEEPALIVE=10
working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
command: /bin/bash
volumes:
- /var/run/:/host/var/run/
- ../chaincode:/opt/gopath/src/github.com/
- ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
networks:
- basic
#depends_on:
# - orderer.example.com
# - peer0.org1.example.com
# - couchdb

```

## Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters

Inhalt der Konfigurationsdatei von Ihrem Hyperledger Fabric-Blockchain-Netz, das in einem Kubernetes-Cluster in IBM Cloud (formerly Bluemix) ausgeführt wird, die Sie für die Konfiguration Ihres IBM MQ Bridge to blockchain verwenden können.

Nachdem Sie die Lernprogramme [IBM Blockchain Prepare and setup](#), [Simple install](#) und [Interacting with Ihre Blockchain](#) bearbeitet haben, sollte sich in Ihrem Verbindungsprofil-Ordner eine JSON-Datei befinden.

Wenn Sie eine Verbindung zu Ihrem Blockchain-Netz herstellen wollen, müssen Sie beim „Konfigurationsdatei für den IBM MQ Bridge to blockchain erstellen“ auf Seite 686 die Konfigurationsdetails aus dieser Datei verwenden.

```
{
  "name": "ibm-bc-org1",
  "description": "Connection profile for IBM Blockchain Platform",
  "type": "hlfv1",
  "orderers": [
    {
      "url": "grpc://INSERT_PUBLIC_IP:31010"
    }
  ],
  "ca": {
    "url": "http://INSERT_PUBLIC_IP:30000",
    "name": "CA1"
  },
  "peers": [
    {
      "requestURL": "grpc://INSERT_PUBLIC_IP:30110",
      "eventURL": "grpc://INSERT_PUBLIC_IP:30111"
    }
  ],
  "keyValStore": "INSERT_CREDENTIALS_PATH",
  "channel": "channel1",
  "mspID": "Org1MSP",
  "timeout": 300
}
```

## V 9.0.4 IBM MQ Bridge to blockchain ausführen

Führen Sie die IBM MQ Bridge to blockchain aus, um eine Verbindung zur IBM Blockchain und IBM MQ herzustellen. Wenn eine Verbindung besteht, ist die Brücke bereit, Abfragen zu verarbeiten und Nachrichten zu aktualisieren, sie an das 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 die IBM MQ Bridge to blockchain, um eine Verbindung zu Ihrem Blockchain-Netz und zu Ihrem IBM MQ Advanced-Warteschlangenmanager herzustellen.

Führen Sie den Befehl `bridge` aus.

```
runmqcb -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:

```
Fri Oct 06 06:32:11 PDT 2017 IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2023.

Fri Oct 06 06:32:17 PDT 2017 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.

```
runmqbc -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.

```
runmqbc -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 sich mit Ihrem Warteschlangenmanager und dem Blockchain-Netz verbunden.

## Nächste Schritte

- Führen Sie die Schritte in „IBM MQ Bridge to blockchain-Clientbeispiel ausführen“ auf Seite 846 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](#).

## V 9.0.4 Nachrichtenformate für IBM MQ Bridge to blockchain

Informationen zur Formatierung der Nachrichten, die von der IBM MQ Bridge to blockchain gesendet und empfangen werden.

Eine Anwendung fordert an, dass IBM MQ Bridge to blockchain eine Abfrage oder Aktualisierung von Informationen ausführt, die in der Blockchain gespeichert sind. Die Anwendung führt dies durch die Anforderung einer Anforderungsnachricht in die Brückenanforderungswarteschlange durch. Die Ergebnisse der Abfrage oder die Aktualisierung werden von der Brücke 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 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. Alle Felder mit Ausnahme von **args** sind erforderlich, die Argumentliste für dieses Feld erfordert Kenntnisse der Funktionen des gespeicherten Kettencodes.

## Anforderungsnachrichtenformat

Format der Eingabenachricht:

```
{ "function": functionName,
  "channel" : chainName,
  "chaincodeName" : codeName,
  "args" : [argument list]
}
```

Für das lokale Hyperledger-Netzbeispiel mit dem Beispiel für die Arbeit [Fabcar](#).

- Wenn Sie die Abfragenachricht verwenden möchten, die die Funktion `queryAllCars` in dem Chaincode 'fabcar' aufruft, die eine Liste von JSON-Objekten zurückgibt, die die in der Blockkette gehaltenen Autodetails darstellen, formatieren Sie die Nachricht wie folgt:

```
{ "function": "queryAllCars",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": []
}
```

Beispielantwort:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": [
    {"Record": {"owner": "Tomoko", "colour": "blue", "model": "Prius", "make": "Toyota"}, "Key": "CAR0"},
    {"Record": {"owner": "Brad", "colour": "red", "model": "Mustang", "make": "Ford"}, "Key": "CAR1"},
    {"Record": {"owner": "Jin Soo", "colour": "green", "model": "Tucson", "make": "Hyundai"}, "Key": "CAR2"},
    {"Record": {"owner": "Max", "colour": "yellow", "model": "Passat", "make": "Volkswagen"}, "Key": "CAR3"},
    {"Record": {"owner": "Adriana", "colour": "black", "model": "S", "make": "Tesla"}, "Key": "CAR4"},
    {"Record": {"owner": "Michel", "colour": "purple", "model": "205", "make": "Peugeot"}, "Key": "CAR5"},
    {"Record": {"owner": "Aarav", "colour": "white", "model": "S22L", "make": "Chery"}, "Key": "CAR6"},
    {"Record": {"owner": "Pari", "colour": "violet", "model": "Punto", "make": "Fiat"}, "Key": "CAR7"},
    {"Record": {"owner": "Valeria", "colour": "indigo", "model": "Nano", "make": "Tata"}, "Key": "CAR8"},
    {"Record": {"owner": "Shotaro", "colour": "brown", "model": "Barina", "make": "Holden"}, "Key": "CAR9"}
  ]
}
```

Die Antwortnachricht enthält alle Fahrzeugdatensätze, die derzeit in der Blockkette enthalten sind.

- Wenn Sie die Aktualisierungsnachricht verwenden möchten, die die Funktion `createCar` im Beispielchaincode 'fabcar' aufruft, die einen neuen Autoeintrag in der Blockchain ledger erstellt, formatieren Sie die Nachricht wie folgt:

```
{ "function": "createCar",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": ["CAR10", "Ford", "Mustang GT", "Blue", "Bob"]
}
```

Beispielantwort:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

Um zu überprüfen, ob der neue Autoeintrag in der Blockchain hinzugefügt wird, können Sie die Anfangsnachricht erneut verwenden, die alle Autos zurückgibt.

Für das Beispiel eines Kubernetes-Clusternetzes mit dem Arbeitsdemo [example02](#).

- Um die Abfragenachricht zu verwenden, die die Funktion `query` in dem chaincode `example02` aufruft, der den Wert für die Entität "a" in der Blockchain ledger zurückgibt, formatieren Sie die Nachricht wie folgt:

```
{ "function": "query",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a"]
}
```

Beispielantwort:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": "100"
}
```

- Um die Nachricht zu verwenden, die die invoke-Funktion example02 chaincode aufruft, die die im ersten Argument angegebene Entität verringert und die im zweiten Argument angegebene Entität um den Wert erhöht, der im dritten Argument angegeben ist, formatieren Sie die Nachricht wie folgt:

```
{ "function": "invoke",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a", "b", "10"]
}
```

Folgende Werte sind möglich:

- Vorher: a=100, b=200
- Nach: a=90, b=210

Beispielantwort:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

Wenn Sie die neuen Werte überprüfen möchten, müssen Sie eine neue Nachrichtenabfragenachricht übergeben, in der Sie nach Werten von **"a"** und **"b"** suchen.

## Antwortnachrichtenformat

Antwortnachrichten haben ihre Korrelations-ID auf die Nachrichten-ID der eingehenden Nachricht gesetzt. Alle vom Benutzer definierten Eigenschaften werden von der Eingabe in die Ausgabenachrichten kopiert. Die Benutzer-ID in der Antwort wird durch den Kontext der Identitätsgruppe auf die Benutzer-ID des Absenders gesetzt.

Ein Beispiel für eine erfolgreiche Verarbeitung:

```
{ "data": "500", "message": "OK", "statusCode": 200, "statusType": "SUCCESS" }
```

Die Antwortdaten in dieser Nachricht sind alle, die aus der chaincode-Antwort generiert werden (Byte, die in eine UTF-8-Zeichenfolge konvertiert wurden).

Alle Fehlerantworten haben die gleichen Felder, unabhängig davon, ob sie von der Brücke selbst, von den Aufrufen zur Blockkette oder vom Aufruf des Kettencodes generiert werden. Beispiel:

- Ungültiger Kanalname

```
{
  "message": "Bad newest block expected status 200 got 404, Chain myUnknownChannel",
  "statusCode": 404,
  "statusType": "FAILURE"
}
```

- Falsche JSON-Eingabenachricht

```
{
  "message": "Error: Cannot parse message contents.",
  "statusCode": 2110,
  "statusType": "FAILURE"
}
```

- Falsche Parameter in 'chaincode'

```
{
  "message": "Sending proposal to fabric-peer-1a failed because of gRPC failure=Status{code=UNKNOWN, description={\"Error\": \"Nil amount for c\"}, cause=null}",
  "statusCode": 500,
  "statusType": "FAILURE"
}
```

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 Eingabemessage 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**).

## V 9.0.4 IBM MQ Bridge to blockchain-Clientbeispiel ausführen

Sie können das mit dem IBM MQ Bridge to blockchain bereitgestellte JMS-Clientbeispiel verwenden, um eine Nachricht in die Eingabewarteschlange zu stellen, die von der Blockchain-Bridge geprüft wird, und die empfangene Antwort anzuzeigen.

### Vorbereitende Schritte

Ihre IBM MQ Bridge to blockchain ist aktiv und ist mit Ihrem IBM MQ Advanced-Warteschlangenmanager und Ihrem Blockchain-Netz verbunden und ist bereit, Eingabemessages zu verarbeiten.

### Informationen zu diesem Vorgang

Suchen Sie die JMS-Beispielanwendung im Verzeichnis samp der IBM MQ Bridge to blockchain.

### 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 die JSON-Anforderungsnachricht, die an die Bridge gesendet werden soll:

```
// Create the JSON request message.
// Modify "query", "exampleBlockchainChannelName", and "exampleChaincodeName" to
// match your deployed blockchain chaincode.
// The "operation" field is optional, but recommended. It should be set to QUERY
// or UPDATE to match what the chaincode is going to do.

JSONObject inputMsg = new JSONObject();
inputMsg.put("operation", "QUERY");

inputMsg.put("function", "query");
inputMsg.put("channel", "exampleBlockchainChannelName");
inputMsg.put("chaincodeName", "exampleChaincodeName");

// Create the JSON arguments for the request message.
// Modify "a" to match your deployed blockchain chaincode
// requirements, and add further arguments as necessary

JSONArray myArgs = new JSONArray();
myArgs.add("a");
inputMsg.put("args", myArgs);

TextMessage message = session.createTextMessage(inputMsg.serialize());
message.setJMSReplyTo(replyToQueue);
```

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_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar SimpleBCBClient.java
```

### 3. Führen Sie die kompilierte Klasse aus.

```
java -cp $MQ_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar:. SimpleBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Created the message. Starting the connection
Sent message:

JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 2
JMSDeliveryDelay: 0
JMSDeliveryTime: 1508427559117
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:414d5120424342514d2020202020202020209063e859ea36aa24
JMSTimestamp: 1508427559117
JMSCorrelationID: null
JMSDestination: queue:///APPL1.BLOCKCHAIN.INPUT.QUEUE
JMSReplyTo: queue:///APPL1.BLOCKCHAIN.REPLY.QUEUE
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 0
  JMSXUserID: USER1
  JMS_IBM_PutApplType: 6
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15391912
{"args":["a"],"function":"query","channel":"exampleBlockchainChannelName","operation":"QUERY","chaincodeName":"exampleChaincodeName"}
```

#### Antwortnachricht:

```
JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:c3e2d840e2e2f0f840404040404040d2afa27229838af2
JMSTimestamp: 1497439784000
JMSCorrelationID: ID:414d5120424342514d2020202020202020209063e859ea36aa24 *(JMSMessageID of
the input message)
JMSDestination: null
JMSReplyTo: null
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 1
  JMSXUserID: USER1
  JMS_IBM_Character_Set: UTF-8
  JMS_IBM_Encoding: 273
  JMS_IBM_Format: MQSTR
  JMS_IBM_MsgType: 8
  JMS_IBM_PutApplType: 2
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15392014
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
Response text:
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
SUCCESS
```

Wenn der Client eine Zeitlimitüberschreitung beim Warten auf die Antwort empfängt, überprüfen Sie, ob die Brücke aktiv ist.

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

### Vorbereitende Schritte

Lesen Sie vor der Konfiguration von IBM MQ die Informationen zu den Konzepten von IBM MQ for z/OS im Abschnitt [IBM MQ for z/OS-Konzepte](#).

Lesen Sie die Informationen zur Planung Ihrer IBM MQ for z/OS-Umgebung im Abschnitt zur [Planung Ihrer IBM MQ-Umgebung unter z/OS](#).

### 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 Tasks

[„Warteschlangenmanager auf Multiplatforms erstellen und verwalten“](#) auf Seite 5

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.

[„Verteilte Warteschlangensteuerung konfigurieren“](#) auf Seite 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

[„Verbindungen zwischen dem Server und dem Client konfigurieren“](#) auf Seite 17

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 Informationen

[IBM MQ for z/OS - Konzepte](#)

[Sicherung](#)

[IBM MQ for z/OS verwalten](#)

[Planung](#)

[Befehle absetzen](#)

[Dienstprogramme von IBM MQ for 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-Installationsbandes, die Programm- und Service-Level-Informationen für IBM MQ auf und beschreibt die Installation von IBM MQ for z/OS mit Hilfe des SMP/E-Programms (System Modification Program Extended). Die Programmverzeichnisse für IBM MQ for z/OS können von der [IBM Publications Center](#) heruntergeladen werden (siehe [IBM MQ 9.0 PDF-Dokumentation](#)).

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 konfigurieren“](#) auf Seite 702
- [„Warteschlangenmanager auf z/OS testen“](#) auf Seite 764
- [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

French

#### K

Japanese

#### 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 erstellen und ausführen wollen, die die Java Message Service (JMS) verwenden, um eine Verbindung zu IBM MQ for z/OS herzustellen, oder wenn Sie HTTP-Anwendungen erstellen und ausführen wollen, die HTTP für die Verbindung mit IBM MQ for z/OS verwenden.

#### **Y 9.0.1** IBM MQ for z/OS - Unix System Services-Webkomponenten

Diese Funktion ist optional.

Wählen Sie diese Funktion aus, wenn Sie die IBM MQ Console oder die REST API verwenden möchten.

Sie müssen die Funktion "IBM MQ for z/OSUNIX System Services-Komponenten" installieren, um diese Funktion zu installieren.

### Bibliotheken, die nach der Installation vorhanden sind

IBM MQ enthält eine Reihe separater Ladebibliotheken. [Tabelle 38 auf Seite 699](#) zeigt die Bibliotheken, die nach der Installation von IBM MQ möglicherweise vorhanden sind.

<i>Tabelle 38. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind</i>	
<b>Name</b>	<b>Beschreibung</b>
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.
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.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.

Tabelle 38. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind (Forts.)

Name	Beschreibung
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.
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.

Tabelle 38. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind (Forts.)

Name	Beschreibung
thlqual.SCSQPRLF	Enthält ISPF-Fenster, die in die ISPLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Französisch' für die Operationen und Bedienfelder von IBM MQ verwenden.
thlqual.SCSQPRLK	Enthält ISPF-Fenster, die in die ISPLIB-Verkettung miteinbezogen werden müssen, wenn Sie die Sprachkomponente 'Japanisch' für die Operationen und Bedienfelder von IBM MQ verwenden.
thlqual.SCSQPRLU	Enthält ISPF-Anzeigen, die in die ISPLIB-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.
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.

Tabelle 38. IBM MQ-Bibliotheken, die nach der Installation vorhanden sind (Forts.)

Name	Beschreibung
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

„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“ auf Seite 773

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 mit IMS verwenden“ auf Seite 805

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die eine Interaktion von IBM MQ mit IMS ermöglichen.

„IBM MQ mit CICS verwenden“ auf Seite 814

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 816

Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

### Zugehörige Verweise

„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen“ auf Seite 814

Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

### Zugehörige Informationen

[IBM MQ for z/OS - Konzepte](#)

[IBM MQ for z/OS verwalten](#)

## IBM MQ for z/OS konfigurieren

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 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. Die Programmver-

zeichnisse für IBM MQ for z/OS können von der [IBM Publications Center](#) heruntergeladen werden (siehe [IBM MQ 9.0 PDF-Dokumentation](#)).

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 821.

## 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.

Wenn Sie die z/OS-Systemparameter mithilfe von Befehlen ändern und [„SYS1.PARMLIB-Teildateien aktualisieren“](#) auf Seite 716 wie empfohlen ausführen, ist bei keiner dieser Tasks ein Start des z/OS-Systems über IPL erforderlich.

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

[„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“](#) auf Seite 773

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 mit IMS verwenden“](#) auf Seite 805

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die eine Interaktion von IBM MQ mit IMS ermöglichen.

[„IBM MQ mit CICS verwenden“](#) auf Seite 814

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 816

Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

### Zugehörige Verweise

[„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen“](#) auf Seite 814

Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

## Zugehörige Informationen

[IBM MQ for z/OS - Konzepte](#)

[IBM MQ for z/OS verwalten](#)

[Programmverzeichnis für IBM MQ for 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.

## 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:

### **„APF-Autorisierung der IBM MQ-Ladebibliotheken“ auf Seite 704:**

PROG=xx oder APF=aa verweist auf die Liste der berechtigten APF-Bibliotheken (Teildatei PROGxx oder IEFAPFaa)

### **„Aktualisieren der z/OS-Linkliste und LPA“ auf Seite 705:**

LNK=kk zeigt auf die Linkliste (Mitglied LNKLSTkk) LPA=mm verweist auf die LPA-Liste (Member LPALSTmm)

### **„Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften“ auf Seite 708:**

SCH=xx zeigt auf die Tabelle mit den Programmeigenschaften (PPT) (Member SCHEDxx)

### **„IBM MQ-Subsystem für z/OS definieren“ auf Seite 708:**

SSN=ss verweist auf die definierte Subsystemliste (Member IEFSSNss)

## APF-Autorisierung der IBM MQ-Ladebibliotheken

APF-Berechtigten 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 Verkettungszeichnungen der Datendefinitionsanweisung abzurufen (dies kann oft mehr E/A sein als die Programm-last selbst).

Verwenden Sie den Betriebssystembefehl "F LLA, REFRESH", nachdem Sie Änderungen an diesen Bibliotheken vorgenommen haben.

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 berechnen. Wenn Sie Advanced Message Security verwenden, müssen Sie auch die Bibliothek thlqual.SDRQAUTH mit APF berechnen.

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 705), 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. Weitere Informationen zu APF-Listen finden Sie im Handbuch [z/OS MVS Initialization and Tuning Reference](#).

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=h1q.SCSQ XXXX,VOLUME= YYYYYY ausgeben, damit die Änderungen wirksam werden: Dabei ist XXXX abhängig vom Bibliotheksnamen und wobei YYYYYY 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 705

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 697

Verwenden Sie dieses Thema, wenn Sie Ihre Warteschlangenmanager mit Details zu installierbaren Features, landessprachlichen Features und Informationen zu Tests anpassen und die Sicherheit konfigurieren.

## 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 9.0-Stufe aktualisieren, bevor Sie einen der Warteschlangenmanager in IBM MQ 9.0 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. Die Programmverzeichnisse für IBM MQ for z/OS können von der [IBM Publications Center](#) heruntergeladen werden (siehe [IBM MQ 9.0 PDF-Dokumentation](#)).

**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 hinzugefügt werden, damit IBM MQ als Subsystem agieren kann. 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 Benutzeränderung heißt CSQ8UERL und wird im *Programmverzeichnis für IBM MQ for z/OS* für Long Term Support oder Continuous Delivery beschrieben, das von der [IBM Publications Center](#) heruntergeladen werden kann.

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-Membren 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 708 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=thqua1.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqua1.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=thqua1.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=thqua1.SCSQLINK
```

Wenn Sie eine Wartung angewendet haben oder einen Warteschlangenmanager mit einer höheren Version oder einem höheren Release von IBM MQ erneut starten wollen, kann der Vorabcode für bereits definierte Warteschlangenmanagersubsysteme verfügbar gemacht werden. Führen Sie die folgenden Schritte aus, um sie verfügbar zu machen:

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 ist es vorzuziehen, die IBM MQ-Ladebibliotheken in die Linkliste zu stellen, die nach der Wartung aktualisiert werden kann, indem der z/OS-Befehl REFRESH LLA ausgegeben wird.

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 LNKLISTk-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.

## Zugehörige Konzepte

„Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften“ auf Seite 708

Für den IBM MQ-Warteschlangenmanager sind einige zusätzliche Einträge in der Programmeigententabelle (Program Properties Table, PPT) erforderlich.

## **z/OS Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften**

Für den IBM MQ-Warteschlangenmanager sind einige zusätzliche Einträge in der Programmeigententabelle (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 (CSQ4SCHED) 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/OS 1.12 und höheren Versionen ist CSQYASCP bereits für das Betriebssystem mit den Attributen definiert und muss nicht mehr in ein SCHEDxx-Member von PARMLIB aufgenommen 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.

Geben Sie den z/OS-Befehl SET SCH= aus, damit diese Änderungen wirksam werden.

### **Zugehörige Konzepte**

„IBM MQ-Subsystem für z/OS definieren“ auf Seite 708

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 712

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 716 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 710).

### **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 711).

Im Abschnitt Abbildung 99 auf Seite 709 finden Sie verschiedene Beispiele für IEFSSNss-Anweisungen.

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'  
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'  
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

Abbildung 99. 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 39](#) auf Seite 710 finden Sie eine Reihe von Beispielen, in denen die Zuordnungen von Subsystemnamen und Befehlspräfixzeichenfolgen, wie in den Anweisungen im Abschnitt [Abbildung 99](#) auf Seite 709 definiert, dargestellt werden.

<i>Tabelle 39. Subsystemname zu CPF-Zuordnungen</i>	
<b>IBM MQ-Subsystemname</b>	<b>CPF</b>
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.

#### *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 41](#) auf Seite 711 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 41](#) auf Seite 711 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 40](#) auf Seite 710.

<i>Tabelle 40. Beispiel für die CPF-Subset-und Superset-Regeln</i>		
<b>Subsystemname</b>	<b>CPF definiert</b>	<b>Weitergeleitete Befehle</b>
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 711 zu Informationen zur CPF-Registrierung).

In Tabelle 41 auf Seite 711 sind die Zeichen aufgeführt, die Sie bei der Definition von CPF-Zeichenfolgen verwenden können:

<i>Tabelle 41. Gültiger Zeichensatz für CPF-Zeichenfolgen</i>	
<b>Zeichensatz</b>	<b>Inhalt</b>
Buchstaben	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.

#### *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.

Der CPF wird bei der System-IPL-Zeit von IBM MQ bei z/OS registriert und bleibt während der gesamten 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.

Der CPF wird beim Start des IBM MQ-Subsystems bei z/OS registriert und bleibt so lange aktiv, bis das Subsystem von 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.

Der CPF wird bei der System-IPL-Zeit von IBM MQ bei z/OS registriert und bleibt während der gesamten 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.

## **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-Proze-

durbibliothek 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

Viele Beispiele und Anweisungen in dieser Produktdokumentation gehen davon aus, dass Sie ein Subsystem mit dem Namen CSQ1 haben. Diese Beispiele sind möglicherweise einfacher zu verwenden, wenn ein Subsystem mit dem Namen CSQ1 zu Beginn für Installationsprüfungs- und Testzwecke 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 Regionsgrößen unterhalb der 16-MB-Grenze oberhalb der 16-MB-Grenze und oberhalb des 2-GB-Balkens finden Sie unter [Empfohlene Regionsgrößen](#). 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.

Bei Verwendung von Advanced Message Security muss die STEPLIB-Verkettung *thlqual.SDRQAUTH* enthalten und APF-autorisiert sein.

**Anmerkung:** 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.

### Zugehörige Konzepte

„Prozeduren für den Kanalinitiator erstellen“ auf Seite 713

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.

### **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 Taskprozedur `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 Kanalinitiator task 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)`), verketteten Sie sie in der Datendefinitionsanweisung `STEPLIB`.

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`
- HFS-Datei, `/etc/resolv.conf`
- // 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 715](#)

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.

### **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 .

#### **Zugehörige Konzepte**

[„Konfiguration der Db2-Umgebung“ auf Seite 752](#)

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie die erforderlichen Db2-Objekte erstellen, indem Sie eine Reihe von Beispieljobs anpassen und ausführen.

### **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 Transport Layer Security (TLS) oder eine Sockets-Schnittstelle verwenden, stellen Sie sicher, dass die Benutzer-ID, unter deren Berechtigung der Kanalinitiator ausgeführt wird, für die Verwendung von UNIX System Services konfiguriert ist, wie in der Dokumentation zu *OS/390 UNIX System Services Planning* 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.

Weitere Informationen zu diesem Thema finden Sie unter [Tasks zur Sicherheitsinstallation für z/OS\(r\)](#).

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 716

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 756

Implementieren Sie Sicherheitsmaßnahmen für alle Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange für den Zugriff auf Db2 und die Listenstrukturen der Coupling-Facility.

### **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 708 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 704 verwendet werden
  - die Member LNKLSTkk und LPALSTmm, die im Abschnitt [„Aktualisieren der z/OS-Linkliste und LPA“](#) auf Seite 705 verwendet werden
  - das Member SCHEDxx, das im Abschnitt [„Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften“](#) auf Seite 708 verwendet wird
  - das Member IEFSSNss, das im Abschnitt [„IBM MQ-Subsystem für z/OS definieren“](#) auf Seite 708 verwendet wird

### Zugehörige Konzepte

„Passen Sie die Initialisierungseingabedatensätze an.“ auf Seite 716

Erstellen Sie Arbeitskopien der Eingabedatensätze für die Beispielinialisierung und passen Sie sie an Ihre Systemanforderungen an.

### **Passen Sie die Initialisierungseingabedatensätze an.**

Erstellen Sie Arbeitskopien der Eingabedatensätze für die Beispielinialisierung 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 Datensätze wird durch die DDnames CSQINP1, CSQINP2 und CSQINPT verwiesen, die in der gestarteten Taskprozedur des Warteschlangenmanagers definiert sind.

Antworten auf diese Befehle werden in die Initialisierungsausgabedatensätze geschrieben, die von den DDnames CSQOUT1, CSQOUT2 und CSQOUTT referenziert werden.

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

Der Datensatz thlqual.SCSQPROC enthält zwei Teildateien, die Definitionen von Pufferpools, Seitengruppe zu Pufferpoolzuordnungen und einen ALTER SECURITY-Befehl enthalten.

Member CSQ4INP1 verwendet eine Seitengruppe für jede Nachrichtenklasse. Member CSQ4INPR verwendet mehrere Seitengruppen für die Hauptklassen der Nachricht.

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 Verwalten von Pufferpools und Seitengruppen finden Sie im Handbuch [Planung für z/OS](#).

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 Taskprozedur für den Warteschlangenmanager aufgenommen werden können, mit einer Beschreibung ihrer Funktion. Die Namenskonvention lautet CSQ4INS\*. CSQ4INY \* muss 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.

<i>Tabelle 42. Mitglieder von thlqual.SCSQPROC</i>	
<b>Mitgliedsname</b>	<b>Beschreibung</b>
CSQ4INSG	Systemobjektdefinitionen.
CSQ4INSA	Systemobjekt und Standardregeln für die Kanalauthentifizierung.
CSQ4IN SX	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 erweiterte Nachrichtensicherheit.
CSQ4INSR	Passen Sie dieses Member an und schließen Sie es ein, wenn Sie WebSphere Application Server oder die Schnittstelle für eingereichtes Publish/Subscribe verwenden, die vom Dämon für eingereichtes Publish/Subscribe in IBM MQ V7 oder höher unterstützt 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'

Das Beispiel thlqual.SCSQPROC (CSQ4INPX) enthält eine Reihe 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. Weitere Informationen hierzu finden Sie im Abschnitt [Dienstprogramm CSQUTIL](#).

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 Member von thlqual.SCSQPROC aufgelistet, die in die CSQINPT-Verkettung Ihrer Taskprozedur für die gestartete WS-Manager aufgenommen werden können, mit einer Beschreibung ihrer Funktion.

<i>Tabelle 43. Mitglieder von thlqual.SCSQPROC</i>	
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. Weitere Informationen hierzu finden Sie im Abschnitt [Dienstprogramm CSQUTIL](#).

## Zugehörige Konzepte

„Erstellen Sie die Bootstrap- und Protokoll datengruppen.“ auf Seite 720

Verwenden Sie das mitgelieferte Programm CSQJU003, um die Bootstrap-Dateigruppen (BSDSs) und die Protokoll datengruppen vorzubereiten.

### **Erstellen Sie die Bootstrap- und Protokoll datengruppen.**

Verwenden Sie das mitgelieferte Programm CSQJU003, um die Bootstrap-Dateigruppen (BSDSs) und die Protokoll datengruppen vorzubereiten.

- Wiederholen Sie diese Tasks für jeden IBM MQ-Warteschlangenmanager.
- Bei der Migration von einer früheren Version muss diese Task nicht ausgeführt werden.

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 712 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 Protokoll datensätze 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 Protokoll datensätze 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 Protokoll datengruppen mit SHAREOPTIONS(2 3) definieren.

Informationen zum Planen von Bootstrap- und Protokoll datengruppen 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.

1. Benennen Sie unter Verwendung von **IDCAMS ALTER** die BSDSs im Format der Version 1 (erstellt mit dem Programm CSQJU003) in ++HLQ++.V1.BSDS01 um.

**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 8.0.0 handelt, muss er mit **OPMODE(NEWFUNC,800)** gestartet worden sein.
- Wenn es sich um einen Warteschlangenmanager in IBM MQ 9.0.0 LTS handelt, muss er mit **OPMODE(NEWFUNC,900)** oder **OPMODE(NEWFUNC,800)** worden sein.
- Wenn der Warteschlangenmanager IBM MQ 9.0.x CD, IBM MQ 9.1.0 LTS oder höher hat, muss er nur mit dieser Version gestartet worden sein.

### Zugehörige Konzepte

[„Definieren Sie Ihre Seitengruppen“ auf Seite 721](#)

Definieren Sie Seitengruppen für jeden Warteschlangenmanager mit einem der bereitgestellten Beispiele.

### **Definieren Sie Ihre Seitengruppen**

Definieren Sie Seitengruppen für jeden Warteschlangenmanager mit einem der bereitgestellten Beispiele.

- *Wiederholen Sie diese Tasks für jeden IBM MQ-Warteschlangenmanager.*
- *Bei der Migration von einer früheren Version muss diese Task nicht ausgeführt werden.*

Definieren Sie separate Seitengruppen für jeden IBM MQ-Warteschlangenmanager. thlqual.SCSQPROC (CSQ4PAGE) und thlqual.SCSQPROC (CSQ4PAGR) enthalten JCL- und AMS-Steueranweisungen, um Seitengruppen zu definieren und zu formatieren. 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 im Handbuch [Planung für z/OS](#).

Die in [„Prozeduren für den IBM MQ-Warteschlangenmanager erstellen“ auf Seite 712](#) beschriebene Prozedur für gestartete Tasks CSQ4MSTR bezieht sich auf die Seitengruppen, mit einer Anweisung in folgendem Format:

```
//CSQP00 nn DD DISP=OLD,DSN= xxxxxxxxxx
```

Hierbei steht *nn* für die Nummer der Seitengruppe zwischen 00 und 99, und *xxxxxxxxxx* 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. Dies wird in [IBM MQ for z/OS verwalten](#) beschrieben.
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 755](#)

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 724 beschrieben auf YES setzen. Wenn Sie diesen WS-Manager zum ersten Mal konfigurieren, führen Sie die gesamte Task aus.

Das Systemparametermodul verfügt über **V 9.0.3** vier Makros wie folgt:

<b>Makroname</b>	<b>Verwendungszweck</b>
CSQ6SYSP	Gibt die Verbindungs- und Tracerstellungsparameter an (siehe „ <a href="#">CSQ6SYSP verwenden</a> “ auf Seite 724).
CSQ6LOGP	Steuert die Protokollinitialisierung (siehe „ <a href="#">CSQ6LOGP verwenden</a> “ auf Seite 733).
CSQ6ARVP	Steuert die Archivinitialisierung (siehe „ <a href="#">CSQ6ARVP verwenden</a> “ auf Seite 737).
<b>V 9.0.3</b> <b>V 9.0.3</b> CSQ6USGP	Steuert die Nutzungserfassung (siehe „ <a href="#">CSQ6USGP verwenden</a> “ auf Seite 744)

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 im Abschnitt [Warteschlangenmanager starten und stoppen](#).

**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

Einige Systemparameter können während der Ausführung eines Warteschlangenmanagers geändert werden. Weitere Informationen finden Sie in den Befehlen [SET SYSTEM](#), [SET LOG](#) und [SET ARCHIVE](#).

Setzen Sie die SET-Befehle in Ihre Initialisierungseingabedatensätze, so dass sie bei jedem Starten des Warteschlangenmanagers wirksam werden.

### Zugehörige Konzepte

„Kanalinitiatorparameter anpassen“ auf Seite 745

Verwenden Sie ALTER QMGR, um den Kanalinitiator an Ihre Anforderungen anzupassen.

### 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 44 auf Seite 724](#) angezeigt. Wenn Sie einen dieser Werte ändern möchten, lesen Sie die ausführlichen Beschreibungen der Parameter.

Parameter	Beschreibung	Standardwert	SET, Befehl
ACELIM	Größe des ACE-Speicherpools in 1-KB-Blöcken.	0 (kein Grenzwert)	✓
CLCACHE	Gibt den Typ des zu verwendenden Cluster-Cache an.	STATISCH	-
CMDUSER	Die Standardbenutzer-ID für die Befehlssicherheitsüberprüfung.	CSQOPR	-
CONNSWAP	 <b>Achtung:</b> Ab IBM MQ 9.0 hat dieses Schlüsselwort keine Auswirkung. Legt fest, ob Jobs, die bestimmte IBM MQ-API-Aufrufe ausgeben, auslagerbar oder nicht auslagerbar sind.	YES	-
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. Daher ist die Verwendung des Parameters EXCLMSG zum Ausschließen von Nachrichten aus CPU-Sicht effizienter als die Verwendung der Methoden, die in „ <a href="#">Informationsnachrichten unterdrücken</a> “ auf Seite 750 beschrieben werden.	( )	✓
EXITLIM	Zeit (in Sekunden), für die WS-Manager-Exits während jedes Aufrufs ausgeführt werden können.	30	-
EXITTCB	Gibt an, wie viele gestartete Server-Tasks zum Ausführen von WS-Managerexits verwendet werden sollen.	8	-

Tabelle 44. Standardwerte für CSQ6SYSP-Parameter (Forts.)

Parameter	Beschreibung	Standardwert	SET, Befehl
LOGLOAD	Anzahl Protokollsätze, die von IBM MQ zwischen dem Anfang eines Prüfpunkts und dem nächsten geschrieben werden.	500 000	✓
MULCCAPT	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 <a href="#">Parameterbeschreibung</a>	-
OTMACON	OTMA-Verbindungsparameter.	Siehe <a href="#">Parameterbeschreibung</a>	-
QINDXBLD	Legt fest, ob der Neustart des Warteschlangenmanagers wartet, bis alle Indizes neu erstellt oder abgeschlossen sind, bevor alle Indizes erneut erstellt werden.	WARTEN	-
QMCCSID	ID des codierten Zeichensatzes für den WS-Manager.	Null	-
QSGDATA	Parameter der Gruppe mit gemeinsamer Warteschlange.	Siehe <a href="#">Parameterbeschreibung</a>	-
RESAUDIT	Prüfparameter RESLEVEL.	YES	-
ROUTCDE	Nachrichten-Routing-Code, der Nachrichten zugeordnet ist, die nicht von einer bestimmten Konsole angefordert wurden.	1	-
SERVICE	Reserviert für IBM.	0	✓
SMFACCT	Gibt an, ob SMF-Abrechnungsdaten beim Starten des Warteschlangenmanagers erfasst werden sollen.  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.  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	Standardzeit (in Minuten) zwischen den einzelnen Zusammenstellen von Statistiken.	30	✓
TRACSTR	Gibt an, ob die Tracefunktion automatisch gestartet werden soll.	Nein	-

Tabelle 44. 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	-

### 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.

Zur Begrenzung der vom Warteschlangenmanager zugeordneten Gesamtmenge an ECSA für thread-spezifische Steuerblöcke bei 5120Kist 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 (also die ersten beiden QSRSPHBT-Elemente addiert).

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. Weitere Informationen finden Sie im Abschnitt „WS-Manager-Cluster konfigurieren“ auf Seite 255.

#### 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.

## **CONNSWAP**

Legt fest, ob Stapeljobs, die bestimmte IBM MQ-API-Aufrufe ausgeben, für die Dauer der IBM MQ-API-Anforderung auslagerbar oder nicht auslagerbar sind. Folgende Werte sind möglich:

### **Nein**

Jobs sind während bestimmter IBM MQ-API-Aufrufe nicht auslagerbar.

### **YES**

Jobs sind während aller IBM MQ-API-Aufrufe auslagerbar.

Der Standardwert ist YES.

Verwenden Sie diesen Parameter, wenn Jobs mit niedriger Priorität ausgelagert werden und während der Auslagerung IBM MQ-Ressourcen binden, auf die womöglich andere Jobs oder Tasks warten.

IBM MQ betrachtet WebSphere Application Server als Teil einer RRSBATCH-Umgebung. Wenn das Schlüsselwort CONNSWAP verwendet wird, wird es auf jede Anwendung in einer BATCH- oder RRSBATCH-Umgebung angewendet. Das Schlüsselwort CONNSWAP gilt auch für TSO-Benutzer, ist jedoch auf CICS- und IMS-Anwendungen nicht anwendbar. CONNSWAP-Änderungen werden implementiert, wenn ein Recycle des Warteschlangenmanagers stattfindet. Nach der Schlüsselwortänderung ist ein Neustart erforderlich, da das Makro CSQ6SYSP erneut erstellt wird und der Warteschlangenmanager unter Verwendung des Lademoduls erneut gestartet wird, das durch das Makro aktualisiert wird.

Alternativ kann der Adressraum von WebSphere Application Server mit PPT als nicht auslagerbar festgelegt werden.

## **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 müs-

sen 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:

<b>Testsystem</b>	10 000
<b>Produktionssystem</b>	500 000

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).

### **Teildatei**

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 816.

### **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 Sicherheitscacheinstellungen 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.

### **QINDEXBLD**

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.

## **QSGDATA**

Daten der Gruppe mit gemeinsamer Warteschlange. Für dieses Schlüsselwort werden fünf positionsgewundene Parameter verwendet:

**QSGDATA = ( Qsgname , Dsgname , Db2name , Db2serv , Db2blob)**

### **Qsgname**

Dies ist der Name der Gruppe mit gemeinsamer Warteschlange, zu der der Warteschlangenmanager gehört.

Informationen zu den gültigen 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.

##### **YES**

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 *Nachrichtenbeschreibung* in einem der Datenträger des Handbuchs *z/OS MVS Routing and Descriptor Codes* .

#### **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.

##### **YES**

Die Erfassung von Abrechnungsdaten automatisch für die Standardklasse 1 starten.

##### **integers**

Eine Liste der Klassen, für die die Abrechnung automatisch im Bereich von 1 bis 4 erfasst wird.

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.

##### **YES**

Beginnen Sie, die Statistikdaten automatisch für die Standardklasse 1 zu erfassen.

##### **integers**

Eine Liste der Klassen, für die Statistikdaten automatisch im Bereich von 1 bis 4 erfasst werden.

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 Verwendung ist durch ein separat installiertes Produkt (Advanced Message Security (AMS)) lizenziert, durch das ein aktivierendes Modul in der Bibliothek SDRQAUTH bereitgestellt wird.

Die Sicherheitsrichtlinienverarbeitung wird für diesen Warteschlangenmanager aktiviert, indem SPLCAP mit einem der folgenden Werte konfiguriert wird:

#### **Nein**

Die Funktion zur Implementierung von Richtlinien zur Nachrichtensicherheit für Warteschlangen ist bei der Initialisierung des Warteschlangenmanagers nicht aktiviert.

#### **YES**

Nachrichtensicherheitsfunktionen werden während der Initialisierung des Warteschlangenmanagers aktiviert.

Wenn dieses Steuerelement festgelegt ist, versucht der Warteschlangenmanager, das Modul für die Lizenzberechtigung während der Initialisierung von SDRQAUTH zu laden und einen zusätzlichen Adressraum (AMSM) zu starten.

Der WS-Manager wird erst gestartet, wenn AMS lizenziert ist und die erforderliche Konfiguration für die Nachrichtensicherheit vorhanden ist.

Der Standardwert ist NEIN.

### **STATIZEIT**

Gibt die Standardzeit (in Minuten) zwischen aufeinanderfolgenden Erfassungen von Statistiken an.

Geben Sie einen Wert im Bereich zwischen 0 und 1440 an.

Bei Angabe des Wertes 0 werden sowohl statistische Daten als auch Abrechnungsdaten gesammelt. Informationen zum Festlegen dieser Einstellung finden Sie im Abschnitt [System Management Facility verwenden](#).

Der Standardwert ist 30.

### **TRACSTR**

Gibt an, ob die globale Traceverarbeitung automatisch gestartet werden soll.

Folgende Werte sind möglich:

#### **Nein**

Starten Sie die globale Tracefunktion nicht automatisch.

#### **YES**

Die globale Traceverarbeitung wird automatisch für die Standardklasse 1 gestartet.

#### **integers**

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 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 733](#)

Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

[„CSQ6ARVP verwenden“ auf Seite 737](#)

Verwenden Sie dieses Thema als Referenz für die Angabe Ihrer Archivierungsumgebung mit CSQ6ARVP.

## 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.

Parameter	Beschreibung	Standardwert	SET, Befehl
<a href="#">COMPLOG</a>	Steuert, ob die Protokollkomprimierung aktiviert ist.	KEINE	X
<a href="#">DEALLCT</a>	Zeitdauer, die eine Archivierungsbandeinheit nicht verwendet wird, bevor sie nicht zugeordnet wird.	Null	X
<a href="#">INBUFF</a>	Größe des Eingabepufferspeichers für aktive Dateien und Archivierungsprotokolldateien.	60 KB	-
<a href="#">MAXARCH</a>	Maximale Anzahl von Archivierungsprotokolldateienträgern, die aufgezeichnet werden können.	500	X
<a href="#">MAXCNOFF</a>	Maximale Anzahl der parallelen Auslastungstasks (CSQJOFF7), die parallel ausgeführt werden können.	31	-

Tabelle 45. Standardwerte für CSQ6LOGP-Parameter (Forts.)

Parameter	Beschreibung	Standardwert	SET, Befehl
<u>MAXRTU</u>	Maximale Anzahl dedizierter Bandeinheiten, die gleichzeitig zum Lesen von Archivprotokollbanddatenträgern zugeordnet sind.	2	X
<u>OFFLOAD</u>	Archivieren an oder aus.	JA (ON)	-
<u>OUTBUFF</u>	Größe des Ausgabepufferspeichers für aktive Dateien und Archivierungsprotokolldateien.	4 000 KB	-
<u>TWOACTV</u>	Einfache oder doppelte aktive Protokollierung.	JA (Dual)	-
<u>TWOARCH</u>	Einzel-oder Dual-Archive-Protokollierung.	JA (Dual)	-
<u>TWOBSDS</u>	Einzel-oder DoppelBSDS.	JA (duales BSBS)	-
<u>WRTHRSH</u>	Die Anzahl der Ausgabepuffer, die gefüllt werden sollen, bevor sie in die aktiven Protokolldateien geschrieben werden.	20	X
<u>ZHYWRITE</u>	Gibt an, ob die zHyperWrite-Funktion aktiviert ist.	Nein	-

#### 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

Setzen Sie diesen Wert auf das Maximum für die beste Protokollleseleistung.

### MAXARCH

Gibt die maximale Anzahl von Archivprotokolldateinträgern 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

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 Archivprotokoll-dateien 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:

#### **YES**

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:

<b>Testsystem</b>	400 KB
<b>Produktionssystem</b>	4 000 KB

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

#### **YES**

Doppelte aktive Protokolle

Der Standardwert ist YES.

Weitere Informationen zur Verwendung von einfacher und doppelter Protokollierung finden Sie im Abschnitt [IBM MQ-Ressourcen verwalten](#).

### **TWOICHE**

Gibt die Anzahl der Archivprotokolle an, die IBM MQ erstellt, wenn das aktive Protokoll ausgelagert wird.

Geben Sie Folgendes an:

#### **Nein**

Einzelne Archivprotokolle

#### **YES**

Doppelte Archivprotokolle

Der Standardwert ist YES.

Empfohlene Einstellungen:

<b>Testsystem</b>	Nein
<b>Produktionssystem</b>	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

#### YES

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 die zHyperWrite-Funktion aktiviert ist.

Folgende Werte sind möglich:

#### Nein

zHyperWrite ist nicht aktiviert.



**Achtung:** zHyperWrite ist in IBM MQ 9.0 nicht aktiviert, daher ist *NO* der einzige zulässige Wert.

### Zugehörige Verweise

„[CSQ6SYSP verwenden](#)“ auf Seite 724

Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

„[CSQ6ARVP verwenden](#)“ auf Seite 737

Verwenden Sie dieses Thema als Referenz für die Angabe Ihrer Archivierungsumgebung mit CSQ6ARVP.

 [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 46 auf Seite 738](#) 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/OS](#).

Tabelle 46. Standardwerte für CSQ6ARVP-Parameter

Parameter	Beschreibung	Standardwert	SET, Befehl
ALCUNIT	Einheiten, in denen primäre und sekundäre Bereichszuordnungen vorgenommen werden.	BLK (Blöcke)	X
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.	YES	X
BLKGRÖSS	Blockgröße der Archivprotokolldatei.	28 672	X
CATALOG	Gibt an, ob Archivierungsprotokolldatensätze in der ICF katalogisiert werden.	Nein	X
COMPACT	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

In einem Testsystem sind Archivprotokolle über lange Zeiträume wahrscheinlich nicht erforderlich.

**Produktionssystem** 9 999 (Standardwert)

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 *Nachrichtenbeschreibung* in einem der Datenträger des Handbuchs *z/OS MVS Routing and Descriptor Codes*.

### **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:

#### **YES**

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)

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 2 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.

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.

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

### YES

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** YES

**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 komprimiert werden sollen.

Geben Sie Folgendes an:

### Nein

Die Dateigruppen nicht komprimieren

### YES

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)

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 Protokoll Daten 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 Protokoll datengruppen.“ auf Seite 720 erläutert.
2. Ermitteln Sie die Anzahl der 4096-Byte-Blöcke in jedem Archivprotokollblock:

$$d = \text{BLKSIZE} / 4096$$

wobei BLKSIZE der aufgerundete Wert ist.

3. Wenn ALCUNIT = BLK:

$$\text{PRIQTY} = \text{INT}(c / d) + 1$$

wobei INT auf eine ganze Zahl abgerundet ist.

Wenn ALCUNIT = TRK:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}))) + 1$$

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:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}) * f)) + 1$$

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 Protokoll datengruppen.“ auf Seite 720 und „Definieren Sie Ihre Seitengruppen“ auf Seite 721.

Empfohlene Einstellungen:

**Testsystem** 1 680

Ausreichend, um das gesamte aktive Protokoll aufzunehmen, d. a.:

$$10\ 080 / 6 = 1\ 680 \text{ blocks}$$

**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 Bereichszuordnung für aktive Protokolle finden Sie im Abschnitt Planung für z/OS.

## PROTECT

Gibt an, ob Archivierungsprotokoll datensätze durch diskrete ESM-Profilen (ESM = External Security Manager) geschützt werden, wenn die Datensätze erstellt werden.

Geben Sie Folgendes an:

**Nein**

Profile werden nicht erstellt.

**YES**

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 finden Sie in den Handbüchern *z/OS MVS JCL Reference* und *z/OS MVS JCL User's 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.

**YES**

Die Namen enthalten einen Zeitstempel. Die Archivprotokolldateien werden wie folgt benannt:

```
arcpfxi.cyyddd.T hhmsst.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 yyyddd.T hhmsst.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 Einheitenamen der Einheit an, die zum Speichern der ersten Kopie der Archivprotokolldatei verwendet wird.

Geben Sie einen Einheitentyp oder einen Einheitenamen 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:

<b>Testsystem</b>	DASD
<b>Produktionssystem</b>	BAND

Weitere Informationen zur Auswahl einer Position für Archivprotokolle finden Sie im Abschnitt [Planung für z/OS](#).

## UNIT2

Gibt den Einheitentyp oder den Einheitenamen der Einheit an, die zum Speichern der zweiten Kopie der Archivierungsprotokolldatensätze verwendet wird.

Geben Sie einen Einheitentyp oder einen Einheitenamen 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 724

Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

„CSQ6LOGP verwenden“ auf Seite 733

Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

  *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 47 auf Seite 745](#) 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 47. 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

Gibt das Produkt an, für das die AMS-Verwendung aufgezeichnet werden soll, falls 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 724

Verwenden Sie dieses Thema als Referenz für das Festlegen von Systemparametern mit CSQ6SYSP.

„CSQ6LOGP verwenden“ auf Seite 733

Verwenden Sie dieses Thema als Referenz für die Angabe von Protokollierungsoptionen mit CSQ6LOGP.

### **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 ( $\text{MIN}(\text{MAXCHL} / \text{CHIDISPS}, 10)$ ), 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](#).

## Einrichten der z/OSUNIX System Services-Umgebung für Kanalinitiatoren

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(PROCUSER-MAX(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 747

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 Informationen

[Kanalinitiatorstatistikdatensätze](#)

## **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 MQRD\_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 Standard subsystemnamen CSQ1. Sie können diesen Namen für Test- und Installationsprüfung beibehalten. Bei Produktionssystemen 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 748

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.

### **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-Bibliotheksconfiguration 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 ISPLIB-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 ISPLIB-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`.
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)
```

Weitere Informationen zu thlqual und langletter finden Sie in Schritt „5“ auf Seite 749.

Weitere Informationen hierzu finden Sie im Handbuch *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 widerspiegeln, die Sie vorgenommen haben.

### Zugehörige Konzepte

„Teildatei für die Speicherauszugsformatierung in IBM MQ einschließen“ auf Seite 750

Damit IBM MQ-Speicherauszüge mit IPCS (Interactive Problem Control System) formatiert werden können, müssen Sie einige Systembibliotheken aktualisieren.

### **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. Ausführliche Informationen zu IPCSPARM finden Sie im Handbuch *z/OS MVS IPCS Customization*.

Sie müssen auch die Bibliothek thlqual.SCSQPNLA in Ihre ISPLIB-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 750

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.

### **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-Memberrn 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.

Beispiel: thlqual.SCSQPROC (CSQ4MPFL) zeigt empfohlene Einstellungen für MPFLSTxx an. Weitere Informationen zu MPFLSTxx finden Sie im Handbuch *z/OS MVS Initialization and Tuning Reference*.

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 Parametern finden Sie in der Dokumentation zu [MVS-Installationsexits](#).
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 724.

#### **Zugehörige Tasks**

„[Warteschlangenmanager auf z/OS testen](#)“ auf Seite 764

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.

## **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 722 zum Hinzufügen

von Daten zur Gruppe mit gemeinsamer Warteschlange lesen. Sie müssen den Parameter `CSQ6SYSP` ändern, um den Parameter `QSGDATA` anzugeben.

## **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 Db2 Version 11 und früheren Versionen. Wenn Sie eine neue Gruppe mit gemeinsamer Datennutzung mit Db2 V12 oder höher einrichten, werden Sie aufgefordert, die Jobs mit dem Präfix `CSQ4X` zu verwenden, da diese Jobs aktuellere Db2-Funktionen für dynamische Dimensionierung und Universal Table Spaces nutzen.

Sie müssen eine Umgebung einrichten, in der IBM MQ auf die Db2-Pläne, die für Gruppen mit gemeinsamer Warteschlange verwendet werden, zugreifen und diese ausführen kann.

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 `CSQ45CSG`  (oder `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 `CSQ45CDB`  (oder `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-Datennutzung-Gruppe herstellen.
3. Passen Sie die Beispiel-JCL `CSQ45CTS`  (oder `CSQ4XCTS`) an und führen Sie sie aus, um die Tabellenbereiche zu erstellen, die den Warteschlangenmanager und die Kanalinitiator tabellen enthalten, die für Gruppen mit gemeinsamer Warteschlange verwendet werden (in Schritt 1 erstellt werden).
4. Passen Sie die Beispiel-JCL `CSQ45CTB`  (oder `CSQ4XCTB`) an, und führen Sie sie aus, um die 12 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 für die angezeigt:

- **LTS** Long Term Support-Version in der Spalte LTS.
- **CD** Continuous Delivery-Version in der Spalte CD, wobei n für das CD-Release steht.

In jedem Release wird n um eins erhöht. In IBM MQ 9.0.3 ist CSQ5A90n beispielsweise CSQ5A903.

User	Pläne (LTS)	Pläne (CD)
Warteschlangenmanager	CSQ5A 900, CSQ5C 900, CSQ5D 900, CSQ5K 900, CSQ5L 900, CSQ5M 900, CSQ5P 900, CSQ5R 900, CSQ5S 900, CSQ5T 900, CSQ5U 900, CSQ5W 900	CSQ5A 90n, CSQ5C 90n, CSQ5D 90n, CSQ5K 90n, CSQ5L 90n, CSQ5M 90n, CSQ5P 90n, CSQ5R 90n, CSQ5S 90n, CSQ5T 90n, CSQ5U 90n, CSQ5W 90n
SDEFS-Funktion des Stapeldienstprogramms CSQUTIL	CSQ52 900	CSQ52 90n
CSQ5PQSG-und CSQJUCNV-Stapel-dienstprogramme	CSQ5B 900	CSQ5B 90n
CSQUZAP, Servicedienstprogramm	CSQ5Z 900	CSQ5Z 90n

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.
- CSQ45DTS **V 9.0.4** (oder CSQ4XDTS) , um die Tabellenbereiche zu löschen.
- CSQ45DDB **V 9.0.4** (oder CSQ4XDDB) , um die Datenbank zu löschen.
- CSQ45DSG **V 9.0.4** (oder 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 Db2 finden Sie unter [Db2 Administration](#) in *Db2 für z/OS 11.0.0*.

**V 9.0.4** Weitere Informationen zum Einrichten von Db2 finden Sie in [Db2 Administration](#) in *Db2 für z/OS 12.0.0*.

Informationen zu den Tabellengrößen von Db2 finden Sie in [Planung für z/OS](#).

### Zugehörige Konzepte

„Einrichten der Coupling-Facility“ auf Seite 754

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.

## **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.

- Wiederholen Sie diese Task für jede Gruppe mit gemeinsamer Warteschlange.
- 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.*

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 [Coupling-Facility-Ressourcen definieren](#).

### **Zugehörige Konzepte**

„Implementieren Sie Ihre ESM-Sicherheitskontrollen.“ auf Seite 715

Implementieren Sie die Sicherheitssteuerungen für Warteschlangenmanager und den Kanalinitiator.

## **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 Kapazitätsüberlegung 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 754

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, definieren Sie mithilfe von IXCMIAPI 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.

### **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 für 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 alle Gruppen mit gemeinsamer Warteschlange und Warteschlangenmanager aus, die Mitglied einer Gruppe mit gemeinsamer Warteschlange sein sollen. (Eine Beschreibung von CSQ5PQSG finden Sie im Abschnitt [IBM MQ für z/OS verwalten](#).)

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 722

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.

## **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 715

Implementieren Sie die Sicherheitssteuerungen für Warteschlangenmanager und den Kanalinitiator.

## **Advanced Message Security für z/OS konfigurieren**

In diesen Abschnitten finden Sie eine schrittweise Anleitung für die Konfiguration von Advanced Message Security.

## **Prozeduren für Advanced Message Security erstellen**

Jedes IBM MQ-Subsystem, das für die Verwendung von Advanced Message Security konfiguriert werden soll, erfordert eine katalogisierte Prozedur, um den AMS-Adressraum zu starten. Sie können eigene Prozeduren erstellen oder die von IBM bereitgestellte Prozedurbibliothek verwenden.

Für jedes IBM MQ-Subsystem, das so konfiguriert werden soll, dass Advanced Message Security verwendet wird, wird eine Kopie der Beispielprozedur CSQ4AMSM angepasst. Gehen Sie hierfür wie folgt vor:

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 Prozedur für die gestartete AMS-Task 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.

- Überprüfen und ändern Sie die Parameter, die an die AMS-Task übergeben werden, mit der Language Environment® \_CEE\_ENVFILE-Datei. In dem Beispiel thlqual.SCSQPROC (CSQ40ENV) werden die unterstützten Parameter aufgelistet.

**Anmerkung:** Diese Task sollte für jeden Warteschlangenmanager von IBM MQ wiederholt werden.

### **Richten Sie die gestartete Task Benutzer Advanced Message Security ein.**

Die Advanced Message Security-Task erfordert eine Benutzer-ID, die die Bekanntheit als UNIX System Services-Prozess ermöglicht.

Darüber hinaus müssen die Benutzer, in deren Namen die Task ausgeführt wird, auch über eine entsprechende Definition einer UNIX-UID (Benutzer-ID) und einer Gruppen-ID (Gruppen-ID) verfügen, damit diese Benutzer als Benutzer von UNIX System Services bekannt sind. Weitere Informationen zum Definieren von UNIX System Services-UIDs und -GIDs finden Sie im *z/OS: Security Server RACF Security Administrator's Guide*.

*z/OS: UNIX System Services Planning* vergleicht die traditionelle UNIX -Sicherheit mit der z/OS -Sicherheit. 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.

Abhängig von der Sicherheitsrichtlinie Ihrer Installation kann die Advanced Message Security-Task entweder mit der Superuserberechtigung (uid (0)) oder mit der RACF-Identität ausgeführt werden, die für die RACF FACILITY-Klasse BPX.DAEMON und BPX.SERVER zulässig ist, da diese Task in der Lage sein muss, die RACF-Identität der Benutzer zu übernehmen.

Wenn die letztere Methode verwendet wird oder Sie die Profile BPX.DAEMON oder BPX.SERVER bereits aktiviert haben, muss sich das Advanced Message Security-Taskprogramm (thlqual.SCSQAUTH (CSQ0DSRV)) in von RACF programmgesteuerten Bibliotheken befinden.

Informieren Sie sich in der Dokumentation *z/OS: UNIX System Services Planning*, 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 UNIX System Services-Prozesse zu implementieren und auszuführen.

Die Veröffentlichungen, die für diese Überprüfung nützlich sind, sind:

- *z/OS: UNIX System Services Planning*
- *z/OS: Security Server RACF Security Administrator's Guide*.

**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 Profilbehandlungsverarbeitung für die RACF-Klassen STARTED, FACILITY und SURROGAT und die generische Profilprüfung aktiviert haben. Weitere Informationen darüber, wie RACF generische Profile verarbeitet, finden Sie in der Dokumentation *z/OS: Security Server RACF Command Language Reference*.

- Definieren Sie zunächst RACF-Benutzerprofile für den Benutzer der gestarteten Task von Advanced Message Security. Hierbei kann es sich um denselben Benutzer handeln.

```
ADDUSER WMQAMSM NAME(' Advanced Message Security user') OMVS (UID(0)) DFLTGRP(group)
```

Wählen Sie eine Standardgruppe aus, die Ihren Installationsstandards entsprechend ist.

**Anmerkung:** Wenn Sie die USS-Superuser-Berechtigung (UID (0)) nicht erteilen möchten, müssen Sie die Advanced Message Security-Benutzer-ID für die BPX.DAEMON- und BPX.SERVER-Facility-Klassenprofile erlauben:

```
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 ausgewählten oder erstellten Benutzer-IDs an:

```
RDEFINE STARTED qmgr AMSM.* STDATA(USER(WMQAMSM))
```

Hierbei steht *qmgr* für den Namen des Präfix des Namens der gestarteten Task. Die gestarteten Tasks können z. B. CSQ1AMSM genannt werden. In diesem Fall würden Sie *qmgr* AMSM.\* durch CSQ1AMSM.\* ersetzen.

Die Namen der gestarteten Tasks müssen den Namen *qmgr* AMSM.\* haben.

4. Verwenden Sie den RACF-Befehl SETROPTS, um die gestarteten RACLISTed-Klassenprofile im Speicher zu aktualisieren:

```
SETROPTS RACLIST(STARTED) REFRESH
```

5. Die Advanced Message Security-Task übernimmt vorübergehend die Identität der Hostbenutzer-ID des Clientanforderers während der Zugriffsschutzverarbeitung von IBM MQ-Nachrichten. Daher ist es erforderlich, Profile in der Klasse SURROGAT für jede Benutzer-ID, die Anforderungen stellen kann, zu definieren.

Dies kann mit einem einzigen generischen Profil ausgeführt werden, wenn die RACF-Klasse SURROGAT aktiv ist. Die Prüfung wird ignoriert, wenn die SURROGAT-Klasse nicht aktiv ist. Die erforderlichen SURROGAT-Profile sind in der Dokumentation *z/OS: UNIX System Services Planning* 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. Generisches Profil einer Ersatzklasse definieren:

```
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(WMQASMS) ACCESS(UPDATE)
```

**Anmerkung:** Sie können spezifischere Profile definieren, wenn Sie bestimmte Benutzer einschränken möchten, die von der Advanced Message Security-Task verarbeitet werden sollen, wie in der Dokumentation *z/OS: UNIX System Services Planning* beschrieben.

- 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(WMQASMS) ACCESS(READ)
```

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.LISTRING CLASS(FACILITY) ID(WMQASMS) ACC(READ)
```

Alternativ können Sie dem Schlüsselring des Datenservice-Taskbenutzers in der Klasse RDATA LIB wie folgt Lesezugriff erteilen:

```
PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATA LIB) ID(WMQASMS) ACC(READ)
```

## Ressourcensicherheit für AMS

Der Benutzer der gestarteten Task benötigt die Leseberechtigung für die Warteschlange SYSTEM.PROTECTION.POLICY.QUEUE.

Der Benutzer der gestarteten Task benötigt die Berechtigung, eine Verbindung zum WS-Manager als BATCH-Anwendung herzustellen. Weitere Informationen finden Sie unter [Verbindungssicherheitsprofile für Stapelverbindungen](#).

## **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.

Geben Sie die entsprechende Benutzer-ID für diese Rolle an und erteilen Sie die Berechtigung, den Befehl RACDCERT zu verwenden. Beispiel:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

Dabei ist `admin` die Benutzer-ID Ihres Advanced Message Security -Sicherheitsadministrators.

## **z/OS Benutzern Ressourcenberechtigungen für Advanced Message Security erteilen**

Advanced Message Security-Benutzer benötigen relevante Ressourcenberechtigungen.

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 RDATA LIB
- Berechtigungen für ICSF-Klassen CSFSERV- und CSFKEYS-Profile

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 Handbuch *z/OS: Security Server RACF Security Administrator's Guide* enthält die detaillierte Vorgehensweise zum Definieren der Standard-OMVS-Segmente. 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.

Um allen Benutzern von Advanced Message Security die READ-Berechtigung für die Klassenfunktion IRR.DIGTCERT.LISTRING zu erteilen, müssen Sie folgenden Befehl ausgeben:

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
```

erteilen oder erteilen Sie die Berechtigung READ auf Benutzerbasis, indem Sie den folgenden Befehl ausgeben:

```
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 die Klasse RDATA LIB verwenden, um Zugriff auf bestimmte Schlüsselringe zu erteilen (die RDATA LIB-Berechtigungen haben Vorrang vor Berechtigungen von IRR.DIGTCERT.LISTRING). Beispiel:

```
PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(user) ACC(READ)
```

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 48. Erforderlicher Benutzerzugriff auf Profile der Klassen CSFSERV und CSFKEYS*

Klasse	Profil	Erlaubnis
CSFSERV	CSFDSG	READ
CSFSERV	CSFPKE	READ
CSFSERV	CSFPKD	READ
CSFSERV	CSFDSV	READ
CSFKEYS	ICSF-PKDS-Kennsatz	READ

## 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 643

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.

## Liberty-Serverdefinition erstellen

Wenn Sie die IBM MQ for z/OS Unix System Services-Webkomponenten installiert haben und die MQ Console oder die REST API verwenden möchten, müssen Sie die Liberty-Serverdefinition erstellen und anpassen.

### Vorbereitende Schritte

Sie müssen die SYSTEM.REST.REPLY.QUEUE erstellen, um den Liberty-Server verwenden zu können. Verwenden Sie hierfür das neueste **CSQ4INSG**-Beispiel in „Passen Sie die Initialisierungseingabedatensätze an.“ auf Seite 716.

### Informationen zu diesem Vorgang

- Sie müssen diese Task einmal für jedes z/OS-System ausführen, auf dem die MQ Console oder REST API ausgeführt werden soll.
- Sie benötigen einen Liberty-Server für jede ausgeführte IBM MQ-Version.
- Möglicherweise müssen Sie die Serverkonfiguration aktualisieren oder ändern, wenn Sie eine Migration von einer früheren Version durchführen.

Für die IBM MQ for z/OS Unix System Services-Webkomponenten muss ein einzelner Liberty-Server namens mqweb erstellt werden.

Die Serverkonfigurations- und -protokolldateien sind alle unter dem Liberty-Benutzerverzeichnis gespeichert.

Gehen Sie wie folgt vor, um die mqweb-Serverdefinition zu erstellen:

## Vorgehensweise

1. 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 dessen Inhalt. Da dieses Benutzerverzeichnis sowohl Protokolldateien als auch die Serverkonfiguration enthält, sollten Sie dieses Verzeichnis in einem separaten Dateisystem erstellen.
2. Stellen Sie sicher, dass Ihr aktuelles Verzeichnis PathPrefix/web/bin lautet. Dies ist die Position des Scripts **crtmqweb.sh**.  
PathPrefix ist der Installationspfad von IBM MQ UNIX System Services Components.
3. Erstellen Sie das Liberty-Benutzerverzeichnis mit der mqweb-Serverdefinitionsvorlage. Führen Sie hierfür das Script **crtmqweb.sh** aus.

**Anmerkung:** Das Script **crtmqweb.sh** akzeptiert einen optionalen Parameter - den Namen des Liberty-Benutzerverzeichnisses.

Wenn Sie keinen Namen für das Liberty-Benutzerverzeichnis angeben, wird der Standardwert `/var/mqm/web/installation1` verwendet.

4. Ändern Sie mit dem folgenden Befehl das Eigentumsrecht für die Verzeichnisse und Dateien im Liberty-Benutzerverzeichnis, sodass sie zu der Benutzer-ID und Gruppe gehören, unter der der mqweb-Server ausgeführt wird:

```
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 Liberty-Server erstellen“ auf Seite 762

### Zugehörige Tasks

„IBM MQ Console und REST API konfigurieren“ auf Seite 643

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 für den Liberty-Server erstellen**

Wenn Sie die IBM MQ for z/OS Unix System Services-Webkomponenten installiert haben und die MQ Console oder REST API verwenden möchten, müssen Sie eine katalogisierte Prozedur erstellen, um den Liberty-Server mqweb zu starten.

- Sie müssen diese Task einmal für jedes z/OS-System ausführen, auf dem IBM MQ ausgeführt werden soll.
- Sie benötigen eine Liberty-Serverinstanz für jede aktive Version von IBM MQ. Beispiel: Eine gestartete Task namens MQWB0901 für Warteschlangenmanager bei IBM MQ 9.0.1 und eine gestartete Task namens MQWB0902 für Warteschlangenmanager bei IBM MQ 9.0.2.

Wenn Sie nur einen Warteschlangenmanager haben, können Sie eine einzige gestartete Liberty-Server-task ausführen und die Bibliotheken ändern, die sie bei der Migration des Warteschlangenmanagers verwendet.

- 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 `th1qual.SCSQPROC (CSQ4WEBS)` in die Prozedurbibliothek.  
Benennen Sie das Verfahren gemäß den Standards Ihres Unternehmens.  
Beispiel: `MQWB0901`, um anzugeben, dass es sich um die katalogisierte Prozedur für Liberty für IBM MQ 9.0.1 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 angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.  
Weitere Informationen finden Sie im Abschnitt „[Liberty-Serverdefinition erstellen](#)“ auf Seite 761.
3. Autorisieren Sie die Prozedur, die unter Ihrem externen Sicherheitsmanager ausgeführt werden soll.
4. Verwenden Sie den Befehl `S procname`, um die Prozedur zu starten.  
Dadurch sollte die Nachricht `+ CWWKE0001I` erstellt werden: Der Server `mqweb` wurde gestartet.  
Wenn der Server nicht erfolgreich gestartet wird, überprüfen Sie die Nachrichten.  
Wenn die Prozedur gestartet wird, wird die Ausgabe in Dateien unter dem Parameter `USERDIR` gespeichert. Wenn das Benutzerverzeichnis z. B. `/u/mq/mqweb` ist, überprüfen Sie `/u/mq/mqweb/servers/mqweb/logs`.  
Die Dateien werden in ASCII geschrieben, so dass Sie Ihre normalen Systemtools verwenden können, um die Dateien anzuzeigen.
5. Verwenden Sie IBM Workload Manager (WLM), um diesen Adressraum zu klassifizieren.  
Der Liberty-Server ist eine IBM MQ-Anwendung und Benutzer interagieren mit dieser Anwendung. Die Anwendung muss in WLM nicht von großer Bedeutung sein, und es kann eine Serviceklasse von `STCUSER` geeignet sein.
6. Stoppen Sie die Prozedur mit dem Befehl `P procname`.

**Anmerkungen:**

- a. Stellen Sie sicher, dass Sie **Caps off** beim Bearbeiten des Mitglieds angeben, da die Datei Daten in Kleinschreibung enthält.
- b. Der Web-Server kann einige Zeit in Anspruch nehmen, um zum Beispiel mehr als eine Minute zu starten oder herunterzufahren.

## Nächste Schritte

[Benutzer und Rollen konfigurieren](#)

### Zugehörige Tasks

„[IBM MQ Console und REST API konfigurieren](#)“ auf Seite 643

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.

## **Zeilengruppe ReportingService (früher BluemixRegistration) konfigurieren**

Diese Task war Teil der Veröffentlichungsregistrierungs- und Nutzungsdaten für den IBM Cloud Product Insights-Service in IBM Cloud (formerly Bluemix). Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im Blogbeitrag [Service Deprecation: IBM Cloud Product Insights](#).

## Warteschlangenmanager auf z/OS testen

Wenn Sie Ihren Warteschlangenmanager angepasst oder migriert haben, können Sie ihn testen, indem Sie die Installationsprüfprogramme und einige der Beispieldanwendungen 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 Beispieldanwendungen 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 764
  - [„Gruppen mit gemeinsamer Warteschlange testen“](#) auf Seite 768
  - [„Tests für verteilte Steuerung von Warteschlangen“](#) auf Seite 769
  - [„Tests für C-, C++-, COBOL-, PL/I- und CICS-Programme mit IBM MQ for z/OS“](#) auf Seite 772

### Zugehörige Tasks

[„Warteschlangenmanager unter z/OS erstellen“](#) auf Seite 697

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

### Zugehörige Informationen

[IBM MQ for z/OS - Konzepte](#)

[IBM MQ-Umgebung unter z/OS planen](#)

[IBM MQ for z/OS verwalten](#)

## 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. Ausführliche Informationen zum Definieren von Ressourcen für RACF finden Sie im Abschnitt [Sicherheit unter 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	MQCMD5
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 100 auf Seite 766](#) 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 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 100. 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 einzeileine Informationsnachricht, falls zutreffend.

In [Abbildung 101 auf Seite 768](#) wird ein Beispielbericht gezeigt.

 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.



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. Ausführliche Informationen zum Definieren von Ressourcen für RACF finden Sie im Abschnitt [Sicherheit unter 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 102 auf Seite 769](#) 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 102. 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 766, um die Ergebnisse des IVP für Gruppen mit gemeinsamer Warteschlange zu prüfen.

## 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 Warteschlangen 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 Warteschlangenmanagers 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. Ausführliche Informationen zum Definieren von Ressourcen für RACF finden Sie im Abschnitt Sicherheit unter z/OS einrichten. Die Benutzer-ID, unter der der IVP ausgeführt wird, muss über die folgende Zugriffsberechtigung verfügen:

Berechtigung	Profil	Klasse
STEUERUNG	ssid.START.CHINIT und ssid.STOP.CHINIT	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQUEUE
UPDATE	ssid.SYSTEM.CSQUTIL.*	MQUEUE
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 103 auf Seite 771 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	MQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.INITQ	MQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.SYNCQ	MQUEUE
ALTER	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQUEUE
UPDATE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQUEUE
ALTER	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQUEUE
STEUERUNG	ssid.CONTEXT.**	MQADMIN

Die RACF-Befehle dazu werden auch in Abbildung 103 auf Seite 771 angezeigt.

```

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 103. 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 104 auf Seite 772](#) 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 CSQXRQDM Channel initiator active
CSQX002I +CSQ1 CSQXRQDM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRQDM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRQDM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRQDM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRQDM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRQDM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRQDM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRQDM TCP/IP system name is TCP/IP
CSQX848I +CSQ1 CSQXRQDM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRQDM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRQDM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRQDM 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 104. 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 für IBM MQ for z/OS](#).

## Kommunikation mit anderen Warteschlangenmanagern konfigurieren

In diesem Abschnitt werden die Vorbereitungen für IBM MQ für z/OS beschrieben, die Sie vor der Verwendung der verteilten Steuerung von Warteschlangen ausführen müssen.

Zum Definieren der Anforderungen für die verteilte Steuerung von Warteschlangen müssen Sie die folgenden Elemente definieren:

- Definieren Sie die Kanalinitiatorprozeduren und -datensätze.
- Definieren Sie die Kanaldefinitionen.
- Warteschlangen und andere Objekte definieren
- Zugriffsschutz definieren

Um die verteilte Steuerung von Warteschlangen zu aktivieren, müssen Sie die folgenden drei Tasks ausführen:

- 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 697 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 794 ein.

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, lesen Sie den Abschnitt [Verteilte Warteschlangen- und Warteschlangen-Sharing-Gruppen](#).

In den folgenden Abschnitten finden Sie weitere Hinweise zur Verwendung der verteilten Steuerung von Warteschlangen mit IBM MQ für z/OS.

### 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 Stufe anzugeben. Wenn der Kanalinitiator den Befehl verarbeitet hat, werden weitere Nachrichten angezeigt, die den Erfolg des Befehls angeben oder andernfalls zusammen mit der Nachricht CSQ9022I oder CSQ9023I an den Befehlsaussteller gesendet werden. Alle generierten Fehlernachrichten können auch an die z/OS-Konsole gesendet werden.

Alle Clusterbefehle mit Ausnahme des Befehls 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 anzeigt, dass eine Anforderung gesendet wurde. Auf diese Nachricht folgt die Nachricht CSQ9022I, um anzugeben, 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 Mitgliedern 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 im Abschnitt [The dead-letter queue handler utility \(CSQUDLQH\)](#).

## 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 hierzu finden Sie im Abschnitt [Sicherheitsaspekte für den Kanalinitiator unter z/OS, Profile für die Warteschlangensicherheit](#) und [„Implementieren Sie Ihre ESM-Sicherheitskontrollen.“](#) auf Seite 715.)

## 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 LSTRTMR angegebenen Intervall erneut zu versuchen. Diese Versuche können fortgesetzt werden, bis TCP/IP zurückkehrt und das Empfangsprogramm automatisch erneut gestartet wird. Informationen zu LSTRTMR finden Sie unter [ALTER QMGR](#) und [Distributed queuing messages \(CSQX ...\)](#).

## Kommunikation gestoppt-LU6.2

Wenn APPC gestoppt wird, wird auch der Listener gestoppt. In diesem Fall wiederholt das Empfangsprogramm in diesem Fall automatisch erneut das Intervall für LSTRTMR, so dass, wenn APPC erneut gestartet wird, das Empfangsprogramm 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 konfigurieren“](#) auf Seite 702

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

[„Kanäle unter z/OS überwachen und steuern“](#) auf Seite 776

Mit den DQM-Befehlen und -Anzeigen können Sie die Kanäle zu fernen Warteschlangenmanagern erstellen, überwachen und steuern.

[„Kommunikation für z/OS konfigurieren“](#) auf Seite 794

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 799  
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 804

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 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

## IBM MQ-Objekte definieren

Verwenden Sie eine der Befehlseingabemethoden von IBM MQ, um IBM MQ-Objekte zu definieren. Weitere Informationen zum Definieren dieser Objekte finden Sie in den Informationen in diesem Abschnitt.

Informationen zum Definieren von Objekten finden Sie in „[Kanäle unter z/OS überwachen und steuern](#)“ auf Seite 776.

## 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. Beispiel:

```
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 'CSQ4INXD' enthält zusätzliche Beispiele für die erforderlichen Definitionen.

 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 verfügbar ist (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 Kanalbefehlwarteschlange 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 WS-Manager stoppen müssen, können Sie den MQSC-Befehl STOP CHINIT verwenden.

## Kanäle unter 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 175.

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 Kanalsteuerungsprogramm 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 gemeinsamen 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:

<b>Task, die ausgeführt werden soll</b>	<b>MQSC-Befehl</b>
<a href="#">Kanal definieren</a>	<a href="#">DEFINE CHANNEL</a>
<a href="#">Kanaldefinition ändern</a>	<a href="#">ALTER CHANNEL</a>
<a href="#">Kanaldefinition anzeigen</a>	<a href="#">ANZEIGEN CHANNEL</a>
<a href="#">Kanaldefinition löschen</a>	<a href="#">DELETE CHANNEL</a>
<a href="#">Kanalinitiator starten</a>	<a href="#">START CHINIT</a>
<a href="#">Kanalinitiator stoppen</a>	<a href="#">STOP CHINIT</a>
<a href="#">Informationen zum Kanalinitiator anzeigen</a>	<a href="#">ANZEIGEN CHINIT</a>
<a href="#">Kanallistener starten</a>	<a href="#">START LISTENER</a>

<i>Tabelle 49. Kanaltasks (Forts.)</i>	
<b>Task, die ausgeführt werden soll</b>	<b>MQSC-Befehl</b>
<u>Kanallistener stoppen</u>	STOP LISTENER
<u>Kanal starten</u>	START CHANNEL
<u>Kanal testen</u>	<u>Pingkanal</u>
<u>Nachrichtenfolgenummern für einen Kanal zurücksetzen</u>	<u>Kanal zurücksetzen</u>
<u>Unbestätigte Nachrichten in einem Kanal auflösen</u>	<u>Auflösungskanal</u>
<u>Kanal stoppen</u>	STOP CHANNEL
<u>Kanalstatus anzeigen</u>	ANZEIGEN CHSTATUS
<u>Clusterkanäle anzeigen</u>	DISPLAY CLUSQMGR

### Zugehörige Konzepte

„Verwenden der Anzeigen und der Befehle“ auf Seite 778

Sie können die MQSC-Befehle, die PCF-Befehle oder die Operationen und Steuerkonsolen verwenden, um DQM zu verwalten.

„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“ auf Seite 773

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 konfigurieren“ auf Seite 702

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

„Kommunikation für z/OS konfigurieren“ auf Seite 794

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 799

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 804

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 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

## **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 zur Syntax der MQSC-Befehle finden Sie im Abschnitt Scriptbefehle (MQSC). Informationen zu PCF-Befehlen finden Sie im Abschnitt Einführung in Programmierbare Befehlsformate.

## 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 [verwalten IBM MQ for z/OS](#) und in den untergeordneten Abschnitten. [Abbildung 105](#) auf Seite 779 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, 2023. All rights reserved.

Command ==>-----
F1=Help F2=Split F3=Exit F4=Prompt F9=SwapNext F10=Messages
F12=Cancel
```

*Abbildung 105. 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 106](#) auf Seite 780 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 106. Kanäle auflisten

## Kanal unter z/OS definieren

Unter z/OS können Sie einen Kanal mit MQSC-Befehlen oder über die Operationen und Steuerkonsolen definieren.

Wenn Sie einen Kanal mit den MQSC-Befehlen definieren möchten, verwenden Sie [DEFINE 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	2 (Definieren wie)
Objektyp	Kanaltyp (z. B. SENDER) oder CHANNEL
Name	
Disposition	Die Position des neuen Objekts.

In einigen Anzeigen werden Informationen zu dem Namen und den Attributen angezeigt, die für den Kanal, den Sie definieren, angezeigt werden sollen. Sie werden mit den Standardattributwerten initialisiert. Ändern Sie alle gewünschten Änderungen, bevor Sie die Eingabetaste drücken.

**Anmerkung:** Wenn Sie im Feld **object type** den Befehl CHANNEL eingegeben haben, werden Sie zuerst die Anzeige "Gültiges 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.

Nachdem Sie Ihren Kanal definiert haben, müssen Sie den Kanal sichern (siehe [„Kanal schützen“](#) auf Seite 782).

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	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](#).

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	1 (Liste oder Anzeige)
Objekttyp	Kanaltyp (z. B. SENDER) oder CHANNEL
Name	CHANNEL.TO.DISPLAY
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](#).

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	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:

<b>Feld</b>	<b>Value</b>
Aktion	1 (Anzeigen)
Objekttyp	SYSTEM
Name	keinen

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:

<b>Feld</b>	<b>Wert</b>
Action	8

Sie werden mit einem Editor dargestellt, in dem Sie einen MQSC-Befehl angeben können, in diesem Fall ein Befehl CHLAUTH, siehe [Abbildung 107](#) auf Seite 782. 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 107. Befehlseingabe

Die Ausgabe des Befehls wird dann an Sie übergeben, siehe [Abbildung 108](#) auf Seite 783.

```
***** ***** Top of Data *****
000001 CSQU000I CSQUTIL IBM MQ for z/OS 7.1.0
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 CSQMCA ' 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 108. Befehlsausgabe

### **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	Value
Aktion	6 (Start)
Objekttyp	SYSTEM
Name	keinen

Die Anzeige "Systemfunktion starten" wird aufgerufen. Der Text im Anschluss an die folgende Anzeige erläutert, welche Aktion zu ergreifen ist:

## Systemfunktion starten

Wählen Sie den Funktionstyp aus, füllen Sie die Felder aus und drücken Sie die Eingabetaste, um das

.

```
Funktionstyp..... _ 1. Kanalinitiator
2. Kanallistener
Aktionswarteschlangenmanager... : MQ25
```

### Kanalinitiator

```
JCL-Substitution..... _____
_____
```

### Kanallistener

```
Eingehende Disposition... Q G=Gruppe, Q = Qmgr
Transportart..... _ L=LU6.2, T = TCP/IP
LU-Name (LU6.2)..... _____
Portnummer (TCP/IP)... 1414
IP-Adresse (TCP/IP)... _____
```

Befehl == = > \_\_\_\_\_

F1 = Hilfe F2 = Teilen F3 = Verlassen F9 = SwapNext F10 = Nachrichten F12 = Abbrechen

Abbildung 109. Systemfunktion starten

Wählen Sie Funktionsart 1 (Kanalinitiator) aus, und drücken Sie die Eingabetaste.

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	7 (Stoppen)
Objekttyp	SYSTEM
Name	keinen

Die Anzeige "Systemfunktion stoppen" wird angezeigt. Der Text im Anschluss an die Anzeige erläutert, wie Sie diese Anzeige verwenden:

Systemfunktion stoppen

Wählen Sie den Funktionstyp aus, füllen Sie die Felder aus und drücken Sie die Eingabetaste, um das

.

Funktionstyp..... \_ 1. Kanalinitiator  
2. Kanallistener  
Aktionswarteschlangenmanager... : MQ25

Kanalinitiator  
Gemeinsam genutzte Kanäle erneut starten Y J=Ja, N = Nein

Kanallistener  
Eingehende Disposition... Q G=Gruppe, Q = Qmgr  
Transportart..... \_ L=LU6.2, T = TCP/IP

Portnummer (TCP/IP)... \_\_\_\_\_  
IP-Adresse (TCP/IP)... \_\_\_\_\_

Befehl == = > \_\_\_\_\_  
F1 = Hilfe F2 = Teilen F3 = Verlassen F9 = SwapNext F10 = Nachrichten F12 = Abbrechen

Abbildung 110. 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.

### 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	Value
Aktion	6 (Start)
Objekttyp	SYSTEM
Name	keinen

Die Anzeige "Systemfunktion starten" wird angezeigt (siehe [Abbildung 109](#) auf Seite 784).

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.

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	7 (Stoppen)
Objektyp	SYSTEM
Name	keinen

Die Anzeige "Systemfunktion stoppen" wird angezeigt (siehe [Abbildung 110](#) auf Seite 785).

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.

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	6 (Start)
Objektyp	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:

Kanal starten

Wählen Sie Disposition aus und drücken Sie die Eingabetaste, um den Kanal zu starten

```
Kanalname..... : CHANNEL.TO.USE
Kanaltyp..... : SENDER
Beschreibung..... : Beschreibung von CHANNEL.TO.USE
```

```
Disposition..... P P = Privat auf MQ25
S = Gemeinsam genutzt in MQ25
A = Gemeinsam genutzt auf jedem WS-Manager
```

```
Befehl == = > -----
F1 = Hilfe F2 = Teilen F3 = Verlassen F9 = SwapNext F10 = Nachrichten F12 = Abbrechen
```

Abbildung 111. 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.

### **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.

## **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:

<b>Feld</b>	<b>Value</b>
Aktion	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:

Kanalfunktion ausführen

Wählen Sie den Funktionstyp aus, füllen Sie die Felder aus und drücken Sie die Eingabetaste

Funktionstyp..... 1. Zurücksetzen 3. Mit Commit auflösen  
2. Ping 4. Mit Backout auflösen

Kanalname..... : CHANNEL.TO.USE  
Kanaltyp..... : SENDER  
Beschreibung..... : Beschreibung von CHANNEL.TO.USE

Disposition..... P P = Privat auf MQ25  
S = Gemeinsam genutzt in MQ25  
A = Gemeinsam genutzt auf jedem WS-Manager

Folgenummer für Zurücksetzen.. 1 1-99999999  
Datenlänge für Pingsignale.... 16 16-32768

Befehl == > \_\_\_\_\_  
F1 = Hilfe F2 = Teilen F3 = Verlassen F9 = SwapNext F10 = Nachrichten F12 = Abbrechen

### *Abbildung 112. 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.

## **Nachrichtenfolgennummern für einen Kanal zurücksetzen**

Sie können Nachrichtenfolgennummern für einen Kanal mit Hilfe von MQSC-Befehlen oder mit den Operationen und Steuerkonsolen zurücksetzen.

Verwenden Sie RESET CHANNEL, um Kanalfolgennummern 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:

<b>Feld</b>	<b>Value</b>
Aktion	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 112 auf Seite 788](#)).

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.

### **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:

<b>Feld</b>	<b>Value</b>
Aktion	5 (Ausführen)
Objekttyp	SENDER, SERVER oder CHANNEL
Name	CHANNEL.TO.USE
Disposition	Die Disposition des Objekts.

Die Anzeige "Kanalfunktion ausführen" wird angezeigt (siehe [Abbildung 112 auf Seite 788](#)).

Wählen Sie Funktionstyp 3 oder 4 aus (Auflösung mit Commit oder Backout). (Weitere Informationen finden Sie in „Unbestätigte Kanäle“ auf Seite 196.)

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.

### **-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:

<b>Feld</b>	<b>Value</b>
Aktion	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:

Kanal stoppen

Füllen Sie die Felder aus und drücken Sie die Eingabetaste, um den Kanal zu stoppen

```
Kanalname..... : CHANNEL.TO.USE
Kanaltyp..... : SENDER
Beschreibung..... : Beschreibung von CHANNEL.TO.USE

Disposition..... P P = Privat auf MQ25
A = Gemeinsam genutzt auf jedem WS-Manager

Stoppmodus..... 1 1. Quiescemosus 2. Erzwingen
Status stoppen..... 1 1. Gestoppt 2. Inaktiv

Warteschlangenmanager..... -----
Verbindungsname..... -----

Befehl == > -----
F1 = Hilfe F2 = Teilen F3 = Verlassen F9 = SwapNext F10 = Nachrichten F12 = Abbrechen
```

Abbildung 113. -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 Quiescemosus versetzt“ auf Seite 194. Informationen zum erneuten Starten von gestoppten Kanälen finden Sie in „Gestoppte Kanäle erneut starten“ auf Seite 195.

**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.

## **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 106 auf Seite 780](#)) wird für jeden Kanal eine Zusammenfassung des Kanalstatus wie folgt angezeigt:

INACTIVE	Es sind keine Verbindungen aktiv.
<i>status</i>	Eine Verbindung ist aktiv
<i>nnn status</i>	Mehr als eine Verbindung ist aktuell und alle aktuellen Verbindungen haben denselben Status.
<i>nnn</i> CURRENT	Es ist mehr als eine Verbindung vorhanden und die aktuellen Verbindungen haben nicht alle denselben Status.
keinen	IBM MQ kann nicht feststellen, wie viele Verbindungen aktiv sind (z. B., weil der Kanalinitiator nicht aktiv ist)

**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
START	STARTING
AUSFUEF	RUNNING
STOP	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 114 auf Seite 792](#)).

Listkanäle-Aktueller Status-MQ25 Zeile 1 von 16

Aktionscodes eingeben und die Eingabetaste drücken. F11 drücken, um den gesicherten Status anzuzeigen.

1 = Aktu. Status anzeigen

```
Kanalname Verbindungsname Status
Startzeitnachrichten Letzte Nachrichtenart '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
***** Ende der Liste *****
Befehl == = > -----
F1 = Hilfe F2 = Aufteilen F3 = Verlassen F4 = Filter F5 = Aktualisieren F7 = Bkwd
F8 = Fw F9 = SwapNächstes F10 = Nachrichten F11 = Gespeichert F12 = Abbrechen
```

#### Abbildung 114. Kanalverbindungen auflisten

Die Werte für den Status lauten wie folgt:

INIT	INITIALISIERUNG
BIND	BINDING
START	STARTING
AUSFUEF	RUNNING
STOP	STOPPING oder STOPPED
WIEDERHO	RETRYING
REQST	REQUESTING
DOUBT	STOPPED und INDOUBT (YES)

Weitere Informationen finden Sie unter „Kanalstatus“ auf Seite 186.

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.

#### 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	Value
Aktion	1 (Liste oder Anzeige)
Objekttyp	CLUSCHL
Name	*

Sie werden mit einer Anzeige wie [Abbildung 115 auf Seite 793](#) 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.

```
Clusterwarteschlangenmanager auflisten Kanäle-MQ25 Zeile 1 von 9

Aktionscodes eingeben und die Eingabetaste drücken. F11 drücken, um den Verbindungsstatus anzuzeigen.
1 = Anzeigen 5 = Perform 6 = Start 7 = Stoppen

Kanalname Verbindungsname Status
Typ Clustername ausgesetzt
Name des Clusterwarteschlangenmanagers 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
***** Ende der Liste *****

Befehl == = > -----
F1 = Hilfe F2 = Aufteilen F3 = Verlassen F4 = Filter F5 = Aktualisieren F7 = Bkwd
F8 = Fw F9 = SwapNext F10 = Nachrichten F11 = Status F12 = Abrechnen
```

Abbildung 115. 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).

## **IBM MQ für z/OS für die Verwendung der zEnterprise Data Compression Express-Facility vorbereiten**

Die Funktion zEnterprise Data Compression (zEDC) Express ist ab IBM zEC12 GA2 für bestimmte Modelle von IBM Z -Maschinen mit mindestens z/OS Version z/OS 2.1 verfügbar.

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 zEDC ersetzt. 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/OS an:

- z/OS 2.5: UJ00639
- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638

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<sup>®</sup> -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 795](#)
- [„Definieren einer LU6.2-Verbindung für z/OS mit APPC/MVS“ auf Seite 798](#)

Jede Kanaldefinition muss nur ein Protokoll als Attribut "Übertragungsprotokoll" (Transporttyp) angeben. Ein WS-Manager kann mehr als ein Protokoll für die Kommunikation verwenden.

Es kann auch hilfreich sein, auf [Beispielkonfiguration- IBM MQ for z/OS](#) zu verweisen. Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, lesen Sie [„Kommunikation für IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange einrichten“](#) auf Seite 804.

### Zugehörige Konzepte

[„Verwenden der Anzeigen und der Befehle“](#) auf Seite 778

Sie können die MQSC-Befehle, die PCF-Befehle oder die Operationen und Steuerkonsolen verwenden, um DQM zu verwalten.

[„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“](#) auf Seite 773

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 konfigurieren“](#) auf Seite 702

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

[„Kanäle unter z/OS überwachen und steuern“](#) auf Seite 776

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 799

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 804

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 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### **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
- HFS-Datei, /etc/resolv.conf
- // 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-Systemservices](#).

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

#### **DATABUFFERPOOLSIZ**

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 SYS1.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.

#### **Zugehörige Konzepte**

„[Sendende Beendigung](#)“ auf Seite 797

Am sendenden Ende der TCP/IP-Verbindung gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

„[Empfang auf TCP](#)“ auf Seite 797

Am empfangenden Ende der TCP/IP-Verbindung gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

„[Verwendung der Option TCP-Listener-Backlog](#)“ auf Seite 797

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 (CONNNAME) 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.

Standardmäßig:

- 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 *Verwendung der Option TCP-Listener-Backlog*

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.

## **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:

<i>Tabelle 50. Einstellungen auf dem lokalen z/OS-System für eine ferne Warteschlangenmanagerplattform</i>	
<b>Ferne Plattform</b>	<b>TPNAME</b>
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.
UNIX and Linux-Systeme	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 TPNames in den Kanaldefinitionen eindeutig sind.

Weitere Informationen zu den VTAM-Definitionen, die möglicherweise erforderlich sind, finden Sie im Handbuch *Multiplattform APPC Configuration Guide*.

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 799

Um eine Verbindung zu LU 6.2 herzustellen, gibt es eine Reihe von Einstellungen für die Konfiguration.

„Empfangen auf LU 6.2“ auf Seite 799

Für den Empfang von LU 6.2 gibt es eine Reihe von Einstellungen, die konfiguriert werden müssen.

#### *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 (CONNNAME) 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.

#### *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.

#### **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 804.

Informationen zur Überwachung und Steuerung von Kanälen bei der Verwendung von Gruppen mit gemeinsamer Warteschlange finden Sie unter [„Kanäle unter z/OS überwachen und steuern“](#) auf Seite 776.

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 801.

## 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 801 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 Kommunikationssystem 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

„IBM MQ for z/OS konfigurieren“ auf Seite 702

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

„Kommunikation mit anderen Warteschlangenmanagern konfigurieren“ auf Seite 773

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.

„Cluster und Gruppen mit gemeinsamer Warteschlange“ auf Seite 803

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 803

Bei der Peer-Recovery der gemeinsamen Warteschlange werden Nachrichtenkanalagenten, die Nachrichten in gemeinsam genutzten Warteschlangen verarbeiten, ihren Zugriff auf die Warteschlangen serialisiert.

### Zugehörige Tasks

„Verteilte Warteschlangensteuerung konfigurieren“ auf Seite 151

Dieser Abschnitt enthält ausführlichere Informationen zur übergreifenden Kommunikation zwischen IBM MQ-Installationen, einschließlich Warteschlangendefinition, Kanaldefinition, Auslöserverfahren und Synchronisationspunktprozeduren.

### Zugehörige Informationen

[Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange](#)

[Einreihung in Warteschlange innerhalb von Gruppen](#)

## 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 798.

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.

rungen zugeordnet. Ein Beispiel für die Einrichtung dieser Technologie finden Sie im Abschnitt [Sysplex Distributor verwenden](#)

## **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 dann mit einem Lastausgleich gestartet (siehe „[IBM MQ for z/OS für DQM mit Gruppen mit gemeinsamer Warteschlange vorbereiten](#)“ auf Seite 799), der ausgelöst wird, um den 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)
```

## **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 Gruppendifinition). 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 Gruppendifinition 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.

## Abgehend

Ein abgehender Kanal ist ein gemeinsam genutzter Kanal, wenn er Nachrichten aus einer gemeinsam genutzten Übertragungswarteschlange verschiebt. In den Beispielbefehlen ist der Senderkanal QSG.T0.QM2 ein gemeinsam genutzter Kanal, da die Übertragungswarteschlange QM2 mit QSGDISP (SHARED) definiert ist.

### **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 definieren“](#) auf Seite 775 ). Dies bedeutet, dass der Kanal auf einem anderen Warteschlangenmanager und einer Kanalinitiatorinstanz in der Gruppe mit gemeinsamer Warteschlange erneut gestartet werden kann, wenn das Kommunikationssystem, 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 799.

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.

### **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 255.

### **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 gemein-

sam 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.

### **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.

### **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 795.

## **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](#) .)

### 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 (CONNNAME) 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 Ressourcename 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](#) .

## **IBM MQ mit IMS verwenden**

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die eine Interaktion von IBM MQ mit IMS ermöglichen.

Wenn Sie IBM MQ und IMS für die Zusammenarbeit konfigurieren möchten, müssen Sie die folgenden Tasks ausführen:

- [„IMS-Adapter einrichten“ auf Seite 806](#)
- [„IMS-Bridge konfigurieren“ auf Seite 812](#)

### **Zugehörige Konzepte**

[„IBM MQ mit CICS verwenden“ auf Seite 814](#)

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 816](#)

Verwenden Sie dieses Thema, wenn Sie IMS Open Transaction Manager Access-Exits mit IBM MQ for z/OS verwenden wollen.

### **Zugehörige Tasks**

[„Warteschlangenmanager unter z/OS erstellen“ auf Seite 697](#)

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 814](#)

Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

## Zugehörige Informationen

IBM MQ und IMS

IMS- und IMS-Brückenanwendungen unter IBM MQ for z/OS

### **IMS-Adapter einrichten**

Für die Verwendung von IBM MQ in IMS ist der Adapter IBM MQ - IMS erforderlich (wird im Allgemeinen als IMS-Adapter bezeichnet).

In diesem Abschnitt wird beschrieben, wie der IMS-Adapter in einem IMS-Subsystem zur Verfügung gestellt wird. Wenn Sie mit der Anpassung eines IMS -Subsystems nicht vertraut sind, lesen Sie die Informationen zu *IMS in IBM Documentation*.

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 in „[IBM MQ für IMS definieren](#)“ auf Seite 807.

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 Produktionssysteme ändern Sie `NAME=CSQ1` in Ihren eigenen Subsystemnamen oder verwenden Sie `CSQ1`. Sie können bei Bedarf weitere Subsystemdefinitionen hinzufügen. Weitere Informationen zu LITs finden Sie unter „[Definieren von IBM MQ-Warteschlangenmanagern für den IMS-Adapter](#)“ auf Seite 810.

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 116](#) auf Seite 807 dargestellt.
9. Wenn Sie den IMS-Auslösemonitor verwenden möchten, definieren Sie die IMS-Auslösemonitoranwendung CSQQTRMN und führen Sie PSBGEN und ACBGEN aus. Weitere Informationen finden Sie in [„IMS-Auslösemonitor konfigurieren“](#) auf Seite 812.
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.load1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN DD *
INCLUDE ACSQMOD(CSQQSTUB)
INCLUDE IMSLIB(DFSLI000)
ALIAS MQCONN,MQCONN,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 CSQQDYS(R)
/*
```

<sup>1</sup>Specify the name of a library accessible to IMS applications that want to make dynamic calls to IBM MQ.

*Abbildung 116. Beispiel-JCL zum Verlinken-Bearbeiten des Stubs für dynamische Verbindungen*

## Zugehörige Konzepte

[„IMS-Bridge konfigurieren“](#) auf Seite 812

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.

## Zugehörige Informationen

[IBM MQ und IMS](#)

[IMS- und IMS-Brückenanwendungen unter IBM MQ für 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, verketteten 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-Beendigungs- 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-Beendigungs- 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:

<b>CSQ1</b>	Der Standard subsystemname, der im Lieferumfang von IBM MQ enthalten ist. Sie können diese Änderung an Ihre Installation anpassen.
<b>MQM1</b>	Der Standardwert LIT wird in CSQQDEFV angegeben.
<b>CSQQESMT</b>	Der Name des externen Subsystemmoduls. Sie müssen diesen Wert verwenden.
<b>R</b>	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:

<b>SYS3</b>	Der Subsystemname
<b>MQM3</b>	Die LIT wie in CSQQDEFV angegeben
<b>CSQQESMT</b>	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 vierstelligen 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-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 51 auf Seite 810](#) zeigt die verschiedenen Möglichkeiten von SSM-Spezifikationen.

Tabelle 51. 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.
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.

### Definieren von IBM MQ-Warteschlangenmanagern für den IMS-Adapter

Die Namen der WS-Manager von IBM MQ und die zugehörigen Sprachschnittstellentoken (LITs) müssen in der Definitionstabelle des Warteschlangenmanagers definiert sein.

Verwenden Sie das bereitgestellte Makro CSQQDEFX, um das CSQQDEFV-Lademodul zu erstellen. [Abbildung 117 auf Seite 811](#) zeigt die Syntax dieses Assemblermakros an.

```
CSQQDEFX TYPE=ENTRY|DEFAULT,NAME=qmgr-name,LIT=token  
or  
CSQQDEFX TYPE=END
```

Abbildung 117. 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 er in **MQCONN** oder **MQCONNX** angegeben ist.

### LIT = Token

Gibt den Namen des Sprachschnittstellentokens (LIT) an, das IMS für die Identifikation des Warteschlangenmanagers verwendet.

Ein MQCONN -oder MQCONNX -Aufruf verknüpft den Eingabeparameter *name* und den Ausgabe-parameter *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 118 auf Seite 812](#) 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 118. Layout einer Warteschlangenmanager-Definitionstabelle

## 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 119. 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 120. Beispiel-PSB-Definition für CSQQTRMN

Weitere Informationen zum Starten und Stoppen des IMS-Auslösemonitors finden Sie unter [Auslösemonitor für IMS steuern](#).

## 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-Brücke anzupassen.

### Definieren Sie die XCF- und OTMA-Parameter für IBM MQ.

In diesem Schritt werden die XCF-Gruppen- und -Memberebenen 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 -Membnernamen 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-Membnernamen des IMS-Systems an.

Dies ist mit dem in der Speicherklassendefinition angegebenen Membnernamen identisch (siehe nächsten Schritt).

### **Informieren Sie IBM MQ über die XCF-Gruppe und den Membnernamen 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 Membnernamen 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 legt die Sicherheitsstufe fest, die auf **jeden** IBM MQ -Warteschlangenmanager angewendet werden soll, der über OTMA eine Verbindung zu IMS herstellt. Weitere Informationen finden Sie unter [Sicherheitsaspekte bei der Verwendung von IBM MQ mit IMS](#).

### **Zusätzliche IMS-Verbindung zum gleichen Warteschlangenmanager hinzufügen**

Gehen Sie zum Hinzufügen einer IMS-Verbindung zum gleichen Warteschlangenmanager folgendermaßen vor:

- Definieren Sie eine zweite STGCLASS-Speicherklasse, die auf das neue IMS verweist; weitere Informationen finden Sie unter [DEFINE STGCLASS](#).
- Fügen Sie eine neue lokale Warteschlange hinzu, die auf die zweite Speicherklasse verweist.

#### **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**

[„IMS-Adapter einrichten“](#) auf Seite 806

Für die Verwendung von IBM MQ in IMS ist der Adapter IBM MQ - IMS erforderlich (wird im Allgemeinen als IMS-Adapter bezeichnet).

#### **Zugehörige Informationen**

[IBM MQ und IMS](#)

## 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 zur Konfiguration des IBM MQ CICS-Adapters und der Komponenten der IBM MQ CICS bridge finden Sie im Abschnitt [Verbindungen zu MQ konfigurieren](#) in der CICS-Dokumentation.

### Zugehörige Konzepte

„IBM MQ mit IMS verwenden“ auf Seite 805

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die eine Interaktion von IBM MQ mit IMS ermöglichen.

### Zugehörige Verweise

„Upgrade und Serviceaktualisierungen für Language Environment oder z/OS Callable Services durchführen“ auf Seite 814

Die Aktionen, die Sie ausführen müssen, variieren je nach der Verwendung von CALLLIBS oder LINK und Ihrer Version von SMP/E.

### Zugehörige Informationen

[IBM MQ und CICS](#)

## 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 Serviceaktualisierungsdurchführen:

- Language Environment
- z/OS Callable Services (z. B. APPC und RRS)

Tabelle 52. 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	Aktion bei Verwendung von LINK
	<p><b>Anmerkung: Es ist nicht erforderlich, separate Jobs für Language Environment und Callable Services auszuführen. Ein Job reicht aus.</b></p>	
Language Environment	<ol style="list-style-type: none"> <li>1. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.</li> <li>2. Geben Sie auf der SMP_CNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie in <i>SMP/E for z/OS: Commands</i>.</li> <li>3. Führen Sie den SMP/E-Job aus.</li> </ol>	Es ist keine Aktion erforderlich, vorausgesetzt, die SMP/E-Zonen wurden für die automatische Wiederverlinkung konfiguriert, und der Job CSQ8SLDQ wurde ausgeführt.

Tabelle 52. Der Service wurde angewendet, oder das Produkt wurde auf ein neues Release aktualisiert. (Forts.)

Produkt	<b>Aktion bei Verwendung von CALLLIBS und SMP/E V3r2 oder höher</b>  <b>Anmerkung: Es ist nicht erforderlich, separate Jobs für Language Environment und Callable Services auszuführen. Ein Job reicht aus.</b>	Aktion bei Verwendung von LINK
Aufrufbare Services	<ol style="list-style-type: none"> <li>1. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.</li> <li>2. Geben Sie auf der SMP/CNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie in <i>SMP/E for z/OS: Commands</i>.</li> <li>3. Führen Sie den SMP/E-Job aus.</li> </ol>	Es ist keine Aktion erforderlich, vorausgesetzt, die SMP/E-Zonen wurden für die automatische Wiederverlinkung konfiguriert, und der Job CSQ8SLDQ wurde ausgeführt.

Tabelle 53. Eines der Produkte wurde in einer neuen SMP/E-Umgebung und Bibliotheken auf ein neues Release aktualisiert.

Produkt	<b>Aktion bei Verwendung von CALLLIBS und SMP/E V3r2 oder höher</b>  <b>Anmerkung: Sie müssen keine drei separaten Jobs für Language Environment und Callable Services ausführen. Für beide Produkte reicht ein Job aus.</b>	Aktion bei Verwendung von LINK
Language Environment	<ol style="list-style-type: none"> <li>1. Ändern Sie die DDDEFs für SCEELKED und SCEESPC so, dass sie auf die neue Bibliothek zeigen.</li> <li>2. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.</li> <li>3. Geben Sie auf der SMP/CNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie in <i>SMP/E for z/OS: Commands</i>.</li> <li>4. Führen Sie den SMP/E-Job aus.</li> </ol>	<ol style="list-style-type: none"> <li>1. Löschen Sie die XZMOD-Untereinträge für die folgenden LMOD-Einträge in der IBM MQ for z/OS-Zielzone:                       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</li> <li>2. Richten Sie die entsprechenden ZONEINDEXs zwischen den IBM MQ-Zonen und den Language Environment-Zonen ein.</li> <li>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.</li> <li>4. Führen Sie CSQ8SLDQ aus.</li> </ol>

Tabelle 53. Eines der Produkte wurde in einer neuen SMP/E-Umgebung und Bibliotheken auf ein neues Release aktualisiert. (Forts.)

Produkt	Aktion bei Verwendung von CALLLIBS und SMP/E V3r2 oder höher  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
Aufrufbare Services	<ol style="list-style-type: none"> <li>1. DDDEF für CSSLIB so ändern, dass sie auf die neue Bibliothek verweist</li> <li>2. Legen Sie die Grenze für Ihren SMP/E-Job auf die Zielzone fest.</li> <li>3. Geben Sie auf der SMP_CNTL-Karte LINK LMODS CALLLIBS an. Sie können auch andere Parameter angeben, wie z. B. CHECK, RETRY (YES) und RC. Weitere Informationen finden Sie in <i>SMP/E for z/OS: Commands</i>.</li> <li>4. Führen Sie den SMP/E-Job aus.</li> </ol>	<ol style="list-style-type: none"> <li>1. Löschen Sie die XZMOD-Untereinträge für die folgenden LMOD-Einträge in der IBM MQ for z/OS-Zielzone:  CMQXRCTL, CMQXSUPR, CSQBSRV, CSQILPLM, CSQXJST, CSQXRCTL, CSQXSUPR, CSQ3AMGP, CSQ3EPX, CSQ3REPL</li> <li>2. Richten Sie die entsprechenden ZONEINDEXs zwischen den IBM MQ-Zonen und den Callable Services-Zonen ein.</li> <li>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.</li> <li>4. Führen Sie CSQ8SLDQ aus.</li> </ol>

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 816.

### 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 121. Beispiel für einen SMP/E-LINK-CALLLIBS-Job

### 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 Zielaufhebungsbenutzer (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 Zielaufhebung

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. Ein Beispiel für dieses Modul wird von IMS bereitgestellt. Informationen hierzu finden Sie im Handbuch *IMS/ESA Customization Guide*.

### 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 817](#)
- [„Benutzerexit für Zielaufhebung“ auf Seite 819](#)

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 mit IMS verwenden“ auf Seite 805](#)

Der IBM MQ-IMS-Adapter und die IBM MQ-IMS-Bridge sind die beiden Komponenten, die eine Interaktion von IBM MQ mit IMS ermöglichen.

### Zugehörige Informationen

[IBM MQ und IMS](#)

[IMS- und IMS-Brückenanwendungen unter IBM MQ for z/OS](#)

## Exit-Exit DFSYPRX0 vor dem Routing

Dieser Abschnitt enthält einen Beispiexit für die Vorweiterleitung für OTMA in IMS.

Sie müssen zuerst einen Vor-Routing-Exit DFSYPRX0 codieren. Die Parameter, die von IMS an diese Routine übergeben werden, sind im Handbuch *IMS/ESA Customization Guide* dokumentiert.

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. Im Handbuch *IMS/ESA Customization Guide* erfahren Sie, wo Sie diesen Wert festlegen können. Es wird empfohlen, Ihren Clientnamen (im Parameter OTMACON des Makros CSQ6SYSP) auf den Namen des Warteschlangenmanagers zu setzen. Dies ist der Standardwert. 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 122 auf Seite 819](#) angezeigt:

```

TITLE 'DFSYPX0: OTMA PRE-ROUTING USER EXIT'
DFSYPX0 CSECT
DFSYPX0 AMODE 31
DFSYPX0 RMODE ANY
*
SAVE (14,12),,DFSYPX0&SYSDATE&SYSTIME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DFSYPX0,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 122. Assembler-Beispiel für OTMA-Exit vor Routing

## Benutzerexit für Zielauflösung

Dieser Abschnitt enthält ein Beispiel für einen Zielauflösungs-Benutzerexit für IMS.

Wenn Sie die Register 15 bis 4 in DFSYPX0 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 DRUOVCPPE.

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.

Wir weisen darauf hin, dass der DRU-Exit so einfach wie möglich ist. Daher werden in diesem Beispiel alle Nachrichten, die von IMS für einen bestimmten IBM MQ-Warteschlangenmanager stammen, in dieselbe IBM MQ-Warteschlange eingereicht.

Wenn die Nachricht persistent sein muss, muss IMS eine synchronisierte Transaktionspipe verwenden. Dazu muss der DRU-Exit die OUTPUT-Markierung setzen. Weitere Informationen hierzu finden Sie im Handbuch *IMS/ESA Customization Guide*.

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 123 auf Seite 821](#) angezeigt.

```

TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTEME
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 123. Beispiel-Assembler-DRU-Exit

## z/OS V 9.0.1 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/OSMF verstehen, 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-Ressourcenmanagement und Softwareservices
- Configuration-Configuration Assistant und Performance-Workload Management, und
- Leistung-Workload-Management

Details dazu finden Sie zusammen mit dem Abschnitt *Einführung in das Lernprogramm-Cloud* im Abschnitt *Neuerungen in diesem Release*.

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
  - Vorlagen
- instances
- Workflows

und so weiter. Weitere Informationen finden Sie im Abschnitt *Cloud Provisioning* im *z/OSMF Programming Guide* oder in der Hilfe zu z/OSMF WUI.

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 und das Verzeichnis und die Dateistruktur werden im IBM MQ for z/OS-Programmverzeichnis beschrieben, das Sie vom [IBM Publications Center](#) herunterladen können.

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“ auf Seite 823](#)
- [„Sicherheitseinstellungen“ auf Seite 824](#)
- [„Beschränkungen“ auf Seite 827](#)

Beispielworkflowsanwendungen werden bereitgestellt für:

- [„Automatisieren Sie die Bereitstellung oder Löschung von IBM MQ-Warteschlangenmanagern und führen Sie Aktionen für die bereitgestellten Warteschlangenmanager aus“ auf Seite 828](#)

- „Automatisieren Sie die Bereitstellung oder Löschung von lokalen IBM MQ-Warteschlangen und führen Sie Aktionen für die bereitgestellten Warteschlangen aus“ auf Seite 829.

### Zugehörige Konzepte

„IBM MQ for z/OS konfigurieren“ auf Seite 702

Verwenden Sie dieses Thema als schrittweise Anleitung für die Anpassung Ihres IBM MQ for z/OS-Systems.

## Voraussetzungen

Die Voraussetzungen, die Sie für die Ausführung von IBM z/OS Management Facility (z/OSMF) mit IBM MQ benötigen

Die IBM MQ -Workflows, die im Lieferumfang von IBM MQ 9.0.1 enthalten sind, nutzen neue Funktionen in z/OSMF, die über APARs in z/OS 2.1 und 2.2 bereitgestellt werden. 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 UNIX System Services for z/OS (USS) installiert.
8. Das USS-Verzeichnis **'/tmp'** ist verfügbar, da der 'provision.xml' -Workflow möglicherweise eine temporäre Datei in diesem Verzeichnis erstellen kann. 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 USS-Befehl 'SLEEP' als Systemaufruf auf.

Abhängig von Ihrer USS-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 werden kann, können Sie versuchen, die folgenden Zeilen in den Execs `CSQ4ZWS1.rexx` und `CSQ4ZWS2.rexx` zu ersetzen:

```
CALL SYSCALLS('ON')           /* Enable USS calls */
ADDRESS SYSCALL
"SLEEP" 10                    /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF'          /* Disable USS 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 821

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.

## Sicherheitseinstellungen

Die Sicherheitseinstellungen, die für die Ausführung von z/OSMF erforderlich sind.

Die folgenden Eigenschaften für die Benutzer-ID-Variablen sind in der Eigenschaftendatei definiert. Weitere Informationen finden Sie im Abschnitt [„Workflows ausführen“](#) auf Seite 832.

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 <b>CSQ_ADMIN_*</b> -Benutzer-IDs, die im folgenden Text aufgeführt sind, mit unterschiedlichen Benutzer-IDs ausgeführt werden. Die Benutzer-ID, die verwendet wird, wird durch die Eigenschaft <b>runAsUser</b> auf dem jeweiligen Schritt in den Workflows angegeben.

Benutzer-ID, Eigenschaft	Beschreibung
CSQ_ADMIN_APF_USERID	Die Benutzer-ID, die verwendet werden soll, wenn APF die Ladebibliothek autorisiert, die das Systemparametermodul des Warteschlangenmanagers enthält.
CSQ_APF_APPROVAL_ID	Die Genehmigungs-ID, die es Benutzern ermöglicht, den APF-Berechtigungs-schritt 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  <b>Achtung:</b> Diese Benutzer-ID muss UPDATE-Zugriff auf das Profil der gestarteten Task (MVS.START.STC. *) haben. in der Klasse "OPERCMD5". Weitere Informationen finden Sie im Abschnitt <i>z/OS Operator Console Operations</i> in den Informationen zu IBM z/OS in IBM Documentation.
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-Befehls-schritte 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-Befehls-schritt 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(OPERCMD5) 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(OPERCMD5) 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 Bereitstellungwarteschlangen 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 Workflow des WS-Managers provision.xml inaktiviert derzeit die Sicherheit des Subsystems. 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 Schabloneninstanz ordnet der `provision.xml`-Workflow einen Port in dem Bereich zu und startet einen Listener, der 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“ auf Seite 823

Die Voraussetzungen, die Sie für die Ausführung von IBM z/OS Management Facility (z/OSMF) mit IBM MQ benötigen

## Beschränkungen

Einschränkungen bei der Verwendung von z/OSMF mit IBM MQ.

1. Der `provision.xml`-Workflow automatisiert derzeit die folgenden hervorgehobenen Anpassungsaufgaben des Warteschlangenmanagers:

Task	Beschreibung
1	Angeben der z/OS-Systemparameter
2	APF-Autorisierung der IBM MQ-Ladebibliotheken ( <b>provision.xml führt APF-Berechtigung für einige Bibliotheken durch</b> )
3	Aktualisieren der z/OS-Linkliste und LPA
4	Aktualisieren der Tabelle mit den z/OS-Programmeigenschaften
5	<b>Definieren des IBM MQ-Subsystems für z/OS</b>
6	<b>Erstellen von Prozeduren für den IBM MQ-Warteschlangenmanager</b>
7	<b>Prozeduren für den Kanalinitiator erstellen</b>
8	<b>Definieren des IBM MQ-Subsystems für eine z/OS-WLM-Serviceklasse</b>
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	<b>Eingabedatengruppen für die Initialisierung anpassen</b>

Task	Beschreibung
14	<b>Bootstrap- und Protokoll datengruppen erstellen</b>
15	<b>Seitengruppen definieren</b>
16	Hinzufügen der IBM MQ-Einträge zur Db2-Gruppe mit gemeinsamer Datennutzung
17.	<b>Anpassen der Systemparametermodule (einige)</b>
18	<b>Kanalinitiatorparameter anpassen (einige)</b>
19.	Batch-, TSO- und RRS-Adapter konfigurieren
20	Konfigurieren Sie die Operationen und Steuerkonsolen.
21	Teildatei für die Speicherausgangsformatierung 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 Members 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 Informationen finden Sie im Abschnitt „Workflows ausführen“ auf Seite 832.

### Zugehörige Konzepte

„Sicherheitseinstellungen“ auf Seite 824

Die Sicherheitseinstellungen, die für die Ausführung von z/OSMF erforderlich sind.

## **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 Dieser Beispielworkflow:

Workflowname	Beschreibung
	<ul style="list-style-type: none"> <li>• Stellt die erforderlichen Systemressourcen für einen Warteschlangenmanager bereit.</li> <li>• Stellt die erforderlichen Systemressourcen für einen Kanalinitiator bereit.</li> <li>• Startet den Warteschlangenmanager (der auch den Kanalinitiator und den TCP/IP-Listener startet)</li> <li>• Führt das Prüfprogramm für die Installation des Beispielwarteschlangenmanagers aus.</li> </ul> <p>Eine Umgebungseigenschaft kann festgelegt werden, um die Bereitstellung von Warteschlangenmanagern mit unterschiedlichen Merkmalen zu steuern. Weitere Informationen finden Sie unter „<a href="#">Workflows ausführen</a>“ auf <a href="#">Seite 832</a>.</p> <p><b>Anmerkung:</b> Es wird eine Manifestdatei (<code>provision.mf</code>) bereitgestellt, die Sie beim Hinzufügen einer Vorlage für diesen Workflow unterstützt. Diese Datei enthält einen Verweis auf die Datei <code>qaas_readme.pdf</code>, die zusätzliche Informationen enthält. Sie können auf die Datei über einen Link zugreifen, sobald die Vorlage hinzugefügt wurde.</p>
deprovision.xml	<p>Löschung eines IBM MQ for z/OS-Warteschlangenmanagers</p> <p>Dieser Beispielworkflow:</p> <ul style="list-style-type: none"> <li>• Stoppt den Kanalinitiator (der auch den TCP/IP-Listener stoppt) und den Warteschlangenmanager.</li> <li>• Wartet auf die zu stoppenden Subsysteme.</li> <li>• Löscht alle Kanalinitiator- und Warteschlangenmanager-Systemressourcen.</li> </ul>
startQMgr.xml	<p>Start eines IBM MQ for z/OS-Warteschlangenmanagers</p> <p>Dieser Beispielworkflow startet den Warteschlangenmanager (der auch den Kanalinitiator und den TCP/IP-Listener startet).</p>
stopQMgr.xml	<p>Stopp eines IBM MQ for z/OS-Warteschlangenmanagers</p> <p>Dieser Beispielworkflow stoppt den Kanalinitiator (der auch den TCP/IP-Listener stoppt) und den Warteschlangenmanager.</p>

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 Informationen finden Sie im Abschnitt „[Workflowvariablen-XML-Dateien](#)“ auf [Seite 832](#).

**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	<p>Definieren Sie eine lokale Warteschlange.</p> <p>Dieser Beispielworkflow veranschaulicht, wie z/OSMF-Workflows verwendet werden können, um kleine, mittlere oder große Warteschlangen auf der Basis von Eigenschafteneinstellungen zu definieren.</p> <p><b>Anmerkung:</b> Es wird eine Manifestdatei (<code>provision.mf</code>) bereitgestellt, die Sie beim Hinzufügen einer Vorlage für diesen Workflow unterstützt. Diese Datei enthält einen Verweis auf die Datei <b>qaas_readme.pdf</b>, die zusätzliche Informationen enthält. Sie können auf die Datei über einen Link zugreifen, sobald die Vorlage hinzugefügt wurde.</p>
displayQueue.xml	<p>Ausgewählte Attribute einer lokalen Warteschlange anzeigen</p> <p>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.</p> <p>Weitere Informationen finden Sie in den <i>REST-APIs für die Cloud-Bereitstellung</i>, die im z/OSMF-Programmierhandbuch dokumentiert sind, und im Abschnitt <a href="#">z/OSMF-Workflow-Services</a>.</p>
deleteQueue.xml	<p>Lokale Warteschlange löschen</p> <p>Dieser Beispielworkflow löscht eine lokale Warteschlange in einem angegebenen Warteschlangenmanager.</p>
putQueue.xml	<p>Fügen Sie eine oder mehrere Nachrichten in eine lokale Warteschlange ein.</p> <p>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.</p>
getQueue.xml	<p>Ruft eine oder mehrere Nachrichten aus einer lokalen Warteschlange ab.</p> <p>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.</p> <p>Weitere Informationen finden Sie im Artikel <a href="#">REST-APIs für Cloudbereitstellung</a>. Weitere Informationen finden Sie unter <a href="#">z/OSMF-Workflowservices</a>.</p>
loadQueue.xml	<p>Laden von Nachrichten aus einem Datensatz in eine lokale Warteschlange.</p> <p>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 Informationen finden Sie im Abschnitt <a href="#">„Workflows ausführen“</a> auf Seite 832.</p>
offloadQueue.xml	<p>Versetzen von Nachrichten aus einer lokalen Warteschlange in eine Datei.</p> <p>Bei diesem Beispielworkflow werden Nachrichten aus einer lokalen Warteschlange in einen Datensatz geladen. Der Standardname der Datei wird durch Festlegen einer Eigenschaft angegeben. Weitere Informationen finden Sie im Abschnitt <a href="#">„Workflows ausführen“</a> auf Seite 832.</p>

Workflowname	Beschreibung
clearQueue.xml	Nachrichten in einer lokalen Warteschlange löschen.  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 Dienstprogramm IBM MQ for z/OS QLOAD auf, das im Wesentlichen das Dienstprogramm **dmpmqmsg** ist, das mit IBM MQ for Multiplatforms verfügbar ist. Daher wird erwartet, dass Nachrichten, die aus einer Datei in eine Warteschlange oder aus einer Warteschlange in eine Datei geladen wurden, das Format **dmpmqmsg** aufweisen.

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 832 deklariert sind.

### Zugehörige Konzepte

„Beschränkungen“ auf Seite 827

Einschränkungen bei der Verwendung von z/OSMF mit IBM MQ.

## 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

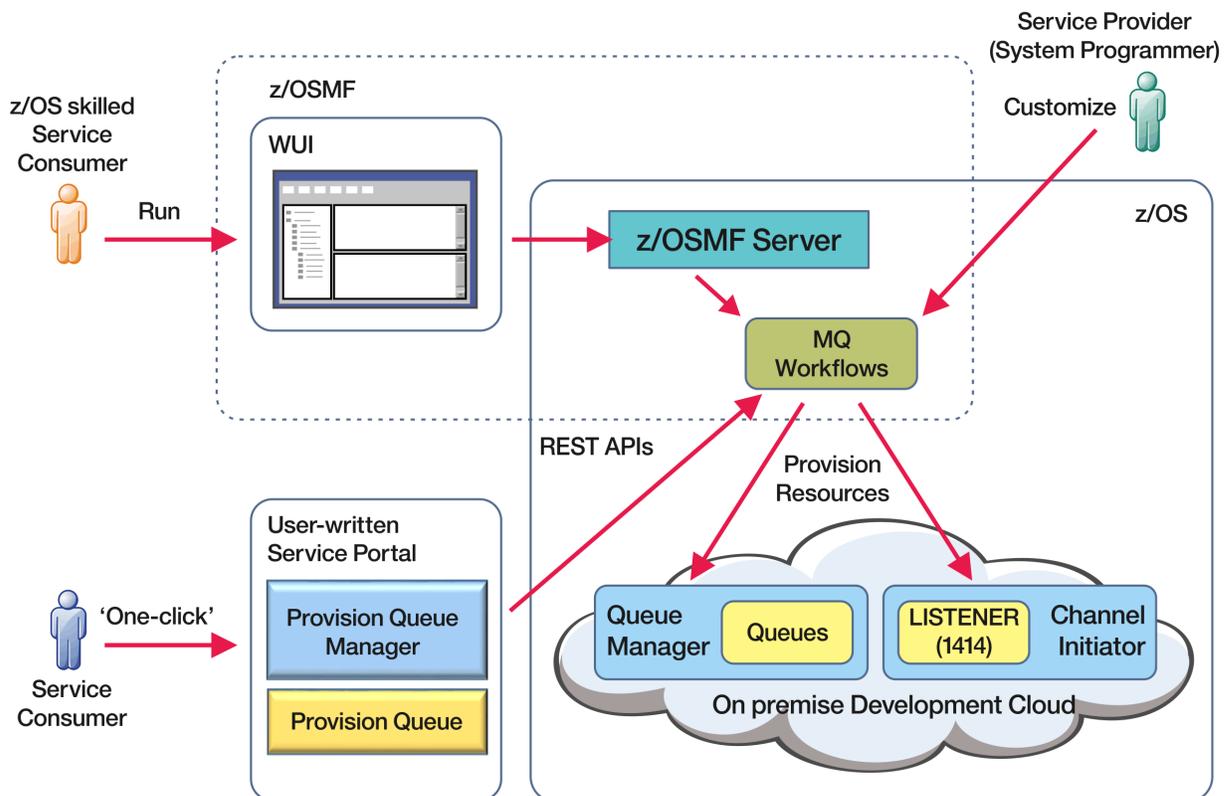
In den folgenden Dateien werden Variablen deklariert, die von den z/OSMF-Beispielworkflows und den zugehörigen JCL-oder REXX-Exec-Dateien referenziert werden:

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 124. Bereitstellung von IBM MQ for z/OS-Ressourcen per Mausclick



Bevor die Workflows ausgeführt werden können, müssen einige Eigenschaften in der folgenden Datei festgelegt werden:

Name der Eigenschaftendatei der Workflowvariablen	Beschreibung
workflow_variables.properties	<p>Anfängliche Eigenschaften für die Workflowvariablen. Kommentare in der Datei geben den Zweck der einzelnen Eigenschaften an.</p> <ul style="list-style-type: none"> <li>Eigenschaften in Meta-Klammern (&lt; &gt;) müssen auf benutzerspezifische Werte gesetzt werden.</li> <li>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.</li> </ul> <p>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 Pagesets für die einzelnen Umgebungstyp ändern.</p> <ul style="list-style-type: none"> <li>Andere Eigenschaften sind auf IBM MQ-Standardwerte gesetzt, können aber bei Bedarf geändert werden, um die lokalen Konventionen zu erfüllen.</li> </ul>

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.

Für weitere Details siehe *Cloud-Bereitstellung* in der z/OSMF WUI-Hilfe sowie [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 828

Es werden Muster bereitgestellt, um die Bereitstellung von Warteschlangenmanagern und lokalen Warteschlangen zu automatisieren.

## V 9.0.3 z/OS MQ Adv. VUE IBM MQ Advanced for z/OS VUE konfigurieren

Verwenden Sie diese Informationen für die Konfiguration der Funktionen, die mit der IBM MQ Advanced for z/OS VUE-Berechtigung verfügbar sind.

### Informationen zu diesem Vorgang

Ab IBM MQ 9.0.3 können Sie die Funktionen von IBM MQ Advanced for z/OS VUE Connector Pack verwenden, um die MFT -Topologie unter z/OS zu vereinfachen und die Konnektivität von IBM MQ Advanced for z/OS, Value Unit Edition -Warteschlangenmanagern zum IBM Blockchain -Service in IBM Cloud (formerly Bluemix) zu nutzen.

**V 9.0.4** Ab IBM MQ 9.0.4 können Sie eine IBM MQ classes for JMS- oder IBM MQ classes for Java-Anwendung über eine Clientverbindung mit einem Warteschlangenmanager unter z/OS, der über das Attribut **ADVCAP**(ENABLED) verfügt, verbinden.

### Vorgehensweise

1. Aktivieren Sie Fernverbindungen des Managed File Transfer-Agenten mit IBM MQ Advanced for z/OS, Value Unit Edition.
2. Konfigurieren Sie IBM MQ Advanced for z/OS VUE für die Verwendung mit dem IBM Blockchain-Service in IBM Cloud.

## V 9.0.3 z/OS MQ Adv. VUE IBM MQ Advanced for z/OS VUE für die Verwendung mit dem IBM Cloud Product Insights-Service in IBM Cloud (formerly Bluemix) konfigurieren

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im [Blogbeitrag Service Deprecation: IBM Cloud Product Insights](#).

## **V 9.0.3** **z/OS** **MQ.Adv.VUE** **IBM Cloud Product Insights-Serviceinstanz in IBM Cloud (formerly Bluemix) erstellen**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im Blogbeitrag [Service Deprecation: IBM Cloud Product Insights](#).

## **V 9.0.3** **z/OS** **MQ.Adv.VUE** **z/OS-Warteschlangenmanager für die Verwendung mit der IBM Cloud Product Insights-Serviceinstanz in IBM Cloud (formerly Bluemix) konfigurieren**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im Blogbeitrag [Service Deprecation: IBM Cloud Product Insights](#).

## **z/OS** **V 9.0.4** **MQ.Adv.VUE** **Verbindung zu IBM Cloud Product Insights in IBM Cloud über einen HTTP-Proxy herstellen**

Der IBM Cloud Product Insights-Service ist nicht mehr verfügbar. Weitere Informationen finden Sie im Blogbeitrag [Service Deprecation: IBM Cloud Product Insights](#).

## **V 9.0.3** **z/OS** **MQ.Adv.VUE** **MFT-Agentenkonnektivität zu fernen z/OS-Warteschlangenmanagern**

Managed File Transfer-Agenten unter z/OS, die unter der Produkt-ID (PID) von IBM MQ Advanced for z/OS VUE ausgeführt werden, können eine Verbindung zu einem fernen Warteschlangenmanager unter z/OS herstellen, indem Sie eine Clientverbindung verwenden.

Weitere Informationen finden Sie im Abschnitt [Nur ferne Verbindungen des Agenten mit IBM MQ Advanced for z/OS, Value Unit Edition aktivieren](#).

## **V 9.0.3** **z/OS** **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 den Status einer Ressource in Ihrer Blockchain asynchron zu verbinden, zu suchen und zu aktualisieren, indem Sie eine Messaging-Anwendung verwenden, die eine Verbindung zu Ihrem IBM MQ Advanced-Warteschlangenmanager herstellt.

### **Vorbereitende Schritte**

- IBM MQ Bridge to blockchain ist als Teil eines Connector-Packs in IBM MQ Advanced for z/OS Value Unit Edition 9.0.4 verfügbar. Sie können eine Verbindung zu z/OS-Warteschlangenmanagern herstellen, die auf derselben Befehlsebene ausgeführt werden.
- IBM MQ Bridge to blockchain wird für die Verwendung mit Ihrem Blockchain-Netz unterstützt, das auf Hyperledger Fabric 1.0 architecture basiert.
- IBM MQ Bridge to blockchain muss in einer x86 Linux-Umgebung installiert, konfiguriert und ausgeführt werden, in der Folgendes installiert ist:
  - IBM MQ 9.0.3 Redistributable Java -Client.
  - IBM Java runtime environment Version 8.

Wenn der IBM MQ 9.0.4 Redistributable Java -Client und IBM Java runtime environment Version 8 bereits installiert sind, müssen Sie die Schritte „4“ auf Seite 837 und „5“ auf Seite 837 nicht ausführen.

## 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 (formerly Bluemix) 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 Projectgemeinschaft entwickelt wird, einschließlich IBM als ursprünglicher Codeentwickler. Hyperledger Project oder Hyperledger ist 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 bereits IBM MQ Advanced for z/OS VUE und IBM Blockchain verwenden, können Sie über die IBM MQ Bridge to blockchain einfache Anfragen und Aktualisierungen senden und Antworten aus Ihrem Blockchain-Netz empfangen. Auf diese Weise können Sie Ihre lokale IBM Software mit dem Cloud-Blockchain-Service 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. 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.

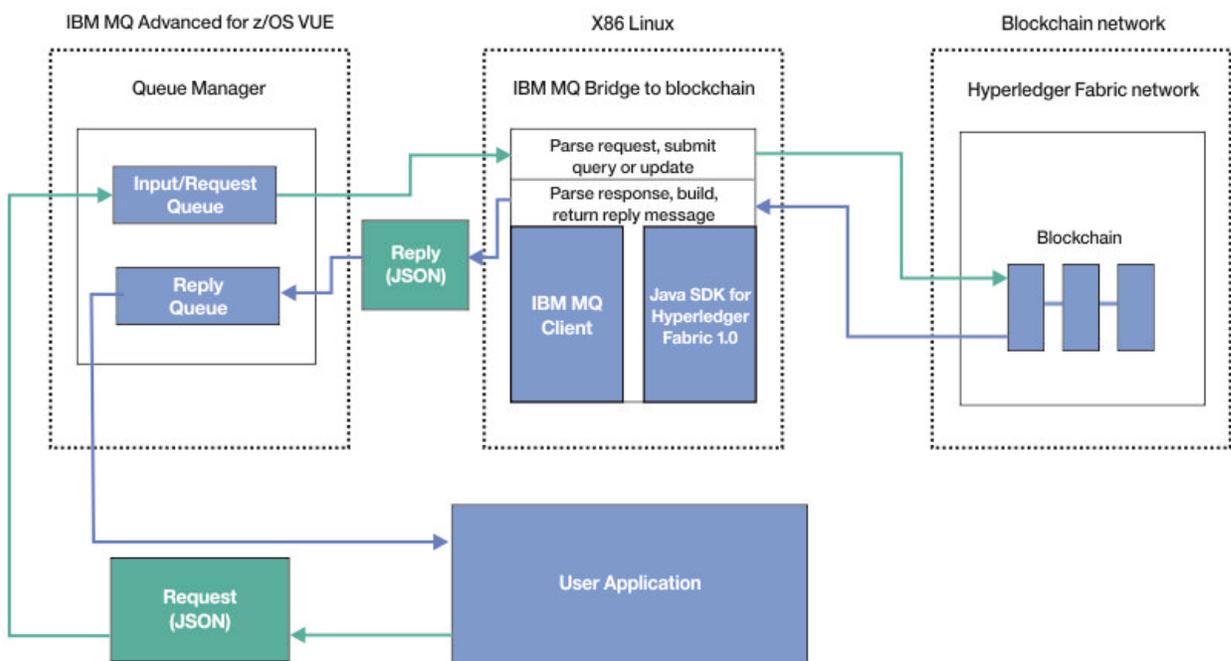


Abbildung 125. IBM MQ Bridge to blockchain

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 Sie die Objekte für die Bridge, die in `csq4bcbq.jc1` definiert sind.  
Musterbrückenwarteschlangendefinitionen werden für die standardmäßigen benannten Warteschlangen bereitgestellt, die für Benutzerberechtigungen und für die Nachrichteneingabe für die Bridge verwendet werden: `SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE` und `SYSTEM.BLOCKCHAIN.INPUT.QUEUE`.
  - a) Kopieren Sie `csq4bcbq.jc1` in eine z/OS-Datei.
  - b) Bearbeiten Sie `csq4bcbq.jc1`, um Ihren z/OS -Warteschlangenmanager anzupassen. Sie müssen einen Warteschlangenmanagernamen und das übergeordnete Qualifikationsmerkmal für die IBM MQ-Produktbibliotheken angeben. Sie können die Beispiele für die **APPL1**-Brückenwarteschlange ändern oder weitere INPUT- und REPLY-Warteschlangen für weitere Anwendungen hinzufügen.
  - c) Übergeben Sie den `csq4bcbq.jc1`, um die Objekte zu erstellen, die Sie definiert haben.
2. Übertragen Sie die Datei `x86download.tar.gz` unter Verwendung Ihrer bevorzugten Methode aus dem Verzeichnis `x86download` in Ihre x86 Linux-Umgebung.  
Stellen Sie sicher, dass die Datei im Binärmodus übertragen wird.
3. Entpacken Sie die Datei `x86download.tar.gz` in x86 Linux.

```
tar -xvzf x86download.tar.gz
```

Die vier Verzeichnisse, die entpackt werden, sind `bin`, `lib`, `prereqs` und `samp`.

4. Laden Sie IBM Java runtime environment Version 8 in Ihre x86 Linux -Umgebung herunter.
  - a) Klicken Sie auf das **Installierbare Paket (InstallAnywhere as root)** auf der Seite [IBM Java SDK Developer Centre Java 8 Downloads](#) mit dem Dateinamen `ibm-java-x86_64-jre-8.0-4.6.bin` vom Abschnitt **Linux on AMD64/EMT64T**.  
Die Lizenzseite für IBM SDK, Java Technology Edition, Version 8 wird angezeigt.
  - b) Vereinbaren Sie die Lizenz, um fortzufahren.  
Wählen Sie im Download-Fenster die Option **Datei speichern** aus, um den Download zu starten.
  - c) Führen Sie die Datei `ibm-java-x86_64-jre-8.0-4.6.bin` aus, um sie in Ihrer x86 Linux-Umgebung zu installieren.  
Die Standardinstallationsposition ist das Verzeichnis `/opt/ibm/`.

```
./ibm-java-x86_64-jre-8.0-4.6.bin
```

- d) Legen Sie den Pfad zu Ihrer IBM 8-JRE fest:

```
export PATH=/opt/ibm/java-x86_64-80/jre/bin:$PATH
```

5. Laden Sie den IBM MQ 9.0.4 Redistributable Java -Client von [Fix Central](#) herunter.
  - a) Klicken Sie auf den Link `9.0.4.0-IBM-MQC-Redist LinuxX64`.
  - b) Wählen Sie **Download mit Ihrem Browser (HTTPS) aus**.  
Klicken Sie auf Fortfahren
  - c) Vereinbaren Sie die Bedingungen der Lizenz.
  - d) Klicken Sie auf den Link `9.0.4.0-IBM-MQC-Redist-LinuxX64.tar.gz` und wählen Sie **Datei speichern** aus, um sie herunterzuladen.
  - e) Entpacken Sie die Datei `9.0.4.0-IBM-MQC-Redist-LinuxX64.tar.gz` in einem Verzeichnis in Ihrer x86 Linux-Umgebung.
  - f) Legen Sie den Pfad zu dem Verzeichnis fest, in dem Sie den Redistributable Java-Client entpackt haben.

```
export MQ_JAVA_INSTALL_PATH=/unpack_location/java
```

## Ergebnisse

Sie haben IBM MQ Bridge to blockchain von Ihrem z/OS in Ihre x86 Linux -Umgebung übertragen, IBM JRE 8 und den IBM MQ 9.0.4 Redistributable Java -Client installiert.

## Nächste Schritte

Verwenden Sie die Informationen zu Ihrem z/OS-Warteschlangenmanager und die Berechtigungsnachweise aus Ihrem Blockchain-Netz, um eine Konfigurationsdatei für die IBM MQ Bridge to blockchain zu erstellen.

## 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.
- Sie verfügen über den IBM MQ Bridge to blockchain, den IBM MQ 9.0.4 Redistributable Java -Client und IBM Java runtime environment Version 8 auf Ihrem x86 Linux.

## 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 1.0 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-Netzberechtigungenachweisen“](#) auf Seite 688.
- Hyperledger Fabric-Netz, das in einem Kubernetes-Cluster in IBM Cloud (formerly Bluemix) ausgeführt wird. Weitere Informationen finden Sie unter [Entwickeln in einer Cloud-Sandbox auf IBM Blockchain Plattform](#) und [„Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters“](#) auf Seite 690.

Weitere Informationen zu den Bedeutungen und Optionen für alle IBM MQ Bridge to blockchain-Parameter finden Sie im Befehl `runmqbcb`. Sie müssen Ihre eigenen Sicherheitsanforderungen berücksichtigen und die Parameter anpassen, die für Ihre Implementierung geeignet sind.

## 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 z/OS-Warteschlangenmanager. Führen Sie das Bridge-Script im Verzeichnis `bin` der Position aus, an der Sie die Bridge entpackt haben, als Sie sie in Task [„IBM MQ Advanced for z/OS VUE für](#)

die Verwendung mit Blockchain konfigurieren” auf Seite 835 aus Ihrer z/OS -Umgebung verschoben haben.

```
./runmqbc -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 z/OS-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 „6” auf Seite 840 verwenden möchten, müssen Sie JNDI oder CCDT verwenden und **MQ CCDT URL** oder **JNDI implementation class** und **JNDI provider URL** entsprechend angeben.

```
Connection to Queue Manager
-----
Queue Manager                : [z/OS_qmgr_name]
Bridge Input Queue           : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
Bridge User Identity Queue   : [SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE]
MQ Channel                   : [SYSTEM.DEF.SVRCONN]
MQ Conname                   : [host1.example.com(3714)]
MQ CCDT URL                  : []
JNDI implementation class    : []
JNDI provider URL           : []
MQ Userid                    : []
MQ Password                  : []
```

3. Geben Sie die Anmeldedetails für die Zertifizierungsstelle für Ihr Blockkettennetz ein.

Die Standardwerte für Ihre lokalen Hyperledger Fabric- und Kubernetes-Cluster-Beispiele sind *admin* für **Userid** und *adminpw* für **Enrollment Secret**. Wenn Sie diese Werte für Ihr Blockkettennetz geändert haben, stellen Sie sicher, dass Sie die richtigen Werte für die Konfiguration der Bridge verwenden.

```
Blockchain - User Identification
-----
Blockchain Userid            : []admin
Enrollment Secret           : []*****
```

4. Geben Sie die ID des Mitgliedschaftsservice-Providers (**MSPid**) ein, der die Mitgliedschaften und Identitätsregeln für Ihr Blockkettennetz regelt.

Geben Sie in Ihrer Berechtigungsnachweisdatei den Parameter **msp\_id** für den **Organisation Name** und den **Organisation MSPId** an. Verwenden Sie in „Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungsnachweisen” auf Seite 688 den Wert **CORE\_PEER\_LOCALMSPID** aus dem Abschnitt `peer` der Datei. Verwenden Sie in der „Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters” auf Seite 690 den Wert **mSPID**.

```
Blockchain - Organisation Identification
-----
Organisation Name           : []Org1MSP
Organisation MSPId          : []Org1MSP
```

5. Geben Sie die Standortwerte für die Blockkettennetzserver ein:

Geben Sie in „[Beispiel für eine Datei mit Hyperledger Fabric-Netzberechtigungs-nachweisen](#)“ auf Seite 688 die Namen und Server: Portpositionen für Zertifizierungsstellen-, Peer- und Anordnungsknotenelemente an.

```
Blockchain server locations
-----
Certificate Authority servers : [ca.example.com Docker_container_host:7054] (for exam
ple ca.example.com localhost:7054)
Peer servers                  : [peer0 localhost:7051]
Orderer servers               : [orderer0 localhost:7050]
Peer Event servers           : [peer0 localhost:7053]
Location of PEM file for Blockchain certificate : []
```

Geben Sie in „[Beispiel für eine Netzkonfigurationsdatei des Kubernetes-Containerclusters](#)“ auf Seite 690 die Namen und Server: Portpositionen für Zertifizierungsstellen-, Peer- und Anordnungsknotenelemente an.

```
Blockchain server locations
-----
Certificate Authority servers : [CA1 your_blockchain_network_public_ip_address:30000] (for example CA1 123.456.789.10:30000)
Peer servers                  : [blockchain-org1peer1 your_blockchain_network_public_ip_address:30110]
Orderer servers               : [blockchain-orderer your_blockchain_network_public_ip_address:31010]
Peer Event servers           : [blockchain-org1peer1 your_blockchain_network_public_ip_address:30111]
Location of PEM file for Blockchain certificate : []
```

#### 6. Geben Sie Zertifikate für TLS-Verbindungen ein.

Die Brücke fungiert als IBM MQ Java-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 Java-Clients. Die Konfiguration von TLS-Verbindungsdetails wird nur dann bereitgestellt, wenn Sie in Schritt „[2](#)“ auf Seite 839 JNDI- oder CCDT-Informationen angegeben haben.

```
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]
```

#### 7. Geben Sie die Position für die Protokolldatei für den IBM MQ Bridge to blockchain ein.

Sie müssen 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 z/OS-Warteschlangenmanager herzustellen.

## Nächste Schritte

Führen Sie die in „[IBM MQ Bridge to blockchain ausführen](#)“ auf Seite 842 beschriebenen Schritte aus.

### Zugehörige Informationen

[runmqbcb \(IBM MQ Bridge to blockchain ausführen\)](#)

## Sicherheit für Warteschlangen, die mit IBM MQ Bridge to blockchain verwendet werden

Hinweise zur Konfiguration der Sicherheit für z/OS-Warteschlangen, die für die Verwendung mit IBM MQ Bridge to blockchain definiert sind.

In den folgenden Beispielen werden RACF-Profilen gezeigt, die eine Möglichkeit zur Sicherung der Warteschlangen für IBM MQ Bridge to blockchain darstellen.

### RESLEVEL

Die IBM MQ Bridge to blockchain stellt die Verbindung über einen **SVRCONN**-Kanal mit dem **CHINIT** her. Es wird davon ausgegangen, dass eine bestimmte Sicherheitsprüfung für die effektive z/OS-Benutzer-ID, die vom Bridge-Benutzer verwendet wird, erforderlich ist. Dies bedeutet, dass wir sicherstellen müssen, dass Benutzer-IDs auf **CHINIT**-Tasks überprüft werden. Die Berechtigung für das Profil **RESLEVEL** bestimmt, ob nur eine Benutzer-ID (die Kanalbenutzer-ID) geprüft wird, oder zwei Benutzer-IDs (sowohl die **channel**-Benutzer-ID als auch die **CHINIT**-Benutzer-ID) werden überprüft. Beispiel:

- Dieser Code erteilt **READ** die Berechtigung für **CHINIT** im **RESLEVEL**-Profil. Daher werden nur die **channel**-Benutzer-IDs überprüft.

```
PERMIT RESLEVEL CLASS(MQADMIN) ID(CHINIT) ACCESS(READ)
```

- Dieser Code erteilt **CHINIT** keine Berechtigung für das **RESLEVEL**-Profil. Daher werden zwei Benutzer-IDs überprüft, und der Benutzer-ID von **CHINIT** müssen zusätzliche Berechtigungen erteilt werden.

```
PERMIT RESLEVEL CLASS(MQADMIN) ID(CHINIT) ACCESS(NONE)
```

Im nächsten Abschnitt werden die Zeilen des Codes hervorgehoben, die zusätzliche Berechtigungen erteilen.

Weitere Informationen finden Sie in [Client-MQI-Anforderungen](#).

### Warteschlangenressourcenberechtigungen

Sperrern Sie die Identitätswarteschlange und lassen Sie die Bridge-ID für die Eingabe und Ausgabe zu verwenden.

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE CLASS(MQQUEUE) ID(MQBBCART) ACCESS(UPDATE)
PERMIT SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE CLASS(MQQUEUE) ID(CHINIT) ACCESS(UPDATE)
```

Bridge-ID kann Warteschlange für Eingabe öffnen

```
DEF QL(CARTAX.BLOCKCHAIN.INPUT.QUEUE) LIKE(SYSTEM.BLOCKCHAIN.INPUT.QUEUE)
RDEFINE MQQUEUE CARTAX.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT CARTAX.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBBCART) ACCESS(UPDATE)
PERMIT APPL1.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(CHINIT) ACCESS(UPDATE)
```

Anwendungs-IDs in der APPCART-Gruppe können die Anforderungswarteschlange für die Ausgabe öffnen.

```
PERMIT CARTAX.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(APPCART) ACCESS(UPDATE)
```

Profil zur Abdeckung von Anwendungsantwortwarteschlangen

```
RDEFINE MQQUEUE CARTAX.APP.REPLY.** UACC(NONE)
```

Anwendungs-IDs in APPCART-Gruppe können Antwortwarteschlange für die Eingabe öffnen.

```
RDEFINE MQADMIN CONTEXT.CARTAX.APP.REPLY.** UACC(NONE)
PERMIT CARTAX.APP.REPLY.** CLASS(MQQUEUE) ID(APPCART) ACCESS(UPDATE)
```

Bridge-ID kann Antwortwarteschlange für Ausgabe öffnen und mit **set\_identity\_context** einlegen

```
PERMIT CARTAX.APP.REPLY.** CLASS(MQQUEUE) ID(MQBBCART) ACCESS(UPDATE)
PERMIT CONTEXT.CARTAX.APP.REPLY.** CLASS(MQADMIN) ID(MQBBCART) ACCESS(UPDATE)
PERMIT CARTAX.APP.REPLY.** CLASS(MQQUEUE) ID(CHINIT) ACCESS(UPDATE)
PERMIT CONTEXT.CARTAX.APP.REPLY.** CLASS(MQADMIN) ID(CHINIT) ACCESS(UPDATE)
```

## Zugehörige Tasks

„IBM MQ Bridge to blockchain-Clientbeispiel ausführen“ auf Seite 846

Sie können das mit dem IBM MQ Bridge to blockchain bereitgestellte JMS-Clientbeispiel verwenden, um eine Nachricht in die Eingabewarteschlange zu stellen, die von der Blockchain-Bridge geprüft wird, und die empfangene Antwort anzuzeigen.

## Zugehörige Informationen

Profile für die Warteschlangensicherheit

API-Ressourcenzugriffsschutz-Kurzreferenz

## V 9.0.3 z/OS MQ Adv. VUE IBM MQ Bridge to blockchain ausführen

Führen Sie die IBM MQ Bridge to blockchain aus, um eine Verbindung zur IBM Blockchain und IBM MQ herzustellen. Wenn eine Verbindung besteht, ist die Brücke bereit, Abfragenachrichten zu verarbeiten, sie an Ihr 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 z/OS-Warteschlangenmanager, den Sie mit der Bridge verwenden möchten.
2. Starten Sie die IBM MQ Bridge to blockchain, um eine Verbindung zu Ihrem Blockchain-Netz und zu Ihrem z/OS-Warteschlangenmanager herzustellen.

Führen Sie das Bridge-Script aus dem Verzeichnis bin der Position aus, an der Sie die Bridge entpackt haben, als Sie sie aus Ihrer z/OS-Umgebung in der Task „IBM MQ Advanced for z/OS VUE für die Verwendung mit Blockchain konfigurieren“ auf Seite 835 verschoben haben.

```
./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:

```
Fri Oct 06 06:32:11 PDT 2017 IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2023.
```

```
Fri Oct 06 06:32:17 PDT 2017 Ready to process input messages.
```

3. Optional: Beheben Sie Fehler bei den Verbindungen zu Ihrem z/OS-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/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.

```
./runmqbc -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 sich mit Ihrem Warteschlangenmanager und dem Blockchain-Netz verbunden.

## Nächste Schritte

- Führen Sie die Schritte in „IBM MQ Bridge to blockchain-Clientbeispiel ausführen“ auf Seite 846 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](#).

## V 9.0.3 z/OS MQ Adv. VUE Nachrichtenformate für IBM MQ Bridge to blockchain

Informationen zur Formatierung der Nachrichten, die von der IBM MQ Bridge to blockchain gesendet und empfangen werden.

Eine Anwendung fordert an, dass IBM MQ Bridge to blockchain eine Abfrage oder Aktualisierung von Informationen ausführt, die in der Blockchain gespeichert sind. Die Anwendung führt dies durch die Anforderung einer Anforderungsnachricht in die Brückenanforderungswarteschlange durch. Die Ergebnisse der Abfrage oder die Aktualisierung werden von der Brücke 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 Nachrichten, die von der Bridge konsumiert und erstellt werden, sind Text-(MQSTR) Nachrichten im JSON-Format. Die Eingabemessage ist eine einfache JSON-Datei und Programme können Zeichenfolgenverkettungen verwenden, um sie zu generieren. Alle Felder mit Ausnahme von **args** sind erforderlich, die Argumentliste für dieses Feld erfordert Kenntnisse der Funktionen des gespeicherten Kettencodes.

## Anforderungsnachrichtenformat

Format der Eingabemessage:

```
{ "function": functionName,
  "channel" : chainName,
  "chaincodeName" : codeName,
  "args" : [ argument list]
}
```

Für das lokale Hyperledger-Netzbeispiel mit dem Beispiel für die Arbeit [Fabcar](#).

- Wenn Sie die Abfragemessage verwenden möchten, die die Funktion `queryAllCars` in dem Chaincode 'fabcar' aufruft, die eine Liste von JSON-Objekten zurückgibt, die die in der Blockkette gehaltenen Autodetails darstellen, formatieren Sie die Message wie folgt:

```
{ "function": "queryAllCars",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": []
}
```

Beispielantwort:

```

{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": [
    {"Record": {"owner": "Tomoko", "colour": "blue", "model": "Prius", "make": "Toyota"}, "Key": "CAR0"},
    {"Record": {"owner": "Brad", "colour": "red", "model": "Mustang", "make": "Ford"}, "Key": "CAR1"},
    {"Record": {"owner": "Jin Soo", "colour": "green", "model": "Tucson", "make": "Hyundai"}, "Key": "CAR2"},
    {"Record": {"owner": "Max", "colour": "yellow", "model": "Passat", "make": "Volkswagen"}, "Key": "CAR3"},
    {"Record": {"owner": "Adriana", "colour": "black", "model": "S", "make": "Tesla"}, "Key": "CAR4"},
    {"Record": {"owner": "Michel", "colour": "purple", "model": "205", "make": "Peugeot"}, "Key": "CAR5"},
    {"Record": {"owner": "Aarav", "colour": "white", "model": "S22L", "make": "Chery"}, "Key": "CAR6"},
    {"Record": {"owner": "Pari", "colour": "violet", "model": "Punto", "make": "Fiat"}, "Key": "CAR7"},
    {"Record": {"owner": "Valeria", "colour": "indigo", "model": "Nano", "make": "Tata"}, "Key": "CAR8"},
    {"Record": {"owner": "Shotaro", "colour": "brown", "model": "Barina", "make": "Holden"}, "Key": "CAR9"}
  ]
}

```

Die Antwortnachricht enthält alle Fahrzeugdatensätze, die derzeit in der Blockkette enthalten sind.

- Wenn Sie die Aktualisierungsnachricht verwenden möchten, die die Funktion `createCar` im Beispielschaincode `'fabcar'` aufruft, die einen neuen Autoeintrag in der Blockchain ledger erstellt, formatieren Sie die Nachricht wie folgt:

```

{ "function": "createCar",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": ["CAR10", "Ford", "Mustang GT", "Blue", "Bob"]
}

```

Beispielantwort:

```

{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}

```

Um zu überprüfen, ob der neue Autoeintrag in der Blockchain hinzugefügt wird, können Sie die Anfangsnachricht erneut verwenden, die alle Autos zurückgibt.

Für das Beispiel eines Kubernetes-Clusternetzes mit dem Arbeitsdemo `example02`.

- Um die Abfragenachricht zu verwenden, die die Funktion `query` in dem chaincode `example02` aufruft, der den Wert für die Entität `"a"` in der Blockchain ledger zurückgibt, formatieren Sie die Nachricht wie folgt:

```

{ "function": "query",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a"]
}

```

Beispielantwort:

```

{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": "100"
}

```

- Wenn Sie die Nachricht verwenden möchten, die die Aufruffunktion `example02`-Kettencode aufruft, die die im ersten Argument angegebene Entität verringert und die Entität, die im zweiten Argument

angegeben ist, um den im dritten Argument angegebenen Wert erhöht, formatieren Sie die Nachricht wie folgt:

```
{ "function": "invoke",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a", "b", "10"]
}
```

Folgende Werte sind möglich:

- Vorher: a=100, b=200
- Nach: a=90, b=210

Beispielantwort:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

Wenn Sie die neuen Werte überprüfen möchten, müssen Sie eine neue Nachrichtenabfragenachrichtis übergeben, in der Sie nach Werten von **"a"** und **"b"** suchen.

## Antwortnachrichtenformat

Antwortnachrichten haben ihre Korrelations-ID auf die Nachrichten-ID der eingehenden Nachricht gesetzt. Alle vom Benutzer definierten Eigenschaften werden von der Eingabe in die Ausgabenachrichten kopiert. Die Benutzer-ID in der Antwort wird durch den Kontext der Identitätsgruppe auf die Benutzer-ID des Absenders gesetzt.

Ein Beispiel für eine erfolgreiche Verarbeitung:

```
{ "data": "500", "message": "OK", "statusCode": 200, "statusType": "SUCCESS" }
```

Die Antwortdaten in dieser Nachricht sind alle, die aus der chaincode-Antwort generiert werden (Byte, die in eine UTF-8-Zeichenfolge konvertiert wurden).

Alle Fehlerantworten haben die gleichen Felder, unabhängig davon, ob sie von der Brücke selbst, von den Aufrufen zur Blockkette oder vom Aufruf des Kettencodes generiert werden. Beispiel:

- Ungültiger Kanalname

```
{
  "message": "Bad newest block expected status 200 got 404, Chain myUnknownChannel",
  "statusCode": 404,
  "statusType": "FAILURE"
}
```

- Falsche JSON-Eingabenachricht

```
{
  "message": "Error: Cannot parse message contents.",
  "statusCode": 2110,
  "statusType": "FAILURE"
}
```

- Falsche Parameter in 'chaincode'

```
{
  "message": "Sending proposal to fabric-peer-1a failed because of gRPC failure=Status{code=UNKNOWN, description={\"Error\": \"Nil amount for c\"}, cause=null}",
  "statusCode": 500,
  "statusType": "FAILURE"
}
```

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**).

## V 9.0.3 z/OS MQ Adv. VUE IBM MQ Bridge to blockchain-Clientbeispiel ausführen

Sie können das mit dem IBM MQ Bridge to blockchain bereitgestellte JMS-Clientbeispiel verwenden, um eine Nachricht in die Eingabewarteschlange zu stellen, die von der Blockchain-Bridge geprüft wird, und die empfangene Antwort anzuzeigen.

### Vorbereitende Schritte

Ihre IBM MQ Bridge to blockchain ist aktiv und ist mit Ihrem IBM MQ Advanced-Warteschlangenmanager und Ihrem Blockchain-Netz verbunden und ist bereit, Eingabenachrichten zu verarbeiten.

### Informationen zu diesem Vorgang

Suchen Sie die JMS-Beispielanwendung im Verzeichnis samp von IBM MQ Bridge to blockchain.

### 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 die JSON-Anforderungsnachricht, die an die Bridge gesendet werden soll:

```
// Create the JSON request message.
// Modify "query", "exampleBlockchainChannelName", and "exampleChaincodeName" to
// match your deployed blockchain chaincode.
// The "operation" field is optional, but recommended. It should be set to QUERY
// or UPDATE to match what the chaincode is going to do.

JSONObject inputMsg = new JSONObject();
inputMsg.put("operation", "QUERY");

inputMsg.put("function", "query");
inputMsg.put("channel", "exampleBlockchainChannelName");
inputMsg.put("chaincodeName", "exampleChaincodeName");

// Create the JSON arguments for the request message.
// Modify "a" to match your deployed blockchain chaincode
// requirements, and add further arguments as necessary

JSONArray myArgs = new JSONArray();
myArgs.add("a");
inputMsg.put("args", myArgs);

TextMessage message = session.createTextMessage(inputMsg.serialize());
message.setJMSReplyTo(replyToQueue);
```

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_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar SimpleBCBClient.java
```

3. Führen Sie die kompilierte Klasse aus.

```
java -cp $MQ_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar:. SimpleBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
```

Created the message. Starting the connection  
Sent message:

```
JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 2
JMSDeliveryDelay: 0
JMSDeliveryTime: 1508427559117
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:414d5120424342514d2020202020209063e859ea36aa24
JMSTimestamp: 1508427559117
JMSCorrelationID: null
JMSDestination: queue:///APPL1.BLOCKCHAIN.INPUT.QUEUE
JMSReplyTo: queue:///APPL1.BLOCKCHAIN.REPLY.QUEUE
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 0
  JMSXUserID: USER1
  JMS_IBM_PutApplType: 6
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15391912
{"args":["a"],"function":"query","channel":"exampleBlockchainChannelName","operati
on":"QUERY","chaincodeName":"exampleChaincodeName"}
```

Antwortnachricht:

```
JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:c3e2d840e2e2f0f840404040404040d2afa27229838af2
JMSTimestamp: 1497439784000
JMSCorrelationID: ID:414d5120424342514d2020202020209063e859ea36aa24 *(JMSMessageID of
the input message)
JMSDestination: null
JMSReplyTo: null
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 1
  JMSXUserID: USER1
  JMS_IBM_Character_Set: UTF-8
  JMS_IBM_Encoding: 273
  JMS_IBM_Format: MQSTR
  JMS_IBM_MsgType: 8
  JMS_IBM_PutApplType: 2
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15392014
}
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
Response text:
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
SUCCESS
```

Wenn der Client eine Zeitlimitüberschreitung beim Warten auf die Antwort empfängt, überprüfen Sie, ob die Brücke aktiv ist.



## Bemerkungen

---

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation 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 IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte 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 dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe  
IBM Europe, Middle East and 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:

Lizenzierung von geistigem Eigentum

IBM Japan, Ltd.

**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 Angaben in dieser Veröffentlichung werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann 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 and 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.

Die 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. Um diese so realistisch wie möglich zu gestalten, enthalten sie auch Namen von Personen, Firmen, Marken und Produkten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musterprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos (d. h. ohne Zahlung an IBM) 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 Musterprogramme geschrieben werden. 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 MQ zu 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](http://www.ibm.com/legal/copytrade.shtml). Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind 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 (<http://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: