

9.0

*IBM MQ Konfigurationsreferenz*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 213 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 IBMsenden, erteilen Sie IBM ein nicht ausschließliches Recht, die Informationen in beliebiger Weise zu verwenden oder zu verteilen, ohne dass eine Verpflichtung für Sie entsteht.

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

---

# Inhaltsverzeichnis

<b>Konfigurationsreferenz.....</b>	<b>5</b>
IBM MQ-Beispielkonfiguration für alle Plattformen.....	5
Verwendung der Kommunikationsbeispiele.....	7
Unterstützung mehrerer Threads — Pipelining.....	8
IBM MQ-Beispielkonfiguration unter AIX.....	9
IBM MQ-Beispielkonfiguration für HP-UX.....	17
MQ-Beispielkonfiguration für IBM i.....	23
MQ-Beispielkonfiguration für Linux.....	41
MQ-Beispielkonfiguration für Solaris.....	49
IBM MQ-Beispielkonfiguration für Windows.....	55
MQ-Beispielkonfiguration für z/OS.....	63
MQ-Beispielkonfiguration für z/OS unter Verwendung von QSGs.....	69
MQ-Beispielkonfiguration für z/OS unter Verwendung der gruppeninternen Warteschlangen- steuerung.....	77
Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen.....	85
IBM MQ-Dateiberechtigungen in /opt/mqm mit setuid für mqm.....	89
IBM MQ-Dateisystemberechtigungen unter Windows.....	91
Benennungseinschränkungen für Warteschlangen.....	92
Benennungseinschränkungen für andere Objekte.....	94
Auflösung des Warteschlangennamens.....	95
Was ist die Auflösung von Warteschlangennamen?.....	97
System- und Standardobjekte.....	98
Objekte der Windows-Standardkonfiguration.....	101
SYSTEM.BASE.TOPIC.....	103
System- und Standardobjekte für IBM i.....	104
Informationen zu Zeilengruppen.....	107
Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen.....	110
Kanalattribute.....	112
Kanalattribute und Kanaltypen.....	112
Kanalattribute in alphabetischer Reihenfolge.....	117
IBM MQ-Clusterbefehle.....	152
Befehle für Warteschlangenmanagerdefinitionen.....	153
Befehle für die Kanaldefinition.....	154
Befehle für die Warteschlangendefinition.....	157
DISPLAY CLUSQMgr.....	159
Befehle SUSPEND QMgr und RESUME QMgr im Cluster.....	162
REFRESH CLUSTER.....	163
RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen.....	164
Lastausgleich in Clustern.....	166
Asynchrones Verhalten von Clusterbefehlen unter z/OS.....	180
Kanalprogramme.....	181
Umgebungsvariablen.....	181
Übergreifende Kommunikationsjobs.....	188
Kanalzustände unter IBM i.....	189
Beispiel für Nachrichtenkanalplanung für UNIX, Linux, and Windows.....	189
Erläuterung der Beispiele für UNIX, Linux, and Windows.....	190
Beispiel für UNIX, Linux, and Windows ausführen.....	193
Beispiel für Nachrichtenkanalplanung für IBM i.....	194
Erläuterung der Beispiele für IBM i.....	194
Beispiel für IBM i ausführen.....	198
Beispiel für IBM i erweitern.....	198
Beispiel für Nachrichtenkanalplanung für z/OS.....	198

Erläuterung der Beispiele für z/OS.....	199
Beispiel für z/OS ausführen.....	202
Beispiel für z/OS erweitern.....	202
Nachrichtenkanalplanungsbeispiel für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange.....	203
Erläuterung des Beispiels der Gruppe mit gemeinsamer Warteschlange für z/OS.....	203
Definitionen der Gruppe mit gemeinsamer Warteschlange.....	205
Warteschlangenmanager QM3 - Beispiel für z/OS.....	206
Beispiel mit Gruppe mit gemeinsamer Warteschlange für z/OS ausführen.....	206
Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek.....	207
Element mqzOSConnectService.....	207
HTTP-Header, die mit dem MQ Service Provider verwendet werden können.....	212
<b>Bemerkungen.....</b>	<b>213</b>
Informationen zu Programmierschnittstellen.....	214
Marken.....	215

# Konfigurationsreferenz

Verwenden Sie die Referenzinformationen in diesem Abschnitt zur Unterstützung beim Konfigurieren von IBM MQ.

Die Konfigurationsreferenzinformationen werden in folgenden Unterabschnitten bereitgestellt:

## Zugehörige Informationen

Konfiguration

 Konfigurationz/OS

## IBM MQ-Beispielkonfiguration für alle Plattformen

In den Konfigurationsbeispielen werden die Aufgaben beschrieben, die zum Aufbau eines funktionierenden IBM MQ-Netztes auszuführen sind. Zu den Aufgaben gehört die Einrichtung von Sender- und Empfängerkanälen von IBM MQ für einen bidirektionalen Nachrichtenfluss zwischen den Plattformen über alle unterstützten Protokolle.

Wenn Sie andere Kanaltypen als Sender-Empfänger-Kanäle verwenden möchten, lesen Sie die Beschreibung des Befehls `DEFINE CHANNEL`.

Abbildung 1 auf Seite 5 ist eine konzeptuelle Darstellung eines einzelnen Kanals und der ihm zugeordneten IBM MQ-Objekte.

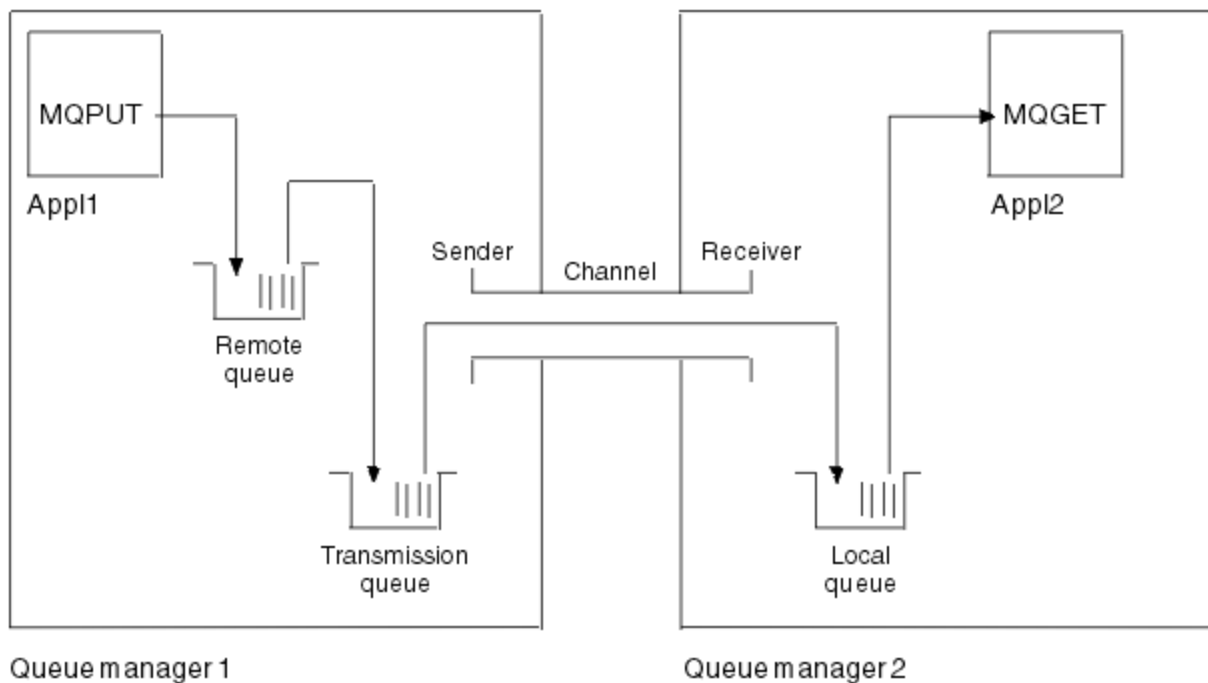


Abbildung 1. IBM MQ-Kanal, der in der Beispielkonfiguration einzurichten ist

Dies ist ein einfaches Beispiel, das eine Einführung in die grundlegenden Elemente des IBM MQ-Netztes geben soll. Es beinhaltet nicht die Auslösefunktion, die im Abschnitt [Kanäle auslösen](#) beschrieben wird.

Objekte in diesem Netz:

- eine ferne Warteschlange
- eine Übertragungswarteschlange
- Eine lokale Warteschlange
- ein Senderkanal

- fein Empfängerkanal

Appl1 und Appl2 sind Anwendungsprogramme; Appl1 reiht Nachrichten ein und Appl2 empfängt sie.

Appl1 reiht Nachrichten in eine ferne Warteschlange ein. In der Definition für diese ferne Warteschlange sind der Name eines Ziel-Warteschlangenmanagers, eine lokale Warteschlange auf diesem Warteschlangenmanager und eine Übertragungswarteschlange auf dem lokalen Warteschlangenmanager angegeben.

Wenn der Warteschlangenmanager die Anforderung von Appl1 zum Einreihen einer Nachricht in die ferne Warteschlange empfängt, ermittelt er anhand der Warteschlangendefinition, dass es sich um eine ferne Zieladresse handelt. Er stellt die Nachricht deshalb zusammen mit einem Übertragungsheader direkt in die Übertragungswarteschlange, die in der Definition angegeben ist. Die Nachricht bleibt in der Übertragungswarteschlange, bis der Kanal verfügbar wird; dies kann sofort der Fall sein.

Ein Senderkanal enthält in seiner Definition einen Verweis auf eine einzige Übertragungswarteschlange. Wenn ein Kanal gestartet wird, und ansonsten während seines normalen Betriebs, schaut er in dieser Übertragungswarteschlange nach und sendet eventuell darin enthaltene Nachrichten an das Zielsystem. Die Nachricht enthält in ihrem Übertragungsheader Informationen zur Zielwarteschlange und zum Warteschlangenmanager.



Die Beispiele für übergreifende Kommunikation beschreiben ausführlich die Erstellung jedes einzelnen der zuvor beschriebenen Objekte für verschiedene Plattformkombinationen.

Auf dem Ziel-Warteschlangenmanager sind Definitionen für die lokale Warteschlange und die Empfängerseite des Kanals erforderlich. Diese Objekte sind voneinander unabhängig und können deshalb in beliebiger Reihenfolge erstellt werden.

Auf dem lokalen Warteschlangenmanager sind Definitionen für die ferne Warteschlange, die Übertragungswarteschlange und die Senderseite des Kanals erforderlich. Da sowohl die Definition der fernen Warteschlange als auch die Kanaldefinition auf den Namen der Übertragungswarteschlange verweisen, ist es ratsam, die Übertragungswarteschlange zuerst zu erstellen.

## Netzinfrastruktur im Beispiel

Die Konfigurationsbeispiele setzen voraus, dass bestimmte Netzinfrastrukturen für bestimmte Plattformen vorhanden sind:

-  z/OS kommuniziert über einen 3745-Netzcontroller (oder funktional entsprechenden Controller), der an einen Token-Ring angeschlossen ist.
-  Solaris ist in einem benachbarten lokalen Netz (LAN) ebenfalls an einen 3745-Netzcontroller (oder funktional entsprechenden Controller) angeschlossen.
- Alle anderen Plattformen sind mit einem Token-Ring-Netz verbunden.

Außerdem wird vorausgesetzt, dass für SNA alle erforderlichen Definitionen im VTAM und im Netzsteuerprogramm (NCP) enthalten und aktiviert sind, sodass die über LAN angeschlossenen Plattformen über das Weitverkehrsnetz (WAN) kommunizieren können.

Entsprechend wird für TCP angenommen, dass die Namensserverfunktion verfügbar ist, entweder über einen Domänennamensserver oder über lokal vorhandene Tabellen (z. B. eine Hostdatei).

## Datenübertragungssoftware im Beispiel

Die Beispiele enthalten funktionsfähige Konfigurationen für folgende Netzsoftwareprodukte:

- SNA (Systems Network Architecture)
  - IBM Personal Communications for Windows 5.9
  - IBM Communications Server for AIX 6.3
  - Hewlett-Packard SNAplus2
  - IBM i
  - Data Connection SNAP-IX Version 7 oder höher

- OS/390 2.4
- TCP
  - Microsoft Windows
  - AIX 4.1.4
  - HP-UX 10.2 oder höher
  - Sun Solaris 2.4 oder höher
  - IBM i
  - TCP for z/OS
  - HP Tru64 UNIX
- NetBIOS
- SPX

### Zugehörige Informationen


#### Konfiguration

 z/OS konfigurieren

## Verwendung der Kommunikationsbeispiele

In den Beispielkonfigurationen werden die Tasks beschrieben, die auf einer einzelnen Plattform zum Konfigurieren der Kommunikation zu einer anderen Plattform ausgeführt werden. Dann werden die Task zum Einrichten eines funktionsfähigen Kanals für diese Plattform beschrieben.



Wo immer möglich, wird versucht, die Informationen so allgemein wie möglich zu halten. Deshalb müssen Sie sich nur auf die beiden relevanten Abschnitte beziehen, wenn Sie zwei beliebige Warteschlangenmanager auf verschiedenen Plattformen miteinander verbinden möchten. Alle Abweichungen oder Sonderfälle werden hervorgehoben. Sie können auch zwei Warteschlangenmanager miteinander verbinden, die auf derselben Plattform (auf verschiedenen Maschinen oder derselben Maschine) aktiv sind. In diesem Fall können alle Informationen dem einen zutreffenden Abschnitt entnommen werden.

 Bevor Sie beginnen, die Anweisungen für Ihre Plattform zu befolgen, müssen Sie unter UNIX, Linux®, and Windows verschiedene Umgebungsvariablen festlegen. Geben Sie dazu einen der folgenden Befehle ein:

-  Unter Windows:

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

Dabei ist `MQ_INSTALLATION_PATH` das Verzeichnis, in dem IBM MQ installiert ist.

-   Unter UNIX and Linux:

```
. MQ_INSTALLATION_PATH/bin/setmqenv
```

Dabei ist `MQ_INSTALLATION_PATH` das Verzeichnis, in dem IBM MQ installiert ist. Dieser Befehl legt die Umgebungsvariablen für die Shell fest, in der Sie zurzeit arbeiten. Wenn Sie eine andere Shell öffnen, müssen Sie erneut den Befehl eingeben.

Es sind Arbeitsblätter vorhanden, in denen Sie die in den Beispielkonfigurationen verwendeten Parameter finden können. Zu jedem Parameter gibt es eine Kurzbeschreibung und Angaben dazu, wo die entsprechenden Werte in Ihrem System zu finden sind. Wenn Sie eigene Werte verwenden, tragen Sie diese in die leeren Stellen des Arbeitsblatts ein. Während Sie den Abschnitt durchgehen, werden Sie Querverweise auf diese Werte finden, sobald sie benötigt werden.

Die Beispiele decken nur Umgebungen ab, in denen kein Clustering verwendet wird. Informationen zur Konfiguration der Datenübertragung bei Verwendung des Clusterings finden Sie im Abschnitt [Warte-](#)

schlangenmanagercluster konfigurieren. Die hier angegebenen Werte für die Konfiguration der Kommunikation gelten weiterhin.

Es sind Beispielkonfigurationen für die folgenden Plattformen vorhanden:

- ▶ **AIX** „IBM MQ-Beispielkonfiguration unter AIX“ auf Seite 9
- ▶ **HP-UX** „IBM MQ-Beispielkonfiguration für HP-UX“ auf Seite 17
- ▶ **IBM i** „MQ-Beispielkonfiguration für IBM i“ auf Seite 23
- ▶ **Linux** „MQ-Beispielkonfiguration für Linux“ auf Seite 41
- ▶ **Solaris** „MQ-Beispielkonfiguration für Solaris“ auf Seite 49
- ▶ **Windows** „IBM MQ-Beispielkonfiguration für Windows“ auf Seite 55
- ▶ **z/OS** „MQ-Beispielkonfiguration für z/OS“ auf Seite 63
- ▶ **z/OS** „MQ-Beispielkonfiguration für z/OS unter Verwendung von QSGs“ auf Seite 69
- ▶ **z/OS** „MQ-Beispielkonfiguration für z/OS unter Verwendung der gruppeninternen Warteschlangensteuerung“ auf Seite 77

## IT-Zuständigkeiten

Die folgenden Erläuterungen sollen als Ausgangspunkt zum besseren Verständnis der in den Beispielen verwendeten Terminologie dienen:

- Systemadministrator: Die Person (oder Gruppe von Personen), die die Software für eine bestimmte Plattform installiert und konfiguriert.
- Netzadministrator: Die Person, die die LAN-Konnektivität, die LAN-Adresszuordnungen, die Namenskonventionen im Netz und andere mit dem Netz verbundenen Tasks steuert. Diese Person kann Mitglied der Systemverwaltungsgruppe oder einer separaten Gruppe sein.

In den meisten z/OS-Installationen ist eine Gruppe für die Aktualisierung der ACF/VTAM-, ACF/NCP- und TCP/IP-Software zur Unterstützung der Konfiguration zuständig. Die Mitglieder dieser Gruppe bilden die Hauptquelle für Informationen, die zur Herstellung einer Verbindung zwischen einer IBM MQ-Plattform und IBM MQ for z/OS benötigt werden. Außerdem können sie Netznamenskonventionen in LANs beeinflussen oder vorgeben. Deshalb müssen Sie deren Zuständigkeitsbereich überprüfen, bevor Sie eigene Definitionen erstellen.

- Ein bestimmter Typ von Administrator, zum Beispiel der CICS-Administrator, wird in den Fällen angegeben, in denen der Verantwortungsbereich der Person genauer beschrieben werden kann.

In den Abschnitten mit den Beispielkonfigurationen wird nicht versucht anzugeben, wer für jeden einzelnen Parameter zuständig ist und ihn festlegen darf. Im Allgemeinen sind mehrere Personen beteiligt.

### Zugehörige Konzepte

„IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5

In den Konfigurationsbeispielen werden die Aufgaben beschrieben, die zum Aufbau eines funktionierenden IBM MQ-Netzes auszuführen sind. Zu den Aufgaben gehört die Einrichtung von Sender- und Empfängerkanälen von IBM MQ für einen bidirektionalen Nachrichtenfluss zwischen den Plattformen über alle unterstützten Protokolle.

### Zugehörige Informationen

[setmqenv](#)

## Multi Unterstützung mehrerer Threads – Pipelining

Sie haben die Option, einem Nachrichtenkanalagenten (MCA) die Übertragung von Nachrichten über mehrere Threads zu gestatten. Dieser Prozess, als *Pipelining* bezeichnet, ermöglicht dem MCA, Nachrichten



effizienter mit weniger Wartestatus zu übertragen, wodurch sich die Kanalleistung verbessern lässt. Für jeden MCA können maximal zwei Threads ausgeführt werden.

Pipelining wird über den Parameter *PipeLineLength* in der Datei "qm.ini" gesteuert. Dieser Parameter wird der Zeilengruppe CHANNELS hinzugefügt:

#### **PipeLineLength= 1 | Anzahl**

Dieses Attribut gibt die maximale Anzahl der von einem Kanal verwendeten parallel ausgeführten Threads an. Der Standardwert ist 1. Alle Werte größer als 1 werden wie der Wert 2 behandelt.

**Anmerkung:** Pipelining ist nur für TCP/IP-Kanäle effektiv.

Bei der Verwendung von Pipelining muss in der Konfiguration der Warteschlangenmanager an beiden Kanalenden der Parameter *PipeLineLength* auf einen größeren Wert als 1 festgelegt sein.

## **Überlegungen zu Kanalexits**

Pipelining kann aus folgenden Gründen bei einigen Exitprogrammen zu Ausfällen führen:

- Exits wurden nicht seriell aufgerufen.
- Exits werden von unterschiedlichen Threads abwechselnd aufgerufen.

Überprüfen Sie die Struktur Ihres Exitprogramms, bevor Sie Pipelining nutzen:

- Exits müssen auf allen Ausführungsstufen wiedereintrittsfähig sein.
- Wenn Sie MQI-Aufrufe verwenden, achten Sie darauf, nicht die gleiche MQI-Kennung zu verwenden, wenn der Exit aus verschiedenen Threads aufgerufen wird.

Ziehen Sie einen Nachrichtenexit in Betracht, der eine Warteschlange öffnet und bei allen nachfolgenden Aufrufen des Exits dessen Kennung für MQPUT-Aufrufe angibt. Dies ist im Pipelining-Modus nicht möglich, da der Exit aus verschiedenen Threads aufgerufen wird. Um dies zu vermeiden, halten Sie eine Warteschlangenkennung für jeden Thread bereit, und überprüfen Sie bei jedem Aufruf des Exits die Thread-ID.

## **AIX IBM MQ-Beispielkonfiguration unter AIX**

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for AIX- und IBM MQ-Produkten.

In den Beispielen werden die folgenden Plattformen behandelt:

- **Windows** Windows
- **UNIX** HP Tru64 UNIX
- **HP-UX** HP-UX
- **Solaris** Solaris
- **Linux** Linux
- **IBM i** IBM i
- **z/OS** z/OS
- VSE/ESA

Unter „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## **AIX Eine LU 6.2-Verbindung einrichten**

Beschreibt die für eine LU 6.2-Verbindung erforderlichen Parameter.

Aktuelle Informationen zum Konfigurieren von SNA über TCP/IP finden Sie in der folgenden IBM Online-dokumentation: [Communications Server for AIX](#).

## Eine TCP-Verbindung einrichten

Das Empfangsprogramm muss unbedingt vor jeglichen Kanälen gestartet werden. Es ermöglicht empfangenden Kanälen das automatische Starten nach dem Erhalt einer Anforderung von einem Senderkanal für eingehende Nachrichten.

Der IBM MQ-Befehl, über den das IBM MQ for TCP-Empfangsprogramm gestartet wird, lautet:

```
runmqclsr -t tcp
```

Wenn Sie das mit UNIX bereitgestellte TCP/IP-Empfangsprogramm verwenden möchten, können Sie alternativ die folgenden Schritte ausführen:

1. Bearbeiten Sie die Datei `"/etc/services"`.

**Anmerkung:** Hierzu müssen Sie sich als Superuser oder Root-Benutzer anmelden. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. Bearbeiten Sie die Datei `"/etc/inetd.conf"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie abgebildet hinzu, und ersetzen Sie dabei `MQ_INSTALLATION_PATH` durch das übergeordnete Verzeichnis, in dem IBM MQ installiert ist:

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name]
```

3. Geben Sie den Befehl `refresh -s inetd` ein.

**Anmerkung:** Sie müssen in der Gruppe `"mqm"` den Zusatz **root** einfügen. Die Primärgruppe muss auf `"mqm"` gesetzt sein. Solange sich `"mqm"` in dem Satz der Gruppen befindet, können Sie die Befehle verwenden. Wenn Sie nur Anwendungen ausführen, die den Warteschlangenmanager verwenden, benötigen Sie keine Gruppenberechtigung für `"mqm"`.

### Wie geht es weiter?

Die Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Fahren Sie dazu mit [„Konfiguration von IBM MQ for AIX“](#) auf Seite 10 fort.

## Konfiguration von IBM MQ for AIX

Definieren von Kanälen zum Abschließen der Konfiguration.

### Anmerkung:

1. Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass der Benutzer und die Gruppe `mqm` erstellt und das Kennwort festgelegt wurden.
2. Wenn die Installation fehlschlägt, weil nicht genügend Speicherplatz im Dateisystem verfügbar ist, können Sie die Größe wie folgt mit dem Befehl `smit C snaerhöhen`. (Geben Sie `df` ein, um den Status des Dateisystems anzuzeigen. Daraufhin wird der volle logische Datenträger angezeigt).

```
-- Physical and Logical Storage  
-- File Systems  
-- Add / Change / Show / Delete File Systems  
-- Journalled File Systems  
-- Change/Show Characteristics of a Journalled File System
```

3. Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

4. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert, wobei `MQ_INSTALLATION_PATH` das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
5. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
6. Unter AIX können Sie einen Trace der IBM MQ-Komponenten durch Eingabe der Standardtracebefehle für IBM MQ oder mithilfe des AIX-Systemtrace starten. Weitere Informationen zum IBM MQ-Trace und zum AIX-Systemtrace finden Sie im Abschnitt [Trace verwenden](#).
7. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

## Basiskonfiguration

1. Erstellen Sie den Warteschlangenmanager aus der AIX-Befehlszeile mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q aix
```

Dabei gilt:

***aix***

ist der Name des Warteschlangenmanagers

***-q***

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

***-u dlqname***

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager aus der AIX-Befehlszeile mithilfe des folgenden Befehls:

```
strmqm aix
```

, wobei *aix* der Name ist, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

3. Starten Sie **runmqsc** aus der AIX-Befehlszeile und erstellen Sie die Warteschlange für unzustellbare Nachrichten durch Eingabe des folgenden Befehls:

```
def ql (dlqname)
```

, wobei *dlqname* der Name ist, den die Warteschlange für unzustellbare Nachrichten bei Erstellung des Warteschlangenmanagers erhalten hat.

## **Kanalkonfiguration für AIX**

Enthält Informationen zum Konfigurieren des Warteschlangenmanagers für einen bestimmten Kanal und eine bestimmte Plattform.

Im folgenden Abschnitt finden Sie ausführliche Informationen zu der Konfiguration des AIX-Warteschlangenmanagers für die Implementierung des in „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf [Seite 5](#) beschriebenen Kanals.

Es ist jeweils der MQSC-Befehl angegeben. Entweder starten Sie **runmqsc** in einer AIX-Befehlszeile und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for AIX und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern,

müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

<i>Tabelle 1. Arbeitsblatt zur Konfiguration für IBM MQ for AIX</i>				
<b>ID</b>	<b>Parameter Name (Parameterna-</b> <b>me)</b>	<b>Re-</b> <b>fe-</b> <b>renz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>AIX</b>	
B	Name der lokalen Warteschlange		<b>AIX.LOCALQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Windows“ auf Seite 58 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (SNA)		<b>AIX.WINNT.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AIX.WINNT.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.AIX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.AIX.TCP</b>	
<b>Verbindung zu IBM MQ for HP Tru64 UNIX</b>				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem HP Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.AIXTCP-</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP-UX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für HP-UX“ auf Seite 19 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>HPUX</b>	
D	Name der fernen Warteschlange		<b>HPUX.REMOTEQ</b>	

Tabelle 1. Arbeitsblatt zur Konfiguration für IBM MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
E	Name der Warteschlange auf fernem System	B	<b>HPUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>HPUX</b>	
G	Name des Senderkanals (SNA)		<b>AIX.HPUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>AIX.HPUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>HPUX.AIX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>HPUX.AIX.TCP</b>	
<b>Verbindung zu IBM MQ for Solaris</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 51 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>SOLARIS</b>	
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>SOLARIS</b>	
G	Name des Senderkanals (SNA)		<b>AIX.SOLARIS.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AIX.SOLARIS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.AIX.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.AIX.TCP</b>	
<b>Verbindung zu IBM MQ for Linux</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 44 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (SNA)		<b>AIX.LINUX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AIX.LINUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.AIX.SNA</b>	

Tabelle 1. Arbeitsblatt zur Konfiguration für IBM MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.AIX.TCP</b>	
<p><b>IBM i</b> <b>Verbindung zu IBM MQ for IBM i</b></p> <p><b>IBM i</b> Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für IBM i“ auf Seite 36 angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	
G	Name des Senderkanals (SNA)		<b>AIX.AS400.SNA</b>	
H	Name des Senderkanals (TCP)		<b>AIX.AS400.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AS400.AIX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AS400.AIXTCP-</b>	
<p><b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS</b></p> <p><b>z/OS</b> Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für z/OS“ auf Seite 65 angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>AIX.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>AIX.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.AIX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>MVS.AIX.TCP</b>	

Tabelle 1. Arbeitsblatt zur Konfiguration für IBM MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<p><b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange</b></p> <p><b>z/OS</b> Die Werte in diesem Abschnitt der Tabelle müssen den in „Beispiel für die Konfiguration gemeinsamer Kanäle“ auf Seite 74 angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>QSG</b>	
D	Name der fernen Warteschlange		<b>QSG.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>QSG.SHAREDQ</b>	
F	Name der Übertragungswarteschlange		<b>QSG</b>	
G	Name des Senderkanals (SNA)		<b>AIX.QSG.SNA</b>	
H	Name des Senderkanals (TCP)		<b>AIX.QSG.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>QSG.AIX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>QSG.AIX.TCP</b>	
<p><b>Verbindung zu MQSeries for VSE/ESA</b></p> <p>Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>AIX.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.AIX.SNA</b>	

**AIX** **Senderkanaldefinitionen unter Verwendung der SNA in IBM MQ for AIX**

Beispielbefehle:

```
def ql (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

def chl (AIX.WINNT.SNA) chltype(sdr) +        G
```

```
trptype(lu62) +
conname('WINNTCPIC') +          17
xmitq(WINNT) +                  F
replace
```

### AIX Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for AIX

Beispielbefehle:

```
def ql (AIX.LOCALQ) replace      B
def chl (WINNT.AIX.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

### AIX TPN-Konfiguration für IBM MQ for AIX

Alternative Möglichkeit, sicherzustellen, dass SNA-Empfängerkanäle ordnungsgemäß aktiviert werden, wenn ein Senderkanal einen Datenaustausch einleitet.

Bei der Konfiguration von AIX Communications Server wurde ein LU 6.2-TPN-Profil erstellt, das den vollständigen Pfad zu einem ausführbaren TP-Programm enthält. Im Beispiel trug die Datei den Namen "u/interop/AIX.crs6a". Sie können einen beliebigen Namen wählen, es kann jedoch sinnvoll sein, den Namen Ihres Warteschlangenmanagers darin aufzunehmen. Die ausführbare Datei muss Folgendes enthalten:

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m aix
```

Dabei ist *aix* der Name des Warteschlangenmanagers (A) und *MQ\_INSTALLATION\_PATH* das übergeordnete Verzeichnis, in dem IBM MQ installiert ist. Nachdem diese Datei erstellt wurde, aktivieren Sie deren Ausführung durch den Befehl:

```
chmod 755 /u/interop/AIX.crs6a
```

Alternativ zum Erstellen einer ausführbaren Datei können Sie den Pfad im Fenster "LU 6.2 TPN-Profil hinzufügen (Add LU 6.2 TPN Profile)" mithilfe von Befehlszeilenparametern angeben.

Indem Sie durch eine der beschriebenen Vorgehensweisen einen Pfad angeben, stellen Sie sicher, dass SNA-Empfängerkanäle richtig aktiviert werden, wenn ein Senderkanal einen Datenaustausch einleitet.

### AIX Senderkanaldefinitionen über TCP in IBM MQ for AIX

Beispielbefehle:

```
def ql (WINNT) +                F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +        D
  rname(WINNT.LOCALQ) +        E
  rqmname(WINNT) +             C
  xmitq(WINNT) +               F
  replace

def chl (AIX.WINNT.TCP) chltype(sdr) + H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) +               F
  replace
```

### AIX Empfängerkanaldefinitionen unter Verwendung von TCP in IBM MQ for AIX

Beispielbefehle:



```
def ql (AIX.LOCALQ) replace B
def chl (WINNT.AIX.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

## HP-UX IBM MQ-Beispielkonfiguration für HP-UX

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for HP-UX- und IBM MQ-Produkten.

Die folgenden Plattformen werden behandelt:

- Windows
- AIX
- HP Tru64 UNIX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf [Seite 5](#) finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

### HP-UX Eine LU 6.2-Verbindung einrichten

Beschreibt die für eine LU 6.2-Verbindung erforderlichen Parameter.

Aktuelle Informationen zum Konfigurieren von SNA über TCP/IP finden Sie in der IBM Onlinedokumentation unter [Communications Server](#) sowie in der folgenden HP-Onlinedokumentation: [HP-UX SNAplus2 Installation Guide](#).

### HP-UX Eine TCP-Verbindung einrichten

Alternative Möglichkeiten zum Verbindungsaufbau und weitere Schritte.

Das Empfangsprogramm muss unbedingt vor jeglichen Kanälen gestartet werden. Es ermöglicht empfangenden Kanälen das automatische Starten nach dem Erhalt einer Anforderung von einem Senderkanal für eingehende Nachrichten.

Wenn Sie das mit UNIX bereitgestellte TCP/IP-Empfangsprogramm verwenden möchten, können Sie alternativ die folgenden Schritte ausführen:

1. Bearbeiten Sie die Datei `"/etc/services"`.

**Anmerkung:** Hierzu müssen Sie sich als Superuser oder Root-Benutzer anmelden. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. Bearbeiten Sie die Datei `"/etc/inetd.conf"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie abgebildet hinzu, und ersetzen dabei `MQ_INSTALLATION_PATH` durch das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
[-m queue.manager.name]
```

3. Ermitteln Sie die Prozess-ID von InetD durch Eingabe folgenden Befehls:

```
ps -ef | grep inetd
```

4. Führen Sie folgenden Befehl aus:

```
kill -1 inetd processid
```

**Anmerkung:** Sie müssen in der Gruppe "mqm" den Zusatz **root** einfügen. Sie müssen die Primärgruppe nicht auf "mqm" setzen. Solange sich "mqm" in dem Satz der Gruppen befindet, können Sie die Befehle verwenden. Wenn Sie nur Anwendungen ausführen, die den Warteschlangenmanager verwenden, benötigen Sie keine Gruppenberechtigung für "mqm".

## Wie geht es weiter?

Die Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Fahren Sie dazu mit „[Konfiguration von IBM MQ for HP-UX](#)“ auf Seite 18 fort.

## HP-UX Konfiguration von IBM MQ for HP-UX

Beschreibt das Definieren der Kanäle, um die Konfiguration abzuschließen.

Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass der Benutzer und die Gruppe *mqm* erstellt und das Kennwort festgelegt wurden.

Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

### Anmerkung:

1. Beispielprogramme sind im Verzeichnis *MQ\_INSTALLATION\_PATH/samp* installiert, wobei *MQ\_INSTALLATION\_PATH* das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
2. Fehlerprotokolle werden unter *"/var/mqm/qmgrs/qmgrname/errors"* gespeichert.
3. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

## Basiskonfiguration

1. Erstellen Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q hpuX
```

Dabei gilt:

### **hpuX**

ist der Name des Warteschlangenmanagers

### **-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

### **-u dlqname**

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten. Er legt das Attribut DEADQ des Warteschlangenmanagers fest, erstellt jedoch nicht die Warteschlange für unzustellbare Nachrichten.

2. Starten Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
strmqm hpux
```

, hierbei ist *hpux* der Name, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

### HP-UX Kanalkonfiguration für HP-UX

Enthält Informationen zum Konfigurieren des Warteschlangenmanagers für einen bestimmten Kanal und eine bestimmte Plattform.

Im folgenden Abschnitt finden Sie ausführliche Informationen zu der Konfiguration des HP-UX-Warteschlangenmanagers für die Implementierung des in [„IBM MQ-Beispielkonfiguration für alle Plattformen“](#) auf Seite 5 beschriebenen Kanals.

Es ist jeweils der MQSC-Befehl angegeben. Entweder starten Sie **runmqsc** in einer UNIX-Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for HP-UX und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern, müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

Tabelle 2. Arbeitsblatt zur Konfiguration für IBM MQ for HP-UX				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>HPUX</b>	
B	Name der lokalen Warteschlange		<b>HPUX.LOCALQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">„Kanalkonfiguration für Windows“</a> auf Seite 58 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (SNA)		<b>HPUX.WINNT.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>HPUX.WINNT.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.HPUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.HPUX.TCP</b>	

Tabelle 2. Arbeitsblatt zur Konfiguration für IBM MQ for HP-UX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 1 auf Seite 12</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (SNA)		<b>HPUX.AIX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>HPUX.AIX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AIX.HPUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.HPUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP Tru64 UNIX</b>				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem HP Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.HPUX.TCP</b>	
J	Name des Empfängerkanals (TCP)	H	<b>HPUX.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for Solaris</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 6 auf Seite 51</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>SOLARIS</b>	
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>SOLARIS</b>	
G	Name des Senderkanals (SNA)		<b>HPUX.SOLARIS.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>HPUX.SOLARIS.TCP</b>	

Tabelle 2. Arbeitsblatt zur Konfiguration für IBM MQ for HP-UX (Forts.)





ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.HPUX.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.HPUX.TCP</b>	
<b>Verbindung zu IBM MQ for Linux</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 5 auf Seite 45</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (SNA)		<b>HPUX.LINUX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>HPUX.LINUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.HPUX.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.HPUX.TCP</b>	
 <b>Verbindung zu IBM MQ for IBM i</b>				
 Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 4 auf Seite 37</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	
G	Name des Senderkanals (SNA)		<b>HPUX.AS400.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>HPUX.AS400.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AS400.HPUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AS400.HPUX.TCP</b>	
 <b>Verbindung zu IBM MQ for z/OS</b>				
 Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 8 auf Seite 65</a> angegebenen entsprechen.				

Tabelle 2. Arbeitsblatt zur Konfiguration für IBM MQ for HP-UX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>HPUX.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>HPUX.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.HPUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>MVS.HPUX.TCP</b>	

**Verbindung zu MQSeries for VSE/ESA**

Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.

C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>HPUX.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.HPUX.SNA</b>	

**HP-UX** Senderkanaldefinitionen unter Verwendung der SNA in IBM MQ for HP-UX

Beispielbefehle:

```

def q1 (WINNT) +                                     F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                             D
  rname(WINNT.LOCALQ) +                             E
  rqmname(WINNT) +                                   C
  xmitq(WINNT) +                                     F
  replace

def chl (HPUX.WINNT.SNA) chltype(sdr) +             G
  trptype(lu62) +
  conname('WINNTCPIC') +                             16
  xmitq(WINNT) +                                     F
  replace
  
```

**HP-UX***Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for HP-UX*

Beispielbefehle:

```
def ql (HPUX.LOCALQ) replace B
def chl (WINNT.HPUX.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

**HP-UX***Aufrufbares Transaktionsprogramm für IBM MQ for HP-UX einrichten*

Sicherstellen, dass die SNA-Empfängerkanäle richtig aktiviert werden, wenn ein Senderkanal einen Datenaustausch einleitet.

Dies ist nicht erforderlich bei HP SNAplus2 Release 6.

Während der HP SNAplus2-Konfiguration haben Sie die Definition eines aufrufbaren Transaktionsprogramms erstellt, die auf eine ausführbare Datei verweist. Im Beispiel trug die Datei den Namen /users/interop/HPUX.crs6a. Sie können den Namen frei wählen, es wäre jedoch sinnvoll, den Namen Ihres Warteschlangenmanagers darin anzugeben. Die ausführbare Datei muss Folgendes enthalten:

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m hpux
```

Dabei steht *hpux* für den Namen des Warteschlangenmanagers A und *MQ\_INSTALLATION\_PATH* ist das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Hierdurch wird sichergestellt, dass SNA-Empfängerkanäle richtig aktiviert werden, wenn ein Senderkanal einen Datenaustausch einleitet.

**HP-UX***Senderkanaldefinitionen über TCP in IBM MQ for HP-UX*

Beispielbefehle:

```
def ql (WINNT) + F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) + D
  rname(WINNT.LOCALQ) + E
  rqmname(WINNT) + C
  xmitq(WINNT) + F
  replace

def chl (HPUX.WINNT.TCP) chltype(sdr) + H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) + F
  replace
```

**HP-UX***Empfängerkanaldefinitionen über TCP/IP in IBM MQ for HP-UX*

Beispielbefehle:

```
def ql (HPUX.LOCALQ) replace B
def chl (WINNT.HPUX.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

**IBM i****MQ-Beispielkonfiguration für IBM i**

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for IBM i- und IBM MQ-Produkten auf anderen Plattformen.

Außerdem werden die folgenden weiteren Plattformen behandelt:

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX
- Solaris
- Linux
- z/OS oder MVS
- VSE/ESA

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

### **IBM i Konfigurationsparameter für eine LU 6.2-Verbindung**

Im folgenden Arbeitsblatt sind alle Parameter aufgelistet, die zum Einrichten der Kommunikation von einem IBM i-System zu einer der anderen IBM MQ-Plattformen erforderlich sind. Das Arbeitsblatt enthält Beispiele der Parameter, die in einer Arbeitsumgebung getestet wurden, und ermöglicht Ihnen, Ihre eigenen Werte einzutragen.

Notieren Sie die Werte für diese Konfiguration in dem Arbeitsblatt dieses Abschnitts. Verwenden Sie das Arbeitsblatt mit dem Arbeitsblatt in dem Abschnitt für die Plattform, zu der Sie eine Verbindung herstellen.

Zahlen in der Referenzspalte zeigen an, dass der Wert dem Wert im entsprechenden Arbeitsblatt an anderer Stelle in diesem Abschnitt entsprechen muss. Die folgenden Beispiele in diesem Abschnitt beziehen sich auf die Werte in der ID-Spalte dieser Tabelle.

Die Einträge in der Spalte mit dem Parameternamen werden in „Begriffserklärungen“ auf Seite 26 erläutert.

<i>Tabelle 3. Arbeitsblatt zur Konfiguration von SNA auf einem IBM i-System</i>				
<b>ID</b>	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten</b>				
1	ID des lokalen Netzes		<b>NETID</b>	
2	Name des lokalen Steuerpunkts		<b>AS400PU</b>	
3	LU-Name		<b>AS400LU</b>	
4	LAN-Zieladresse		<b>10005A5962EF</b>	
5	Subsystembeschreibung		<b>QCMN</b>	
6	Leitungsbeschreibung		<b>TOKENRINGL</b>	
7	Ressourcenname		<b>LIN041</b>	
8	Name des lokalen Transaktionsprogramms		<b>MQSERIES</b>	
<b>Verbindung mit einem Windows-System</b>				
9	Netz-ID	2	<b>NETID</b>	
10	Steuerpunktname	3	<b>WINNTCP</b>	
11	LU-Name	5	<b>WINNTLU</b>	
12	Controllerbeschreibung		<b>WINNTCP</b>	



Tabelle 3. Arbeitsblatt zur Konfiguration von SNA auf einem IBM i-System (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
13	Einheit		<b>WINNTLU</b>	
14	Nebeninformationen		<b>NTCPIC</b>	
15	Transaktionsprogramm	7	<b>MQSERIES</b>	
16	LAN-Adapteradresse	9	<b>08005AA5FAB9</b>	
17.	Mode	17.	<b>#INTER</b>	
<b>Verbindung mit einem AIX-System</b>				
9	Netz-ID	1	<b>NETID</b>	
10	Steuerpunktname	2	<b>AIXPU</b>	
11	LU-Name	4	<b>AIXLU</b>	
12	Controllerbeschreibung		<b>AIXPU</b>	
13	Einheit		<b>AIXLU</b>	
14	Nebeninformationen		<b>AIXCPIC</b>	
15	Transaktionsprogramm	6	<b>MQSERIES</b>	
16	LAN-Adapteradresse	8	<b>123456789012</b>	
17.	Mode	14	<b>#INTER</b>	
<b>Verbindung mit einem HP-UX-System</b>				
9	Netz-ID	4	<b>NETID</b>	
10	Steuerpunktname	2	<b>HPUXPU</b>	
11	LU-Name	5	<b>HPUXLU</b>	
12	Controllerbeschreibung		<b>HPUXPU</b>	
13	Einheit		<b>HPUXLU</b>	
14	Nebeninformationen		<b>HPUXCPIC</b>	
15	Transaktionsprogramm	7	<b>MQSERIES</b>	
16	LAN-Adapteradresse	8	<b>100090DC2C7C</b>	
17.	Mode	17.	<b>#INTER</b>	
<b>Verbindung mit einem Solaris-System</b>				
9	Netz-ID	2	<b>NETID</b>	
10	Steuerpunktname	3	<b>SOLARPU</b>	
11	LU-Name	7	<b>SOLARLU</b>	
12	Controllerbeschreibung		<b>SOLARPU</b>	
13	Einheit		<b>SOLARLU</b>	
14	Nebeninformationen		<b>SOLCPIC</b>	
15	Transaktionsprogramm	8	<b>MQSERIES</b>	
16	LAN-Adapteradresse	5	<b>08002071CC8A</b>	

Tabelle 3. Arbeitsblatt zur Konfiguration von SNA auf einem IBM i-System (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
17.	Mode	17.	#INTER	
<b>Verbindung zu einem Linux -System (x86 -Plattform)</b>				
9	Netz-ID	4	NETID	
10	Steuerpunktname	2	LINUXPU	
11	LU-Name	5	LINUXLU	
12	Controllerbeschreibung		LINUXPU	
13	Einheit		LINUXLU	
14	Nebeninformationen		LXCPIIC	
15	Transaktionsprogramm	7	MQSERIES	
16	LAN-Adapteradresse	8	08005AC6DF33	
17.	Mode	6	#INTER	
<b>Verbindung mit einem z/OS-System</b>				
9	Netz-ID	2	NETID	
10	Steuerpunktname	3	MVSPU	
11	LU-Name	4	MVSLU	
12	Controllerbeschreibung		MVSPU	
13	Einheit		MVSLU	
14	Nebeninformationen		MVSCPIC	
15	Transaktionsprogramm	7	MQSERIES	
16	LAN-Adapteradresse	8	400074511092	
17.	Mode	6	#INTER	
<b>Verbindung mit einem VSE/ESA-System</b>				
9	Netz-ID	1	NETID	
10	Steuerpunktname	2	VSEPU	
11	LU-Name	3	VSELU	
12	Controllerbeschreibung		VSEPU	
13	Einheit		VSELU	
14	Nebeninformationen		VSECPIC	
15	Transaktionsprogramm	4	MQ01	MQ01
16	LAN-Adapteradresse	5	400074511092	
17.	Mode		#INTER	

### Begriffserklärungen

Eine Erläuterung der im Arbeitsblatt zur Konfiguration verwendeten Begriffe.

## 1 2 3

Ausführliche Informationen zum Ermitteln der konfigurierten Werte finden Sie unter [„Netzattribute finden“](#) auf Seite 27.

### 4 LAN-Zieladresse

Die Hardwareadresse des Token-Ring-Adapters des IBM i-Systems. Sie können den Wert mithilfe des Befehls `DSPLIND Leitungsbeschreibung (6)` ermitteln.

### 5 Subsystembeschreibung

Dieser Parameter ist der Name eines IBM i-Subsystems, das aktiv ist, während der Warteschlangenmanager verwendet wird. Der Name QCMN wurde verwendet, da es sich um das Kommunikationssystem von IBM i handelt.

### 6 Leitungsbeschreibung

Wenn dieser Parameter angegeben wurde, wird er im Beschreibungsfeld des Ressourcennamens angezeigt. Details finden Sie im Abschnitt [„Den Wert des Ressourcennamens ermitteln“](#) auf Seite 28. Ist der Wert nicht angegeben, müssen Sie eine Leitungsbeschreibung erstellen.

### 7 Ressourcename

Ausführliche Informationen zum Ermitteln des konfigurierten Werts finden Sie unter [„Den Wert des Ressourcennamens ermitteln“](#) auf Seite 28.

### 8 Name des lokalen Transaktionsprogramms

IBM MQ-Anwendungen, die versuchen, Daten mit dieser Workstation auszutauschen, geben einen symbolischen Namen für das Programm an, das auf der Empfangsseite ausgeführt werden soll. Dieser Name ist in der Kanaldefinition auf der Senderseite definiert. Verwenden Sie der Einfachheit halber wenn möglich immer den Transaktionsprogrammnamen MQSERIES und bei Verbindungen mit VSE/ESA, wenn die Länge des Namens auf 4 Byte beschränkt ist, verwenden Sie MQTP.

Weitere Informationen hierzu finden Sie im Abschnitt [Einstellungen auf dem lokalen IBM i-System für eine ferne Warteschlangenmanagerplattform](#).

### 12 Controllerbeschreibung

Dieser Parameter ist ein Aliasname für den Steuerpunktnamen (oder Knotennamen) des Partnersystems. In diesem Beispiel wird zur Vereinfachung der tatsächliche Name des Partners verwendet.

### 13 Einheit

Dieser Parameter ist ein Aliasname für die logische Einheit (LU) des Partnersystems. In diesem Beispiel wird zur Vereinfachung der LU-Name des Partners verwendet.

### 14 Nebeninformationen

Dieser Parameter ist der Name des CPI-C-Nebeninformationsprofils. Sie geben einen eigenen Namen, bestehend aus 8 Zeichen, an.

## **Netzattribute finden**

Der lokale Knoten wurde teilweise als Bestandteil der IBM i-Installation konfiguriert. Geben Sie zur Anzeige der aktuellen Netzattribute den Befehl `DSPNETA` ein.

Sie ändern diese Werte bei Bedarf mit dem Befehl `CHGNETA`. Zum Anwenden der Änderungen ist unter Umständen ein IPL erforderlich.

```

Display Network Attributes
System: AS400PU
Current system name . . . . . : AS400PU
Pending system name . . . . . :
Local network ID . . . . . : NETID
Local control point name . . . . . : AS400PU
Default local location . . . . . : AS400LU
Default mode . . . . . : BLANK
APPN node type . . . . . : *ENDNODE
Data compression . . . . . : *NONE
Intermediate data compression . . . . . : *NONE
Maximum number of intermediate sessions . . . . . : 200
Route addition resistance . . . . . : 128
Server network ID/control point name . . . . . : NETID NETCP

```

More...  
Press Enter to continue.

F3=Exit F12=Cancel

Stellen Sie sicher, dass die Werte für **ID des lokalen Netzes** (1), **Name des lokalen Steuerpunkts** (2) und **Lokale Standardposition** (3) mit den Werten auf Ihrem Arbeitsblatt übereinstimmen.

### **Den Wert des Ressourcennamens ermitteln**

Sie ermitteln den Wert des Ressourcennamens, indem Sie WRKHDWRSC TYPE (\*CMN) eingeben und die Eingabetaste drücken.

Die Anzeige "Work with Communication Resources" (Mit DFV-Ressourcen arbeiten) wird angezeigt. Der Wert **Ressourcenname** ist der Token-Ring-Port. In diesem Beispiel lautet er LIN041.

```

Mit Datenübertragungsressourcen arbeiten
System: AS400PU
Geben Sie Optionen ein, und drücken Sie die Eingabetaste
2=Bearbeiten 4=Entfernen 5=Mit Konfigurationsbeschreibung arbeiten
7=Konfigurationsbeschreibung hinzufügen ...

```

```

Konfiguration
Opt Ressource      Beschreibung Typ Beschreibung
CC02              2636 Comm Prozessor
LIN04             2636 LAN Adapter
LIN041           TOKEN-RING 2636 Token-Ring-Port

```

Unten  
F3=Beenden F5=Aktualisieren F6=Drucken F11=Anzeige Ressourcenadresse/Status  
F12=Abbrechen F23=Weitere Optionen

### **Eine LU 6.2-Verbindung einrichten**

In diesem Abschnitt wird beschrieben, wie Sie eine LU 6.2-Verbindung einrichten.

### **Lokalen Knoten konfigurieren**

Zum Konfigurieren des lokalen Knotens müssen Sie eine Leitungsbeschreibung erstellen und einen Routing-Eintrag hinzufügen.

## Eine Leitungsbeschreibung erstellen

1. Wenn die Leitungsbeschreibung nicht bereits erstellt wurde, verwenden Sie den Befehl CRTLINTRN.
2. Geben Sie die Werte für **Leitungsbeschreibung** (6) und **Ressourcenname** (7) an.

```
Line Desc (Token-Ring) erstellen (CRTLINTRN)

Auswahl eingeben und Eingabetaste drücken.

Leitungsbeschreibung..... TOKENRINGL      Name
Ressourcenname..... LIN041                Name, *NWID
NWI-Typ..... *FR *FR, *ATM
Online bei IPL..... *YES *YES, *NO
Wartezeit anhängen..... *NOWAIT *NOWAIT, 15 -180 (1 Sekunde)
Maximale Steuereinheiten..... 40 1-256
Angehängter NWI..... *NONE Name, *NONE

Unten
F3=Beenden F4=Abfragen F5=Aktualisieren F10=Zusätzliche Parameter F12=Abbrechen
F13=Wie diese Anzeige verwendet wird F24=Weitere Tasten
Parameter LIND erforderlich. +
```

## Einen Routing-Eintrag hinzufügen

1. Geben Sie den Befehl ADDRTGE ein und drücken Sie die Eingabetaste.

```
Leitwegeintrag hinzufügen (ADDRTGE)

Auswahl eingeben und Eingabetaste drücken.

Subsystembeschreibung..... QCMN-Name
Bibliothek..... *LIBL-Name, *LIBL, *CURLIB
Die Folgenummer des Leitwegeintrags. 1 1-9999
Vergleichsdaten:
Vergleichswert.....MQSERIES'

Startposition..... 37 1-80
Aufrufprogramm..... AMQCRC6B Name, *RTGDTA
Bibliothek..... QMAS400 Name, * LI BL, *CURLIB
Klasse..... *SBSD-Name, *SBSD
Bibliothek..... *LIBL-Name, *LIBL, *CURLIB
Maximale aktive Routing-Schritte.. *NOMAX 0-1000, *NOMAX
Speicherpool-ID.... 1 1-10

Unten
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
Parameter SBSDB erforderlich. +
```

2. Geben Sie Ihren Wert für **Subsystembeschreibung** (5) sowie die hier angezeigten Werte für **Folgennummer des Routing-Eintrags**, **Vergleichswert** (8), **Anfangsposition**, **Aufzurufendes Programm** und die **Bibliothek** mit dem Programm an, das aufgerufen werden soll.
3. Geben Sie den Befehl STRSBS *Subsystembeschreibung* (5) ein und drücken Sie die Eingabetaste.



### Verbindung mit Partnerknoten

Um eine Verbindung zu einem Partnerknoten herzustellen, müssen Sie: eine Controllerbeschreibung erstellen, eine Einheitenbeschreibung erstellen, CPI-C-Nebeninformationen erstellen, einen DFV-Eintrag für APPC hinzufügen sowie einen Konfigurationslisteneintrag hinzufügen.

Dieses Beispiel zeigt die Verbindung zu einem Windows-System, die Schritte sind jedoch für alle Knoten gleich.

## Eine Controllerbeschreibung erstellen

1. Geben Sie in die Befehlszeile den Befehl CRTCTLAPPC ein und drücken Sie die Eingabetaste.

```
Create Ctl Desc (APPC) (CRTCTLAPPC)

Auswahl eingeben und Eingabetaste drücken.

Steuereinheitenbeschreibung.... WINNTCP          Name
Linktyp..... *LAN *FAX, *FR, *IDLC,
*LAN...
Online bei IPL..... *NO *YES, *NO

Unten
F3=Beenden   F4=Abfragen   F5=Aktualisieren   F10=Zusätzliche Parameter   F12=Abbrechen
F13=Wie diese Anzeige verwendet wird   F24=Weitere Tasten
Parameter CTLD erforderlich. +
```

2. Geben Sie einen Wert für **Controllerbeschreibung** (12) an und setzen Sie **Verbindungstyp** auf \*LAN und **Online bei IPL** auf\*NO.
3. Drücken Sie zweimal die Eingabetaste und anschließend die Taste F10.

```
Create Ctl Desc (APPC) (CRTCTLAPPC)

Auswahl eingeben und Eingabetaste drücken.

Steuereinheitenbeschreibung.... > WINNTCP-Name
Linktyp..... > *LAN *FAX, *FR, *IDLC, *LAN ...
Online bei IPL..... > *NO *YES, *NO
APPN-fähig..... *YES *YES, *NO
Liste der Wählleitungen..... TOKENRINGL      Name
+ für mehr Werte
Maximale Rahmengröße..... *LINKTYPE 265-16393, 256, 265, 512 ...
Ferne Netzwerk-ID... NETID          Name, *NETATR, *NONE, *ANY
Ferner Kontrollpunkt..... WINNTCP      Name, *ANY
Austausch-ID..... 00000000-FFFFFFF
Anfangsverbindung..... *DIAL *DIAL, *ANS
Wahleinleitung..... *LINKTYPE *LINKTYPE, *IMMED, *DELAY
LAN-ferne Adapteradresse... 10005AFC5D83 000000000001-FFFFFFFFFFFF
APPN-CP-Sitzungsunterstützung.... *YES *YES, *NO
APPN-Knotentyp..... *ENDNODE *ENDNODE, *LENNODE ...
APPN Verbindungsgruppennummer 1          1-20, *CALC
Mehr ...
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

4. Geben Sie die Werte für **Wählleitungsliste** (6), **ID des fernen Netzes** (9), **Fernsteuerpunkt** (10) und **Adresse des fernen LAN-Adapters** (16) an.
5. Drücken Sie die Eingabetaste.

## Eine Einheitenbeschreibung erstellen

1. Geben Sie den Befehl CRTDEVAPPC ein und drücken Sie die Eingabetaste.

```

Device Desc erstellen (APPC) (CRTDEVAPPC)

Auswahl eingeben und Eingabetaste drücken.

Einheitenbeschreibung..... WINNTLU           Name
Ferner Standort..... WINNTLU           Name
Online bei IPL..... *YES *YES, *NO
Lokaler Standort..... AS400LU           Name, *NETATR
Ferne Netzwerk-ID... NETID             Name, *NETATR, *NONE
Angehängter Controller..... WINNTCP      Name
Modus..... *NETATR-Name, *NETATR
+ für mehr Werte
Nachrichtenwarteschlange..... QSYSOPR      Name, QSYSOPR
Bibliothek..... *LIBL-Name, *LIBL, *CURLIB
APPN-fähig..... *YES *YES, *NO
Einzelsitzung:
Einzelsitzung in der Lage... *NO *NO, *YES
Anzahl der Dialoge... 1-512

Unten
F3=Beenden F4=Abfragen F5=Aktualisieren F10=Zusätzliche Parameter F12=Abbrechen
F13=Wie diese Anzeige verwendet wird F24=Weitere Tasten
Parameter DEVD erforderlich. +

```

2. Geben Sie die Werte für **Einheitenbeschreibung** (13), **Ferner Standort** (11), **Lokaler Standort** (3), **ID des fernen Netzes** (9) und **Angehängter Controller** (12) an.

**Anmerkung:** Controller- und Einheitenbeschreibungen müssen nicht manuell erstellt werden, wenn Sie die automatische Konfiguration in IBM i verwenden. Weitere Informationen finden Sie in der Dokumentation zu IBM i.

## CPI-C-Nebeninformationen erstellen

1. Geben Sie den Befehl CRTCSI ein und drücken Sie die Taste F10.

```

Erstellen von Comm Side Information (CRTCSI)

Auswahl eingeben und Eingabetaste drücken.

Seiteninformationen..... NTCPIC           Name
Bibliothek..... *CURLIB Name, *CURLIB
Ferner Standort..... WINNTLU           Name
Transaktionsprogramm..... MQSERIEN

Text 'Beschreibung'..... *BLANK

Zusätzliche Parameter

Einheit..... *LOC-Name, *LOC
Lokaler Standort..... AS400LU           Name, *LOC, *NETATR
Modus..... #INTER Name, *NETATR
Ferne Netzwerk-ID... NETID             Name, *LOC, *NETATR, *NONE
Berechtigung..... *LIBCRTAUT Name, *LIBCRTAUT, *CHANGE ...

Unten
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
Parameter CSI erforderlich.

```

2. Geben Sie die Werte für **Nebeninformationen** (14), **Ferner Standort** (11), **Transaktionsprogramm** (15), **Lokaler Standort** (3), **Modus** und **ID des fernen Netzes** (9) an.
3. Drücken Sie die Eingabetaste.

## Einen DFV-Eintrag für APPC hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDCMNE ein und drücken Sie die Eingabetaste.

DFV-Eintrag hinzufügen (ADDCMNE)

Auswahl eingeben und Eingabetaste drücken.

```
Subsystembeschreibung.... QCMN-Name
Bibliothek..... *LIBL-Name, *LIBL, *CURLIB
Einheit..... WINNTLU Name, generic*, *ALL...
Ferner Standort..... Name
Jobbeschreibung..... *USRPRF Name, *USRPRF, *SBSD
Bibliothek..... Name, *LIBL, *CURLIB
Standardbenutzerprofil..... *NONE Name, *NONE, *SYS
Modus..... *ANY Name, *ANY
Maximale aktive Jobs..... *NOMAX 0-1000, *NOMAX
```

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige

F24 = Weitere Tasten

Parameter SBSDB erforderlich.

2. Geben Sie die Werte für **Subsystembeschreibung** (5) und **Gerät** (13) an und drücken Sie die Eingabetaste.

## Einen Konfigurationslisteneintrag hinzufügen

1. Geben Sie den Befehl ADDCFGLE \*APPNRMT ein und drücken Sie die Taste F4.

Konfigurationslisteneinträge hinzufügen (ADDCFGLE)

Auswahl eingeben und Eingabetaste drücken.

```
Konfigurationslistentyp... > *APPNRMT *APPNLCL, *APPNRMT ...
APPN Eintrag ferner Standort:
Name des fernen Standorts.... WINNTLU Name, generic*, *ANY
Ferne Netzwerk-ID.. NETID Name, *NETATR, *NONE
Lokaler Standortname.... AS400LU Name, *NETATR
Ferner Kontrollpunkt..... WINNTCP Name, *NONE
Kontrollpunktnetz-ID..... NETID Name, *NETATR, *NONE
Standortkennwort..... *NONE
Sichere Position..... *NO *YES, *NO
Einzelsitzung..... *NO *YES, *NO
Lokal gesteuerte Sitzung.. *NO *YES, *NO
Voreingestellte Sitzung... *NO *YES, *NO
Eintrag 'description'..... *BLANK
Anzahl der Dialoge... 10 1-512
+ für mehr Werte
```

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige

F24 = Weitere Tasten

2. Geben Sie die Werte für **Name des fernen Standorts** (11), **ID des fernen Netzes** (9), **Name des lokalen Standorts** (3), **Fernsteuerpunkt** (10) und **Netz-ID des Kontrollpunkts** (9) an.
3. Drücken Sie die Eingabetaste.

### **Wie geht es weiter?**

Die LU 6.2-Verbindung ist nun eingerichtet. Sie können die Konfiguration abschließen.

Fahren Sie dazu mit „Konfiguration von IBM MQ for IBM i“ auf Seite 34 fort.



Wenn TCP bereits konfiguriert ist, fallen keine zusätzlichen Konfigurationsaufgaben an. Ist TCP/IP nicht konfiguriert, müssen Sie eine TCP/IP-Schnittstelle, eine TCP/IP-Loopback-Schnittstelle und eine Standardroute hinzufügen.

### Eine TCP/IP-Schnittstelle hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDTCPIFC ein und drücken Sie die Eingabetaste.

```
TCP/IP-Schnittstelle hinzufügen (ADDTCPICF)

Auswahl eingeben und Eingabetaste drücken.

Internetadresse..... 19.22.11.55
Leitungsbeschreibung..... TOKENRINGL      Name, *LOOPBACK
Teilnetzmaske..... 255.255.0.0
Art des Service..... *NORMAL *MINDELAY, *MAXTHRPUT ..
Maximale Übertragungseinheit... *LIND 576-16388, *LIND
Autostart..... *YES *YES, *NO
PVC Kennung des logischen Kanals                001-FFF
+ für mehr Werte
Zeitlimit für inaktive X.25 -Schaltung... 60 1-600
Maximale virtuelle X.25 -Schaltkreise. 64 0-64
X.25 DDN-Schnittstelle..... *NO *YES, *NO
TRLAN-Bitfolgeplanung..... *MSB *MSB, *LSB

Unten
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

2. Geben Sie die **IP-Adresse**, die **Leitungsbeschreibung** sowie eine **Teilnetzmaske** des Systems an.
3. Drücken Sie die Eingabetaste.

### Eine TCP/IP-Loopback-Schnittstelle hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDTCPIFC ein und drücken Sie die Eingabetaste.

```
TCP-Schnittstelle hinzufügen (ADDTCPICF)

Auswahl eingeben und Eingabetaste drücken.

Internetadresse..... 127.0.0.1
Leitungsbeschreibung..... *LOOPBACK-Name, *LOOPBACK
Teilnetzmaske..... 255.0.0.0
Art des Service..... *NORMAL *MINDELAY, *MAXTHRPUT ..
Maximale Übertragungseinheit... *LIND 576-16388, *LIND
Autostart..... *YES *YES, *NO
PVC Kennung des logischen Kanals                001-FFF
+ für mehr Werte
Zeitlimit für inaktive X.25 -Schaltung... 60 1-600
Maximale virtuelle X.25 -Schaltkreise. 64 0-64
X.25 DDN-Schnittstelle..... *NO *YES, *NO
TRLAN-Bitfolgeplanung..... *MSB *MSB, *LSB

Unten
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
```

2. Geben Sie die Werte für die **IP-Adresse**, die **Leitungsbeschreibung** und die **Teilnetzmaske** an.

## Eine Standardroute hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDTCP RTE ein und drücken Sie die Eingabetaste.

```
TCP Routing hinzufügen (ADDTCP RTE)
```

```
Auswahl eingeben und Eingabetaste drücken.
```

```
Route-Ziel..... *DFTRROUTE  
Teilnetzmaske..... *NONE  
Art des Service..... *NORMAL *MINDELAY, *MAXTHRPUT.  
Nächster Hop..... 19.2.3.4  
Maximale Übertragungseinheit... 576 576-16388, *IFC
```

```
Unten
```

```
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur  
Verwendung dieser Anzeige  
F24 = Weitere Tasten  
Eingabeaufforderung wurde beendet, nachdem Benutzer F12 gedrückt hat.
```

2. Geben Sie die Werte für Ihr Netz ein und drücken Sie die Eingabetaste, um einen Standardrouteneintrag zu erstellen.

## Wie geht es weiter?

Die TCP-Verbindung ist nun eingerichtet. Sie können die Konfiguration abschließen. Fahren Sie dazu mit „Konfiguration von IBM MQ for IBM i“ auf Seite 34 fort.

### Konfiguration von IBM MQ for IBM i

Verwenden Sie zum Konfigurieren von IBM MQ for IBM i den Befehl WRKMQMQ, um das Konfigurationsmenü anzuzeigen.

Starten Sie das TCP-Kanal-Empfangsprogramm mit dem Befehl STRMQMLSR.

Starten Sie einen beliebigen Senderkanal mit dem Befehl STRMQMCHL CHLNAME(*Kanalname*).

Mit dem Befehl WRKMQMQ zeigen Sie das Konfigurationsmenü von IBM MQ an.

**Anmerkung:** AMQ\*-Fehler werden im Protokoll in Relation zu dem Job gesetzt, der den Fehler gefunden hat. Mit dem Befehl WRKACTJOB zeigen Sie die Liste der Jobs an. Suchen Sie unter dem Subsystemnamen QSYSWRK nach dem Job und geben Sie 5 ein, um mit diesem Job zu arbeiten. IBM MQ -Protokolle haben das Präfix AMQ.

### Einen Warteschlangenmanager erstellen

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager der Basiskonfiguration einzurichten.

1. Zunächst müssen Sie einen Warteschlangenmanager erstellen. Geben Sie dazu den Befehl CRTMQM ein und drücken Sie die Eingabetaste.

### Nachrichten-Warteschlangenmanager erstellen (CRTMQM)

Auswahl eingeben und Eingabetaste drücken.

Name des Nachrichten-WS-Managers...

Text 'Beschreibung'..... \*BLANK

Auslöserintervall..... 999999999 0-999999999

Nicht abgegebene Nachrichtenwarteschlange... \*NONE

Standardübertragungswarteschlange... \*NONE

Maximale Griffgrenze..... 256 1-999999999

Maximale nicht festgeschriebene Nachrichten.. 1000 1-10000

Standard-WS-Manager..... \*NO \*YES, \*NO

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen  
zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

2. Geben Sie im Feld **Message Queue Manager name** (Name des Nachrichten-Warteschlangenmanagers) den Namen AS400 ein. Geben Sie im Feld **Undelivered message queue** (Warteschlange für nicht zugestellte Nachrichten) DEAD.LETTER.QUEUE ein.
3. Drücken Sie die Eingabetaste.
4. Starten Sie nun den Warteschlangenmanager, indem Sie den Befehl STRMQM MQMNAME (AS400) eingeben.
5. Erstellen Sie die Warteschlange für nicht zugestellte Nachrichten unter Verwendung der folgenden Parameter. (Ausführliche Informationen und ein Beispiel hierzu finden Sie unter „[Eine Warteschlange definieren](#)“ auf Seite 35.)

```
Local Queue
Queue name :  DEAD.LETTER.QUEUE
Queue type :  *LCL
```

### **Eine Warteschlange definieren**

Mit dem Befehl CRTMQMQ können Sie eine Warteschlange definieren.

Geben Sie in der Befehlszeile den Befehl CRTMQMQ ein.

```
MQM-Warteschlange erstellen (CRTMQMQ)
Auswahl eingeben und Eingabetaste drücken.
Warteschlangenname.....
Warteschlangentyp.....      *ALS, *LCL, *RMT
```

Unten  
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten  
Parameter QNAME erforderlich.

Füllen Sie die beiden Felder in dieser Anzeige aus und drücken Sie die Eingabetaste. Daraufhin wird eine weitere Anzeige mit Eingabefeldern für Ihre weiteren Parameter eingeblendet. Für alle anderen Warteschlangenattribute können Sie die Standardeinstellungen übernehmen.

### **Kanal unter IBM i definieren**

Unter IBM i können Sie einen Kanal mit dem Befehl CRTMQMCHL definieren.

Geben Sie in der Befehlszeile den Befehl CRTMQMCHL ein.

```
Create MQM Channel (CRTMQMCHL)
Type choices, press Enter.
Channel name . . . . .
Channel type . . . . .      *RCVR, *SDR, *SVR, *RQSTR
```

Bottom  
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys  
Parameter CHLNAME required.

Füllen Sie die beiden Felder in dieser Anzeige aus und drücken Sie die Eingabetaste. Daraufhin wird eine weitere Anzeige eingeblendet, in der Sie die Werte für die zuvor angegebenen Parameter festlegen können. Für alle anderen Kanalattribute können Sie die Standardeinstellungen übernehmen.

### **Kanalkonfiguration für IBM i**

Sie müssen Ihre Kanäle konfigurieren, um die Beispielkonfigurationskanäle zu implementieren.

In diesem Abschnitt wird die Konfiguration des IBM i-Warteschlangenmanagers zur Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for IBM i und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Werte im Beispiel für Windows ein.

**Anmerkung:**

1. Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern, müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.
2. Der Pingbefehl für den IBM MQ-Kanal (PNGMQMCHL) wird interaktiv ausgeführt, während das Starten eines Kanals die Übergabe eines Batch-Jobs auslöst. Wenn der Pingbefehl eines Kanals erfolgreich ausgeführt wird, der Kanal jedoch nicht startet, sind möglicherweise das Netz und die IBM MQ-Definitionen korrekt, nicht jedoch die IBM i-Umgebung für den Batch-Job. Stellen Sie beispielsweise sicher, dass QSYS2 nicht nur in Ihre persönliche Bibliotheksliste, sondern auch in den Systemteil der Bibliotheksliste aufgenommen wird.

Ausführliche Informationen und Beispiele zum Erstellen der aufgeführten Objekte finden Sie unter „[Eine Warteschlange definieren](#)“ auf Seite 35 und „[Kanal unter IBM i definieren](#)“ auf Seite 36.

<i>Tabelle 4. Arbeitsblatt zur Konfiguration für IBM i</i>				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>AS400</b>	
B	Name der lokalen Warteschlange		<b>AS400.LOCALQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für Windows</a> “ auf Seite 58 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (SNA)		<b>AS400.WINNT.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AS400.WINNT.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.AS400.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>WINNT.AS400.TCP</b>	
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für AIX</a> “ auf Seite 11 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	

Tabelle 4. Arbeitsblatt zur Konfiguration für IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (SNA)		<b>AS400.AIX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AS400.AIXTCP-</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AIX.AS400.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.AS400.TCP</b>	
<b>Verbindung zu MQSeries for Compaq Tru64 Unix</b>				
Die Werte in diesem Tabellenabschnitt müssen den Werten in Ihrem Compaq Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.AS400.TCP</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AS400.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP-UX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für HP-UX“ auf Seite 19 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>HPUX</b>	
D	Name der fernen Warteschlange		<b>HPUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>HPUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>HPUX</b>	
G	Name des Senderkanals (SNA)		<b>AS400.HPUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>AS400.HPUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>HPUX.AS400.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>HPUX.AS400.TCP</b>	
<b>Verbindung zu IBM MQ for Solaris</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 51 verwendeten Werten wie angegeben entsprechen.				

Tabelle 4. Arbeitsblatt zur Konfiguration für IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
C	Name des fernen Warteschlangenmanagers	A	<b>SOLARIS</b>	
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>SOLARIS</b>	
G	Name des Senderkanals (SNA)		<b>AS400.SOLARIS.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AS400.SOLARIS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.AS400.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.AS400.TCP</b>	
<b>Verbindung zu IBM MQ for Linux</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 44 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (SNA)		<b>AS400.LINUX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>AS400.LINUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.AS400.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.AS400.TCP</b>	
<b>Verbindung zu IBM MQ for z/OS</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für z/OS“ auf Seite 65 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>AS400.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>AS400.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.AS400.SNA</b>	

Tabelle 4. Arbeitsblatt zur Konfiguration für IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
J	Name des Empfängerkanals (TCP)	H	<b>MVS.AS400.TCP</b>	
<b>Verbindung zu MQSeries for VSE/ESA</b>				
Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>AS400.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.AS400.SNA</b>	

**IBM i** Senderkanaldefinitionen für IBM i  
 Beispiel für Senderkanaldefinitionen für SNA und TCP.

### SNA verwenden

```

Local Queue
  Queue name : WINNT
  Queue type : *LCL
  Usage : *TMQ
  F

Remote Queue
  Queue name : WINNT.REMOTEQ
  Queue type : *RMT
  Remote queue : WINNT.LOCALQ
  Remote Queue Manager : WINNT
  Transmission queue : WINNT
  D
  E
  C
  F

Sender Channel
  Channel Name : AS400.WINNT.SNA
  Channel Type : *SDR
  Transport type : *LU62
  Connection name : WINNTCPIC
  Transmission queue : WINNT
  G
  14
  F
    
```

### TCP verwenden

```

Local Queue
  Queue name : WINNT
  Queue type : *LCL
  Usage : *TMQ
  F

Remote Queue
  Queue name : WINNT.REMOTEQ
  Queue type : *RMT
  Remote queue : WINNT.LOCALQ
  Remote Queue Manager : WINNT
  Transmission queue : WINNT
  D
  E
  C
  F

Sender Channel
  Channel Name : AS400.WINNT.TCP
  Channel Type : *SDR
  Transport type : *TCP
  H
    
```



```
Connection name : WINNT.tcpip.hostname
Transmission queue : WINNT F
```

## IBM i Empfängerkanaldefinitionen für IBM i

Beispiel für Empfängerkanaldefinitionen für SNA und TCP.

### SNA verwenden

```
Local Queue
  Queue name : AS400.LOCALQ B
  Queue type : *LCL

Receiver Channel
  Channel Name : WINNT.AS400.SNA I
  Channel Type : *RCVR
  Transport type : *LU62
```

### TCP verwenden

```
Local Queue
  Queue name : AS400.LOCALQ B
  Queue type : *LCL

Receiver Channel
  Channel Name : WINNT.AS400.TCP J
  Channel Type : *RCVR
  Transport type : *TCP
```

## Linux MQ-Beispielkonfiguration für Linux

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for Linux- und IBM MQ-Produkten.

Für folgende Plattformen stehen Beispiele zur Verfügung:

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX
- Solaris
- IBM i
- z/OS
- VSE/ESA

Unter „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## Linux Eine LU 6.2-Verbindung einrichten

Verwenden Sie dieses Arbeitsblatt, um die Werte für Ihre Konfiguration zu erfassen.

**Anmerkung:** Die Informationen in diesem Abschnitt gelten nur für IBM MQ for Linux (x86 -Plattform). Sie gilt nicht für IBM MQ for Linux (Plattformx86-64 ), IBM MQ for Linux (Plattform zSeries s390x ) oder IBM MQ for Linux (Power-Plattform).

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP finden Sie im Administrationshandbuch für Ihre Version von Linux in der folgenden Dokumentation: [Communications Server for Linux library](#).

In einigen Linux-Distributionen wird jetzt der erweiterte inet-Dämon (XINETD) anstelle des inet-Dämons (INETD) verwendet. Die folgenden Anweisungen dienen zum Aufbauen einer TCP-Verbindung mit dem inet-Dämon oder dem erweiterten inet-Dämon.

### inet-Dämon (INETD) verwenden

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

Gehen Sie wie folgt vor, um eine TCP-Verbindung aufzubauen.

1. Bearbeiten Sie die Datei `"/etc/services"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries    1414/tcp    # MQSeries channel listener
```

**Anmerkung:** Zur Bearbeitung dieser Datei müssen Sie als Superuser oder Rootbenutzer angemeldet sein.

2. Bearbeiten Sie die Datei `"/etc/inetd.conf"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name ]
```

3. Ermitteln Sie die Prozess-ID von InetD durch Eingabe folgenden Befehls:

```
ps -ef | grep inetd
```

4. Führen Sie folgenden Befehl aus:

```
kill -1 inetd processid
```

Wenn Sie mehrere Warteschlangenmanager in Ihrem System betreiben und demzufolge mehr als einen Service benötigen, müssen Sie für jeden einzelnen Warteschlangenmanager sowohl in `"/etc/services"` als auch in `"inetd.conf"` eine weitere Zeile hinzufügen.

Beispiel:

```
MQSeries1    1414/tcp  
MQSeries2    1822/tcp
```

```
MQSeries1 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1  
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

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 im Abschnitt [Rückstandsoption des TCP-Empfangsprogramms verwenden](#).

Der InetD-Prozess kann unter Linux die Rate der an einem TCP-Anschluss eingehenden Verbindungen einschränken. Die Standardrate sind 40 Verbindungen mit einem Intervall von 60 Sekunden. Wenn Sie eine höhere Rate benötigen, geben Sie eine neue Beschränkung für die Anzahl der eingehenden Verbindungen in einem 60-Sekunden-Intervall an, indem Sie einen Punkt (.) gefolgt von dem neuen Grenzwert

an den Parameter 'nowait' des betreffenden Service in 'inetd.conf' anhängen. Geben Sie beispielsweise für einen Grenzwert von 500 Verbindungen in einem 60-Sekunden-Intervall Folgendes ein:

```
MQSeries stream tcp nowait.500 mqm / MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
```

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

## Den erweiterten inet-Dämon (XINETD) verwenden

Die folgenden Anweisungen beschreiben, wie der erweiterte inet-Dämon in Red Hat Linux-Systeme implementiert wird. Wenn Sie eine andere Linux-Distribution verwenden, müssen diese Anweisungen gegebenenfalls entsprechend angepasst werden.

Gehen Sie wie folgt vor, um eine TCP-Verbindung aufzubauen.

1. Bearbeiten Sie die Datei "/etc/services". Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries 1414/tcp # MQSeries channel listener
```

**Anmerkung:** Zur Bearbeitung dieser Datei müssen Sie als Superuser oder Rootbenutzer angemeldet sein.

2. Erstellen Sie im XINETD-Konfigurationsverzeichnis "/etc/xinetd.d" eine Datei mit dem Namen IBM MQ. Fügen Sie die folgende Zeilengruppe zur Datei hinzu:

```
# IBM MQ service for XINETD
service MQSeries
{
  disable          = no
  flags            = REUSE
  socket_type     = stream
  wait            = no
  user            = mqm
  server          = MQ_INSTALLATION_PATH/bin/amqcrsta
  server_args     = -m queue.manager.name
  log_on_failure += USERID
}
```

3. Starten Sie den erweiterten inet-Dämon erneut, indem Sie den folgenden Befehl eingeben:

```
/etc/rc.d/init.d/xinetd restart
```

Wenn Sie mehrere Warteschlangenmanager in Ihrem System betreiben und demzufolge mehr als einen Service benötigen, müssen Sie für jeden einzelnen Warteschlangenmanager in der Datei '/etc/services' eine weitere Zeile hinzufügen. Sie können für die einzelnen Services im Verzeichnis '/etc/xinetd.d' eine Datei erstellen. Alternativ können Sie der zuvor erstellten IBM MQ-Datei weitere Zeilengruppen hinzufügen.

Der xinetd-Prozess kann unter Linux die Rate der an einem TCP-Anschluss eingehenden Verbindungen einschränken. Der Standardwert beträgt 50 Verbindungen in einem 10-Sekunden-Intervall. Wenn Sie eine höhere Rate benötigen, geben Sie für den Rate der eingehenden Verbindungen einen Grenzwert an, indem Sie in der xinetd-Konfigurationsdatei das Attribut "cps" angeben. Geben Sie beispielsweise für einen Grenzwert von 500 Verbindungen in einem 60-Sekunden-Intervall Folgendes ein:

```
cps = 500 60
```

## Wie geht es weiter?

Die TCP/IP-Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Fahren Sie dazu mit „IBM MQ for Linux-Konfiguration“ auf Seite 44 fort.

## Linux IBM MQ for Linux-Konfiguration

Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass die Benutzer-ID und die Gruppe "mqm" erstellt und das Kennwort festgelegt wurden.

Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

### Anmerkung:

1. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert, wobei `MQ_INSTALLATION_PATH` das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
2. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
3. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

## Basiskonfiguration

1. Erstellen Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q linux
```

Dabei gilt:

### **linux**

ist der Name des Warteschlangenmanagers

### **-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

### **-u dlqname**

gibt den Namen der Warteschlange für nicht zustellbare Nachrichten an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
strmqm linux
```

, hierbei ist `linux` der Name, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

## Linux Kanalkonfiguration für Linux

Im folgenden Abschnitt finden Sie ausführliche Informationen zu der Konfiguration des Linux-Warteschlangenmanagers für die Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals.

Es ist der MQSC-Befehl zum Erstellen der einzelnen Objekte angegeben. Entweder starten Sie **runmqsc** in einer UNIX-Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for Linux und IBM MQ for HP-UX bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werte im Beispiel für HP-UX ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern,

müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

<i>Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Linux</i>				
<b>ID</b>	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>LINUX</b>	
B	Name der lokalen Warteschlange		<b>LINUX.LOCALQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Windows“ auf Seite 58 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (SNA)		<b>LINUX.WINNT.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>LINUX.WINNT.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.LINUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.LINUX.TCP</b>	
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für AIX“ auf Seite 11 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (SNA)		<b>LINUX.AIX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>LINUX.AIX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AIX.LINUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.LINUX.TCP</b>	
<b>Verbindung zu MQSeries for Compaq Tru64 UNIX</b>				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem Compaq Tru64 UNIX-System entsprechen.				

Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Linux (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.LINUX.TCP</b>	
J	Name des Empfängerkanals (TCP)	H	<b>LINUX.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP-UX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 2 auf Seite 19</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>HPUX</b>	
D	Name der fernen Warteschlange		<b>HPUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>HPUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>HPUX</b>	
G	Name des Senderkanals (SNA)		<b>LINUX.HPUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>LINUX.HPUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>HPUX.LINUX.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>HPUX.LINUX.TCP</b>	
<b>Verbindung zu IBM MQ for Solaris</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 6 auf Seite 51</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>SOLARIS</b>	
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>GIS</b>	
G	Name des Senderkanals (SNA)		<b>LINUX.SOLARIS.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>LINUX.SOLARIS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.LINUX.SNA</b>	

Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Linux (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.LINUX.TCP</b>	
<p><b>IBM i</b> <b>Verbindung zu IBM MQ for IBM i</b></p> <p><b>IBM i</b> Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 4 auf Seite 37</a> angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	
G	Name des Senderkanals (SNA)		<b>LINUX.AS400.SNA</b>	
H	Name des Senderkanals (TCP)		<b>LINUX.AS400.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AS400.LINUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AS400.LINUX.TCP</b>	
<p><b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS</b></p> <p><b>z/OS</b> Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 8 auf Seite 65</a> angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>LINUX.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>LINUX.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.LINUX.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>MVS.LINUX.TCP</b>	

Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Linux (Forts.)				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Verbindung zu MQSeries für VSE/ESA</b> (nur IBM MQ for Linux (x86 -Plattform))				
Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>LINUX.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.LINUX.SNA</b>	

**Linux** Senderkanaldefinitionen unter Verwendung der SNA in IBM MQ for Linux (x86 -Plattform)  
Mustercode:

```
def q1 (HPUX) + usage(xmitq) + replace F
def qr (HPUX.REMOTEQ) + rname(HPUX.LOCALQ) + rqmname(HPUX) + xmitq(HPUX) + replace D
E
C
F
def chl (LINUX.HPUX.SNA) chltype(sdr) + trptype(lu62) + conname('HPUXCPIC') + xmitq(HPUX) + replace G
14
F
```

**Linux** Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for Linux (x86 -Plattform)  
Mustercode:

```
def q1 (LINUX.LOCALQ) replace B
def chl (HPUX.LINUX.SNA) chltype(rcvr) + trptype(lu62) + replace I
```

**Linux** Senderkanaldefinitionen über TCP in IBM MQ for Linux  
Mustercode:

```
def q1 (HPUX) + usage(xmitq) + replace F
def qr (HPUX.REMOTEQ) + rname(HPUX.LOCALQ) + rqmname(HPUX) + xmitq(HPUX) + replace D
E
C
F
```



```

replace

def chl (LINUX.HPUX.TCP) chltype(sdr) +           H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(HPUX) +                                   F
  replace

```

## Linux Empfängerkanaldefinitionen über TCP/IP in IBM MQ for Linux

Mustercode:

```

def ql (LINUX.LOCALQ) replace                       B

def chl (HPUX.LINUX.TCP) chltype(rcvr) +          J
  trptype(tcp) +
  replace

```

## Solaris MQ-Beispielkonfiguration für Solaris

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for Solaris- und IBM MQ-Produkten.

Für folgende Plattformen stehen Beispiele zur Verfügung:

- Windows
- AIX
- HP Tru64 UNIX
- HP-UX
- Linux
- IBM i
- z/OS
- VSE/ESA

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## Solaris LU 6.2-Verbindung mit SNAP-IX aufbauen

Parameter für die Konfiguration einer LU 6.2-Verbindung mit SNAP-IX.

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP finden Sie in der IBM Onlinedokumentation unter [Communications Server](#) sowie in der MetaSwitch-Onlinedokumentation [SNAP-IX Administration Guide](#) und in der Sun-Onlinedokumentation [Configuring Intersystem Communications \(ISC\)](#).

## Solaris Eine TCP-Verbindung einrichten

Informationen zur Konfiguration einer TCP-Verbindung und weitere Schritte.

Gehen Sie wie folgt vor, um eine TCP-Verbindung aufzubauen.

1. Bearbeiten Sie die Datei "/etc/services".

**Anmerkung:** Hierzu müssen Sie sich als Superuser oder Root-Benutzer anmelden. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. 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]
```

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

3. Ermitteln Sie die Prozess-ID von InetD durch Eingabe folgenden Befehls:

```
ps -ef | grep inetd
```

4. Führen Sie den entsprechenden Befehl aus:

- Für Solaris 9:

```
kill -1 inetd processid
```

- Für Solaris 10 oder höher:

```
inetconv
```

## Wie geht es weiter?

Die TCP/IP-Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Fahren Sie dazu mit „[Konfiguration von IBM MQ for Solaris](#)“ auf Seite 50 fort.

## **Solaris** Konfiguration von IBM MQ for Solaris

Beschreibt die zum Abschließen der Konfiguration zu definierenden Kanäle.

Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass der Benutzer und die Gruppe `mqm` erstellt und das Kennwort festgelegt wurden.

Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

### Anmerkung:

1. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert.  
`MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.
2. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
3. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.
4. Wenn bei einem SNA- oder LU 6.2-Kanal bei dem Versuch, die Kommunikationsbibliothek zu laden, ein Fehler auftritt, ist eventuell die Datei "liblu62.so" nicht auffindbar. Dieses Problem lässt sich möglicherweise durch Hinzufügen der Speicherposition dieser Datei, meist ist dies das Verzeichnis `"/opt/SUNWlu62"`, zu `LD_LIBRARY_PATH` lösen.

## Basiskonfiguration

1. Erstellen Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q solaris
```

Dabei gilt:

## **solaris**

ist der Name des Warteschlangenmanagers

### **-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

### **-u dlqname**

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
strmqm solaris
```

, wobei *solaris* der Name ist, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

## **Solaris Kanalkonfiguration für Solaris**

Im folgenden Abschnitt finden Sie ausführliche Informationen zur Konfiguration des Solaris-Warteschlangenmanagers für die Implementierung eines Kanals.

Die beschriebene Konfiguration dient zur Implementierung des in [Abbildung 1 auf Seite 5](#) beschriebenen Kanals.

Es ist der MQSC-Befehl zum Erstellen der einzelnen Objekte angegeben. Entweder starten Sie **runmqsc** in einer UNIX-Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for Solaris und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern, müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

Tabelle 6. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>SOLARIS</b>	
B	Name der lokalen Warteschlange		<b>SOLARIS.LOCALQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 7 auf Seite 59</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	

Tabelle 6. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
G	Name des Senderkanals (SNA)		<b>SOLARIS.WINNT.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>SOLARIS.WINNT.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.SOLARIS.TCP</b>	
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 1</a> auf Seite 12 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.AIX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.AIX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AIX.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.SOLARIS.TCP</b>	
<b>Verbindung zu MQSeries for Compaq Tru64 Unix</b>				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem Compaq Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.SOLARIS.TCP</b>	
J	Name des Empfängerkanals (TCP)	H	<b>SOLARIS.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP-UX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 2</a> auf Seite 19 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>HPUX</b>	
D	Name der fernen Warteschlange		<b>HPUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>HPUX.LOCALQ</b>	

Tabelle 6. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
F	Name der Übertragungswarteschlange		<b>HPUX</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.HPUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.HPUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>HPUX.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>HPUX.SOLARIS.TCP</b>	

**Verbindung zu IBM MQ for Linux**

Die Werte in diesem Abschnitt der Tabelle müssen den in [Tabelle 5 auf Seite 45](#) angegebenen entsprechen.

C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.LINUX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>SOLARIS.LINUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.SOLARIS.TCP</b>	

**IBM i Verbindung zu IBM MQ for IBM i**

**IBM i** Die Werte in diesem Abschnitt der Tabelle müssen den in [Tabelle 4 auf Seite 37](#) angegebenen entsprechen.

C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.AS400.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.AS400.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AS400.SOLARIS.SNA</b>	

Tabelle 6. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
IBM i IBM i J	Name des Empfängerkanals (TCP)	H	<b>AS400.SOLARIS.TCP</b>	
<b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS</b>				
<b>z/OS</b> Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 8 auf Seite 65</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.SOLARIS.SNA</b>	
<b>z/OS</b> <b>z/OS</b> J	Name des Empfängerkanals (TCP)	H	<b>MVS.SOLARIS.TCP</b>	
<b>Verbindung zu MQSeries for VSE/ESA</b>				
Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>SOLARIS.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.SOLARIS.SNA</b>	

**Solaris** *Senderkanaldefinitionen unter Verwendung der SNAP-IX SNA in IBM MQ for Solaris*

Mustercode:

```
def q1 (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqnname(WINNT) +                             C
```

```

xmitq(WINNT) +          F
replace

def chl (SOLARIS.WINNT.SNA) chltype(sdr) +      G
  trptype(lu62) +
  conname('NTCPIC') +          14
  xmitq(WINNT) +          F
  replace

```

### **Solaris** Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for Solaris

Mustercode:

```

def ql (SOLARIS.LOCALQ) replace                B

def chl (WINNT.SOLARIS.SNA) chltype(rcvr) +    I
  trptype(lu62) +
  replace

```

### **Solaris** Senderkanaldefinitionen über TCP in IBM MQ for Solaris

Mustercode:

```

def ql (WINNT) +          F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +  D
  rname(WINNT.LOCALQ) +  E
  rqmname(WINNT) +       C
  xmitq(WINNT) +         F
  replace

def chl (SOLARIS.WINNT.TCP) chltype(sdr) +    H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(WINNT) +         F
  replace

```

### **Solaris** Empfängerkanaldefinitionen über TCP/IP in IBM MQ for Solaris

Mustercode:

```

def ql (SOLARIS.LOCALQ) replace                B

def chl (WINNT.SOLARIS.TCP) chltype(rcvr) +    J
  trptype(tcp) +
  replace

```

## **Windows** IBM MQ-Beispielkonfiguration für Windows

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for Windows- und IBM MQ-Produkten auf anderen Plattformen.

Die Einrichtung von Kommunikationsverbindungen wird für folgende Plattformen erläutert:

- AIX
- HP Tru64 UNIX
- HP-UX
- Solaris
- Linux
- IBM i
- z/OS
- VSE/ESA

Sobald die Verbindung eingerichtet ist, müssen Sie einige Kanäle definieren, um die Konfiguration abzuschließen. Im Abschnitt „[Konfiguration von IBM MQ for Windows](#)“ auf Seite 57 finden Sie eine Beschreibung der Beispielprogramme und Befehle.

Unter „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## Windows Eine LU 6.2-Verbindung einrichten

Verweis auf Informationen zum Konfigurieren von AnyNet SNA über TCP/IP.

Aktuelle Informationen zur Konfiguration von AnyNet SNA über TCP/IP finden Sie online in der folgenden IBM Dokumentation: [AnyNet SNA over TCP/IP](#), [SNA Node Operations](#), and [Communications Server for Windows](#)

## Windows Eine TCP-Verbindung einrichten

Der TCP-Stapel, der mit Windows-Systemen ausgeliefert wird, umfasst keinen *inet*- bzw. funktional entsprechenden Dämon.

Der IBM MQ-Befehl, über den das IBM MQ for TCP-Empfangsprogramm gestartet wird, lautet:

```
runmqldr -t tcp
```

Das Empfangsprogramm muss unbedingt vor jeglichen Kanälen gestartet werden. Es ermöglicht empfangenden Kanälen das automatische Starten nach dem Erhalt einer Anforderung von einem Senderkanal für eingehende Nachrichten.

### Wie geht es weiter?

Sobald die TCP/IP-Verbindung besteht, können Sie die Konfiguration abschließen. Fahren Sie dazu mit „[Konfiguration von IBM MQ for Windows](#)“ auf Seite 57 fort.

## Windows NetBIOS-Verbindung aufbauen

Eine NetBIOS-Verbindung wird aus einem Warteschlangenmanager initialisiert, der den Parameter "ConnectionName" für seine Kanaldefinition verwendet, um eine Verbindung zu einem Zielempfangsprogramm herzustellen.

Gehen Sie wie folgt vor, um eine NetBIOS-Verbindung einzurichten:

1. Geben Sie an beiden Enden des Kanals in der Konfigurationsdatei des Warteschlangenmanagers "qm.ini" den lokalen NetBIOS-Namen an, den die IBM MQ-Kanalprozesse verwenden sollen. Die NETBIOS-Zeilengruppe in Windows kann an der Sendeseite wie folgt aussehen:

```
NETBIOS:  
LocalName=WNTNETB1
```

und an der Empfangsseite:

```
NETBIOS:  
LocalName=WNTNETB2
```

Die IBM MQ-Prozesse müssen unterschiedliche lokale NetBIOS-Namen verwenden. Verwenden Sie nicht den Systemnamen als NetBIOS-Namen, da dieser bereits von Windows verwendet wird.

2. Überprüfen Sie an beiden Kanalenden die auf Ihrem System verwendete LAN-Adaptornummer. Die Standardeinstellung von IBM MQ for Windows für die logische Adaptornummer 0 ist, dass NetBIOS über ein IP-Netz ausgeführt wird. Um das systemeigene NetBIOS zu verwenden, müssen Sie die Nummer des logischen Adapters auf 1 setzen. Informationen hierzu finden Sie im Abschnitt [LAN-Adaptornummer erstellen](#).



Geben Sie die richtige LAN-Adaptornummer in der NETBIOS-Zeilengruppe der Windows-Registrierungsdatenbank an. Beispiel:

```
NETBIOS:  
AdapterNum=1
```

3. Damit der Senderkanal einwandfrei gestartet werden kann, geben Sie den lokalen NetBIOS-Namen in Form der MQNAME-Umgebungsvariablen an:

```
SET MQNAME=WNTNETB1I
```

Dieser Name muss eindeutig sein.

4. Definieren Sie an der Sendeseite einen Kanal mit dem von der anderen Kanalseite verwendeten NetBIOS-Namen. Beispiel:

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(SDR) +  
  TRPTYPE(NETBIOS) +  
  CONNAME(WNTNETB2) +  
  XMITQ(OS2) +  
  MCATYPE(THREAD) +  
  REPLACE
```

Sie müssen die Option MCATYPE(THREAD) angeben. Unter Windows müssen Senderkanäle als Threads ausgeführt werden.

5. Definieren Sie an der Empfangsseite den zugehörigen Empfängerkanal. Beispiel:

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(RCVR) +  
  TRPTYPE(NETBIOS) +  
  REPLACE
```

6. Starten Sie den Kanalinitiator, da jeder neue Kanal als Thread und nicht als neuer Prozess gestartet wird.

```
runmqchi
```

7. Starten Sie auf der Empfangsseite das Empfangsprogramm von IBM MQ:

```
runmqclsr -t netbios
```

Sie haben die Möglichkeit, Werte für den Warteschlangenmanagernamen, den lokalen NetBIOS-Namen, die Anzahl der Sitzungen, die Anzahl der Namen und die Anzahl der Befehle anzugeben. Weitere Informationen zum Einrichten der NetBIOS-Verbindungen finden Sie im Abschnitt [NetBIOS-Verbindung unter Windows einrichten](#).

## Konfiguration von IBM MQ for Windows

Beispielprogramme und -befehle für die Konfiguration.

### Anmerkung:

1. Mithilfe des Beispielprogramms AMQSBCG können Sie die Inhalte und Header aller Nachrichten in einer Warteschlange anzeigen. Beispiel:

```
AMQSBCG q_name qmgr_name
```

zeigt die Inhalte der Warteschlange *q\_name* an, die im Warteschlangenmanager *qmgr\_name* definiert ist.

Alternativ können Sie den Nachrichten-Browser im IBM MQ Explorer benutzen.

- Über die Eingabeaufforderung kann jeder Kanal durch folgenden Befehl gestartet werden:

```
runmqchl -c channel.name
```

- Fehlerprotokolle finden Sie in den Verzeichnissen `MQ_INSTALLATION_PATH\qmgrs\qmgrname\errors` und `MQ_INSTALLATION_PATH\qmgrs\@system\errors`. In beiden Verzeichnissen befinden sich die neuesten Nachrichten am Ende der Datei "amqerr01.log".

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

- Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

### Windows **Standardkonfiguration**

Die PostCard-Anwendung von IBM MQ führt Sie durch die Erstellung von Standardkonfigurationen.

Informationen zur Verwendung der PostCard-Anwendung finden Sie im Abschnitt *Installation mithilfe der PostCard-Anwendung prüfen* für die in Ihrem Unternehmen verwendete Plattform.

### Windows **Basiskonfiguration**

Warteschlangenmanager können in IBM MQ Explorer oder über die Eingabeaufforderung erstellt und gestartet werden.

Bei Verwendung der Eingabeaufforderung:

- Geben Sie zur Erstellung des Warteschlangenmanagers folgenden Befehl ein:

```
crtmqm -u dlqname -q winnt
```

Dabei gilt:

**winnt**

ist der Name des Warteschlangenmanagers

**-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

**-u dlqname**

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

- Starten Sie den Warteschlangenmanager über den folgenden Befehl:

```
strmqm winnt
```

, wobei *winnt* der Name ist, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

### Windows **Kanalkonfiguration für Windows**

Beispielkonfiguration, die für den Windows-Warteschlangenmanager vorgenommen werden muss, um einen angegebenen Kanal zu implementieren.

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte für den Windows-Warteschlangenmanager zur Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Es ist jeweils der MQSC-Befehl angegeben. Entweder starten Sie **runmqsc** aus einer Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for Windows und IBM MQ for AIX bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern, müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

<i>Tabelle 7. Arbeitsblatt zur Konfiguration für IBM MQ for Windows</i>				
	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>WINNT</b>	
B	Name der lokalen Warteschlange		<b>WINNT.LOCALQ</b>	
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für AIX“ auf Seite 11 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.AIX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.AIX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AIX.WINNT.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.WINNT.TCP</b>	
<b>Verbindung zu IBM MQ for HP Tru64 UNIX</b>				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem HP Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.WINNT.TCP</b>	
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP-UX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für HP-UX“ auf Seite 19 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>HPUX</b>	
D	Name der fernen Warteschlange		<b>HPUX.REMOTEQ</b>	

Tabelle 7. Arbeitsblatt zur Konfiguration für IBM MQ for Windows (Forts.)

	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
E	Name der Warteschlange auf fernem System	B	<b>HPUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>HPUX</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.HPUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.HPUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>HPUX.WINNT.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>HPUX.WINNT.TCP</b>	
<b>Verbindung zu IBM MQ for Solaris</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 51 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>SOLARIS</b>	
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>SOLARIS</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.SOLARIS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.SOLARIS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.WINNT.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>SOLARIS.WINNT.TCP</b>	
<b>Verbindung zu IBM MQ for Linux</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 44 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.LINUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.LINUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.WINNT.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>LINUX.WINNT.TCP</b>	

Tabelle 7. Arbeitsblatt zur Konfiguration für IBM MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<p><b>IBM i</b> <b>Verbindung zu IBM MQ for IBM i</b></p> <p><b>IBM i</b> Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für IBM i“ auf Seite 36 angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.AS400.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.AS400.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AS400.WINNT.SNA</b>	
<b>IBM i</b> <b>IBM i</b> J	Name des Empfängerkanals (TCP)	H	<b>AS400.WINNT.TCP</b>	
<p><b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS</b></p> <p><b>z/OS</b> Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für z/OS“ auf Seite 65 angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.WINNT.SNA</b>	
<b>z/OS</b> <b>z/OS</b> J	Name des Empfängerkanals (TCP/IP)	H	<b>MVS.WINNT.TCP</b>	
<p><b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange</b></p> <p><b>z/OS</b> Die Werte in diesem Abschnitt der Tabelle müssen den in „Beispiel für die Konfiguration gemeinsamer Kanäle“ auf Seite 74 angegebenen entsprechen.</p>				
C	Name des fernen Warteschlangenmanagers	A	<b>QSG</b>	

Tabelle 7. Arbeitsblatt zur Konfiguration für IBM MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
D	Name der fernen Warteschlange		<b>QSG.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>QSG.SHAREDQ</b>	
F	Name der Übertragungswarteschlange		<b>QSG</b>	
G	Name des Senderkanals (SNA)		<b>WINNT.QSG.SNA</b>	
H	Name des Senderkanals (TCP)		<b>WINNT.QSG.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>QSG.WINNT.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>QSG.WINNT.TCP</b>	
<b>Verbindung zu MQSeries for VSE/ESA</b>				
Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>WINNT.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.WINNT.SNA</b>	

**Windows** Senderkanaldefinitionen unter Verwendung der SNA in IBM MQ for Windows

Codebeispiel:

```
def ql (AIX) + usage(xmitq) + replace F
def qr (AIX.REMOTEQ) + rname(AIX.LOCALQ) + rqmname(AIX) + xmitq(AIX) + replace D
E
C
F
def chl (WINNT.AIX.SNA) chltype(sdr) + trptype(lu62) + conname(AIXCPIC) + xmitq(AIX) + replace G
18
F
```

**Windows** Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for Windows

Codebeispiel:

```
def ql (WINNT.LOCALQ) replace B
def chl (AIX.WINNT.SNA) chltype(rcvr) + I
```

```
trptype(lu62) +  
replace
```

### Windows *Senderkanaldefinitionen über TCP/IP in IBM MQ for Windows*

Codebeispiel:

```
def ql (AIX) +                               F  
  usage(xmitq) +  
  replace  
  
def qr (AIX.REMOTEQ) +                       D  
  rname(AIX.LOCALQ) +                       E  
  rqmname(AIX) +                            C  
  xmitq(AIX) +                               F  
  replace  
  
def chl (WINNT.AIX.TCP) chltype(sdr) +      H  
  trptype(tcp) +  
  conname(remote_tcpip_hostname) +  
  xmitq(AIX) +                               F  
  replace
```

### Windows *Empfängerkanaldefinitionen unter Verwendung von TCP in IBM MQ for Windows*

Codebeispiel:

```
def ql (WINNT.LOCALQ) replace                B  
  
def chl (AIX.WINNT.TCP) chltype(rcvr) +     J  
  trptype(tcp) +  
  replace
```

### Windows **Automatischer Start**

In IBM MQ for Windows haben Sie die Möglichkeit, den Start eines Warteschlangenmanagers und dessen Kanalinitiatoren, Kanälen, Empfangsprogrammen und Befehlsservern zu automatisieren.

Verwenden Sie das Snap-in "IBM MQ Services", um die Services für den Warteschlangenmanager zu definieren. Wenn Sie die Tests der Kommunikationskonfiguration erfolgreich abgeschlossen haben, setzen Sie die relevanten Services im Snap-in auf **automatisch**. Diese Datei kann beim Systemstart vom mitgelieferten IBM MQ-Service gelesen werden.

Weitere Informationen finden Sie im Abschnitt [IBM MQ verwalten](#).

### Windows **Kanäle als Prozesse oder Threads ausführen**

IBM MQ for Windows bietet die Flexibilität, sendende Kanäle als Windows-Prozesse oder Windows-Threads auszuführen. Dies wird im Parameter MCTYPE der Senderkanaldefinition festgelegt.

In den meisten Installationen werden die sendenden Kanäle als Threads ausgeführt, da auf diese Weise die zur Unterstützung vieler gleichzeitiger Kanalverbindungen benötigte Größe des virtuellen Speichers und Realspeicher reduziert werden kann. Für eine NetBIOS-Verbindung ist jedoch ein eigener Prozess für den sendenden Message Channel Agent erforderlich.

## MQ-Beispielkonfiguration für z/OS

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ für z/OS- und IBM MQ-Produkten auf anderen Plattformen.

Dieses Beispiel gilt auch für folgende andere Plattformen:

- Windows
- AIX
- Compaq Tru64 UNIX
- HP-UX

- Solaris
- Linux
- IBM i
- VSE/ESA

Außerdem sind folgende Verbindungen möglich:

- z/OS auf z/OS
- z/OS zu MVS
- MVS zu MVS

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## **Verbindung aufbauen**

Zum Aufbau einer Verbindung müssen eine Reihe von Einstellungen konfiguriert werden.

### **Eine LU 6.2-Verbindung einrichten**

Aktuelle Informationen zum Konfigurieren von SNA über TCP/IP finden Sie in der folgenden IBM Online-dokumentation: [Communications Server for z/OS](#).

### **Eine TCP-Verbindung einrichten**

Damit die richtigen Parameter für die Steuerung verteilter Warteschlangen verwendet werden, modifizieren Sie das Warteschlangenmanagerobjekt durch folgenden Befehl. Sie müssen dem Warteschlangenmanagerattribut TCPNAME den Namen des TCP-Adressraums hinzufügen.

```
ALTER QMGR TCPNAME(TCPIP)
```

Die TCP-Verbindung ist nun eingerichtet. Sie können die Konfiguration abschließen.

## **Konfiguration von IBM MQ for z/OS**

Die folgenden Schritte dienen zur Konfiguration von IBM MQ sowie zum Starten und Konfigurieren von Kanälen und Empfangsprogrammen.

1. Starten Sie den Kanalinitiator durch folgenden Befehl:

```
/cpf START CHINIT 1
```

2. Starten Sie ein LU 6.2-Empfangsprogramm mithilfe des folgenden Befehls:

```
/cpf START LSTR LUNAME( M1 ) TRPTYPE(LU62)
```

LUNAME von M1 bezieht sich auf den symbolischen Namen, den Sie Ihrer logischen Einheit (5) gegeben haben. Sie müssen TRPTYPE(LU62) angeben, andernfalls geht das Empfangsprogramm davon aus, dass Sie TCP verwenden möchten.

3. Starten Sie ein TCP-Empfangsprogramm mithilfe des folgenden Befehls:

```
/cpf START LSTR
```



Wenn Sie einen anderen Anschluss als 1414 (Standardanschluss in IBM MQ) nutzen möchten, geben Sie folgenden Befehl ein:

```
/cpf START LSTR PORT( 1555 )
```

IBM MQ-Kanäle werden nicht erfolgreich initialisiert, wenn bei der Kanalfestlegung erkannt wird, dass die Nachrichtenfolgennummern an den beiden Enden unterschiedlich sind. Unter Umständen müssen Sie diese Kanäle manuell zurücksetzen.

### **Kanalkonfiguration für z/OS**

Zur Implementierung der Beispielkanäle sind einige Konfigurationsschritte auf dem z/OS-Warteschlangenmanager erforderlich.

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte für den z/OS-Warteschlangenmanager zur Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for z/OS und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Werte im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern, müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

<i>Tabelle 8. Arbeitsblatt zur Konfiguration für IBM MQ for z/OS</i>				
<b>ID</b>	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>MVS</b>	
B	Name der lokalen Warteschlange		<b>MVS.LOCALQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für Windows</a> “ auf Seite 58 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (LU 6.2)		<b>MVS.WINNT.SNA</b>	
H	Name des Senderkanals (TCP)		<b>MVS.WINNT.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>WINNT.MVS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>WINNT.MVS.TCP</b>	
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für AIX</a> “ auf Seite 11 verwendeten Werten wie angegeben entsprechen.				

Tabelle 8. Arbeitsblatt zur Konfiguration für IBM MQ for z/OS (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (LU 6.2)		<b>MVS.AIX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>MVS.AIX.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>AIX.MVS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>AIX.MVS.TCP</b>	
<b>Verbindung zu MQSeries for Compaq Tru64 Unix</b>				
Die Werte in diesem Tabellenabschnitt müssen den Werten in Ihrem Compaq Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>DECUX</b>	
D	Name der fernen Warteschlange		<b>DECUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>DECUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>DECUX</b>	
H	Name des Senderkanals (TCP)		<b>DECUX.MVS.TCP</b>	
J	Name des Empfängerkanals (TCP)	H	<b>MVS.DECUX.TCP</b>	
<b>Verbindung zu IBM MQ for HP-UX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für HP-UX“ auf Seite 19 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>HPUX</b>	
D	Name der fernen Warteschlange		<b>HPUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>HPUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>HPUX</b>	
G	Name des Senderkanals (LU 6.2)		<b>MVS.HPUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>MVS.HPUX.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>HPUX.MVS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>HPUX.MVS.TCP</b>	

Tabelle 8. Arbeitsblatt zur Konfiguration für IBM MQ for z/OS (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Verbindung zu IBM MQ for Solaris</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 51 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>SOLARIS</b>	
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>SOLARIS</b>	
G	Name des Senderkanals (LU 6.2)		<b>MVS.SOLARIS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>MVS.SOLARIS.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>SOLARIS.MVS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.MVS.TCP</b>	
<b>Verbindung zu IBM MQ for Linux</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 44 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (LU 6.2)		<b>MVS.LINUX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>MVS.LINUX.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>LINUX.MVS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.MVS.TCP</b>	
<b>Verbindung zu IBM MQ for IBM i</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für IBM i“ auf Seite 36 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	

Tabelle 8. Arbeitsblatt zur Konfiguration für IBM MQ for z/OS (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
G	Name des Senderkanals (LU 6.2)		<b>MVS.AS400.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>MVS.AS400.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>AS400.MVS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>AS400.MVS.TCP</b>	
<b>Verbindung zu MQSeries for VSE/ESA</b>				
Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>	
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>VSE</b>	
G	Name des Senderkanals		<b>MVS.VSE.SNA</b>	
I	Name des Empfängerkanals	G	<b>VSE.MVS.SNA</b>	

### Senderkanaldefinitionen von IBM MQ for z/OS

In diesem Abschnitt werden die zur Konfiguration von IBM MQ for z/OS unter Verwendung von LU 6.2 oder TCP erforderlichen Senderkanaldefinitionen ausführlich beschrieben.

Für LU 6.2:

```

Local Queue
  Object type : QLOCAL
  Name       : WINNT
  Usage     : X (XmitQ)
                                     F

Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ
  Name on remote system : WINNT.LOCALQ
  Remote system name : WINNT
  Transmission queue : WINNT
                                     D
                                     E
                                     C
                                     F

Sender Channel
  Channel name : MVS.WINNT.SNA
  Transport type : L (LU6.2)
  Transmission queue name : WINNT
  Connection name : M3
                                     G
                                     F
                                     13
    
```

Für TCP:

```

Local Queue
  Object type : QLOCAL
  Name       : WINNT
  Usage     : X (XmitQ)
                                     F

Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ
  Name on remote system : WINNT.LOCALQ
  Remote system name : WINNT
  Transmission queue : WINNT
                                     D
                                     E
                                     C
                                     F

Sender Channel
    
```

```
Channel name : MVS.WINNT.TCP      H
Transport type : T (TCP)
Transmission queue name : WINNT      F
Connection name : winnt.tcpip.hostname
```

## Empfängerkanaldefinitionen von IBM MQ for z/OS

In diesem Abschnitt werden die zur Konfiguration von IBM MQ for z/OS unter Verwendung von LU 6.2 oder TCP erforderlichen Empfängerkanaldefinitionen ausführlich beschrieben.

Für LU 6.2:

```
Local Queue
  Object type : QLOCAL
  Name : MVS.LOCALQ      B
  Usage : N (Normal)

Receiver Channel
  Channel name : WINNT.MVS.SNA      I
```

Für TCP:

```
Local Queue
  Object type : QLOCAL
  Name : MVS.LOCALQ      B
  Usage : N (Normal)

Receiver Channel
  Channel name : WINNT.MVS.TCP      J
```

z/OS

## MQ-Beispielkonfiguration für z/OS unter Verwendung von QSGs

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zu einer Gruppe mit gemeinsamer Warteschlange von IBM MQ-Produkten unter Windows und AIX. Sie können auch Verbindungen von z/OS zu z/OS herstellen.

Die Einrichtung von Kommunikationsverbindungen zwischen einer Gruppe mit gemeinsamer Warteschlange und einer anderen Plattform als z/OS ist mit dem Vorgang identisch, der unter „MQ-Beispielkonfiguration für z/OS“ auf Seite 63 beschrieben wird. Dieser Abschnitt enthält auch Beispiele anderer Plattformen.

Sobald die Verbindung eingerichtet ist, müssen Sie einige Kanäle definieren, um die Konfiguration abzuschließen. Dieser Vorgang wird im Abschnitt „Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS“ auf Seite 74 beschrieben.

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

z/OS

## Konfigurationsparameter für eine LU 6.2-Verbindung

Im folgenden Arbeitsblatt sind alle Parameter aufgelistet, die zum Einrichten der Kommunikation von einem z/OS-System zu einer der anderen IBM MQ-Plattformen erforderlich sind. Das Arbeitsblatt enthält Beispiele der Parameter, die in einer Arbeitsumgebung getestet wurden, und ermöglicht Ihnen, Ihre eigenen Werte einzutragen.

Verwenden Sie das Arbeitsblatt in diesem Abschnitt mit dem Arbeitsblatt im Abschnitt für die Plattformen, mit denen Sie eine Verbindung herstellen.

Die zum Einrichten einer LU 6.2-Verbindung erforderlichen Schritte sind in „LU 6.2-Verbindung in einer Gruppe mit gemeinsamer Warteschlange aufbauen“ auf Seite 71 beschrieben, darin sind nummerierte Querverweise zu den Parametern auf dem Arbeitsblatt enthalten.

Die Zahlen in der Referenzspalte geben an, dass der Wert mit dem auf dem entsprechenden Arbeitsblatt in diesem Abschnitt übereinstimmen muss. Die nachfolgenden Beispiele in diesem Abschnitt beziehen

sich auf die Werte in der Spalte mit der ID. Die Einträge in der Spalte mit dem Parameternamen werden in „Begriffserklärungen“ auf Seite 70 erläutert.

<i>Tabelle 9. Arbeitsblatt zur Konfiguration für z/OS mithilfe von LU 6.2</i>				
<b>ID</b>	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten mithilfe generischer Ressourcen</b>				
1	Befehlspräfix		/cpf	
2	Netz-ID		NETID	
3	Node		MVSPU	
6	Modename		#INTER	
7	Name des lokalen Transaktionsprogramms		MQSERIES	
8	LAN-Zieladresse		400074511092	
9	Name der lokalen logischen Einheit		MVSLU1	
10	Name der generischen Ressource		MVSGR	
11	Symbolisches Ziel		G1	
12	Symbolisches Ziel für den Namen der generischen Ressource		G2	
<b>Verbindung mit einem Windows-System</b>				
13	Symbolisches Ziel		M3	
14	Modename	21	#INTER	
15	Name des fernen Transaktionsprogramms	7	MQSERIES	
16	Name der Partner-LU	5	WINNTLU	
21	ID des fernen Knotens	4	05D 30F65	
<b>Verbindung mit einem AIX-System</b>				
13	Symbolisches Ziel		M4	
14	Modename	18	#INTER	
15	Name des fernen Transaktionsprogramms	6	MQSERIES	
16	Name der Partner-LU	4	AIXLU	

## **Begriffserklärungen**

Eine Erläuterung der im Arbeitsblatt zur Konfiguration verwendeten Begriffe.

### **1 Befehlspräfix**

Dieser Begriff bezeichnet das eindeutige Befehlspräfix für das Warteschlangenmanager-Subsystem Ihres IBM MQ for z/OS-Systems. Der z/OS-Systemprogrammierer definiert diesen Wert bei der Installation in SYS1.PARMLIB(IEFSSNs); bei ihm können Sie den Wert erfragen.

### **2 Netz-ID**

Die VTAM-Startprozedur in Ihrer Installation wird teilweise vom Member ATCSTRxx des Datensatzes angepasst, auf das DDNAME VTAMLST verweist. Die Netz-ID ist der für den Parameter NETID in diesem Member angegebene Wert. Für die Netz-ID müssen Sie den Namen des Parameters NETID

angeben, der Eigner des IBM MQ-Kommunikationssubsystems ist. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

### **3 Knotenname**

VTAM verfügt als Low Entry-Netzknoten nicht über einen Steuerpunktnamen für Advanced Peer-to-Peer Networking (APPN). Ein Steuerpunktnamen für Systemservice (SSCPNAME) liegt jedoch vor. Für den Knotennamen müssen Sie den Namen des Parameters SSCP angeben, der Eigner des IBM MQ-Kommunikationssubsystems ist. Dieser Wert ist im gleichen Member ATCSTRxx als Netz-ID angegeben. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

### **9 Name der lokalen LU**

Eine logische Einheit (logical unit, LU) ist eine Software, die als Schnittstelle bzw. als Übersetzer zwischen einem Transaktionsprogramm und dem Netz dient. Sie verwaltet den Austausch von Daten zwischen Transaktionsprogrammen. Der Name der lokalen logischen Einheit ist die eindeutige Anwendungs-ID der VTAM dieses IBM MQ-Subsystems. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

### **11 12 13 Symbolisches Ziel**

Dieser Begriff steht für den Namen, den Sie dem Nebeninformationsprofil CPI-C gegeben haben. Für jedes LU 6.2-Empfangsprogramm muss ein Nebeninformationseintrag vorhanden sein.

### **6 14 Modusname**

Dies ist der festgelegte Name der Parametergruppe, die den LU 6.2-Datenaustausch steuert. Ein Eintrag mit diesem Namen und ähnliche Attribute müssen an beiden Enden der Sitzung definiert werden. In VTAM entspricht dies einem Modustabelleneintrag. Ihr Netzadministrator kann Ihnen diesen Tabelleneintrag zuweisen.

### **7 15 Transaktionsprogrammname**

IBM MQ-Anwendungen, die versuchen, Daten mit diesem Warteschlangenmanager auszutauschen, geben einen symbolischen Namen für das Programm an, das am Empfangsende ausgeführt werden soll. Dieses ist im Attribut TPNAME der Kanaldefinition beim Sender festgelegt. Verwenden Sie der Einfachheit halber wenn möglich immer den Transaktionsprogrammnamen MQSERIES und bei Verbindungen mit VSE/ESA, wenn die Länge des Namens auf 4 Byte beschränkt ist, verwenden Sie MQTP.

Weitere Informationen finden Sie im Abschnitt [LU6.2-Verbindung für z/OS mit APPC/MVS definieren](#).

### **8 LAN-Zieladresse**

Dies steht für die LAN-Zieladresse, die Ihre Partnerknoten für die Kommunikation mit diesem Host verwenden. Wenn Sie mit einem 3745-Netzcontroller arbeiten, ist dies der im Parameter LOCADD angegebene Wert für die Leitungsdefinition, mit der Ihr Partner physisch verbunden ist. Wenn Ihre Partnerknoten andere Geräte verwenden, beispielsweise 317X- oder 6611-Geräte, wird die Adresse bei der Anpassung dieser Geräte festgelegt. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

### **10 Name der generischen Ressource**

Ein generischer Ressourcename ist ein eindeutiger Name, der einer Gruppe von LU-Namen zugewiesen ist, die von den Kanalinitiatoren in einer Gruppe mit gemeinsamer Warteschlange verwendet werden.

### **16 Name der Partner-LU**

Dies ist der LU-Name des IBM MQ-Warteschlangenmanagers in dem System, mit dem Sie die Kommunikation einrichten. Dieser Wert ist im Nebeninformationseintrag für den fernen Partner angegeben.

### **21 ID des fernen Knotens**

Für eine Verbindung zu Windows ist diese ID die ID des lokalen Knotens in dem Windows-System, mit dem Sie die Kommunikation einrichten.

## **z/OS LU 6.2-Verbindung in einer Gruppe mit gemeinsamer Warteschlange aufbauen**

Eine LU 6.2-Verbindung wird in zwei Schritten hergestellt. Sie definieren sich selbst im Netz und Sie definieren eine Verbindung zum Partner.

## **Selbstdefinition im Netz mithilfe generischer Ressourcen**

Sie können VTAM Generic Resources verwenden, um einen einzigen Verbindungsnamen für die Verbindung zur Gruppe mit gemeinsamer Warteschlange zu erhalten.

1. SYS1.PARMLIB(APPCPMxx) enthält die Startparameter für APPC. Dieser Datei müssen Sie eine Zeile hinzufügen, um APPC mitzuteilen, wo sich die Nebeninformationen befinden. Die Zeile muss folgendes Format aufweisen:

```
SIDEINFO
  DATASET (APPC.APPCSI)
```

2. Fügen Sie SYS1.PARMLIB(APPCPMxx) eine weitere Zeile hinzu, um den Namen der lokalen LU zu definieren, der für das Empfangsprogramm der IBM MQ LU 6.2-Gruppe verwendet werden soll. Diese Zeile muss folgendes Format haben:

```
LUADD ACBNAME(mvslu1)
      NOSCHED
      TPDATA(csq.appctp)
      GRNAME(mvsgrr)
```

Geben Sie Werte für ACBNAME (9), TPDATA und GRNAME (10) an.

Der Parameter NOSCHED teilt APPC mit, dass die neue logische Einheit nicht den LU 6.2-Scheduler (ASCH) verwendet, jedoch über keinen eigenen verfügt. TPDATA bezieht sich auf den Transaktionsprogramm-Datensatz, in dem LU 6.2 Informationen über Transaktionsprogramme speichert. Auch diesen Parameter verwendet IBM MQ nicht, er ist jedoch aufgrund der Syntax des Befehls LUADD erforderlich.

3. Starten Sie das APPC-Subsystem mit dem Befehl:

```
START APPC, SUB=MSTR, APPC=xxx
```

Dabei ist *xxx* das Suffix des PARMLIB-Members, in dem Sie die LU in Schritt 1 hinzugefügt haben.

**Anmerkung:** Wenn die APPC bereits ausgeführt wird, kann Sie durch folgenden Befehl aktualisiert werden:

```
SET APPC=xx
```

Die Auswirkungen hiervon sind kumulativ, das heißt, APPC vergisst keine Informationen zu Objekten, die bereits in diesem Member oder einem anderen PARMLIB-Member definiert sind.

4. Fügen Sie die neue LU einer geeigneten VTAM-Hauptknotendefinition hinzu. Diese befinden Sie in der Regel in SYS1.VTAMLST. Die APPL-Definition sieht wie das abgebildete Beispiel aus.

```

MVSLU APPL ACBNAME=MVSLU1,      9
          APPXC=YES,
          AUTOSES=0,
          DDRAINL=NALLOW,
          DLOGMOD=#INTER,      6
          DMINWML=10,
          DMINWNR=10,
          DRESPL=NALLOW,
          DSESLIM=60,
          LMDENT=19,
          MODETAB=MTCICS,
          PARSESS=YES,
          VERIFY=NONE,
          SECACPT=ALREADYV,
          SRBEXIT=YES
```

5. Aktivieren Sie den Hauptknoten. Diese Aktivierung kann durch folgenden Befehl erfolgen:

```
V, NET, ACT, majornode
```



6. Fügen Sie der CPI-C-Nebeninformationsdatei Einträge hinzu, die Ihre LU und den generischer Ressourcennamen definieren. Verwenden Sie hierfür das APPC-Dienstprogramm ATBSDFMU. Die Beispiel-JCL befindet sich in *thlqual.SCSQPROC(CSQ4SIDE)* (hierbei steht *thlqual* für das übergeordnete Qualifikationsmerkmal für die Zielbibliothek für IBM MQ-Datensätze in der Installation).

Die hinzuzufügenden Einträge haben das im folgenden Beispiel dargestellte Format:

```
SIADD
  DESTNAME(G1)          11
  MODENAME(#INTER)
  TPNAME(MQSERIES)
  PARTNER_LU(MVSLU1)    9
SIADD
  DESTNAME(G2)          12
  MODENAME(#INTER)
  TPNAME(MQSERIES)
  PARTNER_LU(MVSGR)     10
```

7. Damit die richtigen Parameter für die Steuerung verteilter Warteschlangen verwendet werden, modifizieren Sie das Warteschlangenmanagerobjekt durch folgenden Befehl. Sie müssen die lokale LU (9) angeben, die Ihrem Warteschlangenmanager im Attribut LUGROUP des Warteschlangenmanagers zugewiesen ist.

```
ALTER QMGR LUGROUP(MVSLU1)
```

### **Verbindung zu einem Partner definieren**

Sie können eine Verbindung zu einem Partner definieren, indem Sie der CPI-C-Nebeninformationsdatei einen Eintrag hinzufügen.

**Anmerkung:** Dieses Beispiel gilt für eine Verbindung zu einem Windows-System, die Aufgabe ist jedoch für andere Plattformen die gleiche.

Fügen Sie zur Definition der Verbindung in der CPI-C-Nebeninformationsdatei einen Eintrag hinzu. Eine Beispiel-JCL für diese Definition befindet sich in *thlqual.SCSQPROC(CSQ4SIDE)*.

Der hinzuzufügende Eintrag sieht wie folgt aus:

```
SIADD
  DESTNAME(M3)          13
  MODENAME(#INTER)     14
  TPNAME(MQSERIES)     15
  PARTNER_LU(WINNTLU)  16
```

### **Wie geht es weiter?**

Die Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen.

Fahren Sie dazu mit „[Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS](#)“ auf Seite 74 fort.

### **TCP-Verbindung über Sysplex Distributor aufbauen**

Sie können Sysplex Distributor so einrichten, dass ein einziger Verbindungsname für die Verbindung zur Gruppe mit gemeinsamer Warteschlange verwendet wird.

1. Definieren Sie durch folgende Schritte eine verteilte DVIPA-Adresse:
  - a. Fügen Sie IPCONFIG eine DYNAMICXCF-Anweisung hinzu. Diese Anweisung dient zur Verbindung zwischen Images mithilfe dynamisch erstellter XCF TCP/IP-Verknüpfungen.
  - b. Verwenden Sie den Block VIPADYNAMIC auf allen Images im Sysplex.
    - i) Fügen Sie auf dem Eigner-Image eine VIPADEFINE-Anweisung in den Code ein, um die DVIPA zu erstellen. Fügen Sie anschließend eine VIPADISTRIBUTE-Anweisung in den Code ein, um sie auf alle anderen oder ausgewählte Images zu verteilen.

- ii) Fügen Sie auf dem Backup-Image eine VIPABACKUP-Anweisung für die DVIPA-Adresse ein.
2. Wenn mehrere Kanalinitiatoren auf einer beliebigen logischen Partition im Sysplex gestartet werden, fügen Sie für den gemeinsamen Anschluss in die Reservierungsliste PORT im Datensatz PROFILE die Option SHAREPORT ein.

Weitere Informationen hierzu finden Sie in den Veröffentlichungen *z/OS CS: IP Configuration Guide* und *z/OS CS: IP Configuration Reference*.

Sysplex Distributor verteilt die Eingangsverbindungen gleichmäßig zwischen den einzelnen logischen Partitionen. Wenn sich auf einer logischen Partition mehrere Kanalinitiatoren befinden, wird durch die Verwendung von SHAREPORT die Eingangsverbindung an den Anschluss des Empfangsprogramms mit der geringsten Anzahl an Verbindungen übergeben.

Nachdem diese Schritte ausgeführt wurden, ist die TCP-Verbindung aufgebaut. Sie können die Konfiguration abschließen.

Fahren Sie dazu mit „[Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS](#)“ auf Seite 74 fort.

## **Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS**

Konfigurieren Sie den gemeinsamen Kanal, indem Sie den Kanalinitiator starten und die entsprechenden Befehle für Ihre Konfiguration ausgeben.

1. Starten Sie den Kanalinitiator durch folgenden Befehl:

```
/cpf START CHINIT
```

2. Starten Sie durch Eingabe des folgenden Befehls ein Empfangsprogramm für die LU 6.2-Gruppe:

```
/cpf START LSTR TRPTYPE(LU62) LUNAME( G1 ) INDISP(GROUP)
```

LUNAME von G1 bezieht sich auf den symbolischen Namen, den Sie Ihrer logischen Einheit (11) gegeben haben.

3. Wenn Sie in Verbindung mit Sysplex Distributor mit virtueller IP-Adressvergabe arbeiten und über eine bestimmte Adresse empfangsbereit sein möchten, verwenden Sie folgenden Befehl:

```
/cpf START LSTR TRPTYPE(TCP) PORT(1555) IPADDR( mvsvipa ) INDISP(GROUP)
```

Es kann immer nur eine Instanz des gemeinsam genutzten Kanals aktiv sein. Versuche, eine zweite Instanz des Kanals zu starten, schlagen fehl (die ausgegebene Fehlermeldung variiert und hängt von weiteren Faktoren ab). Die gemeinsame Synchronisationswarteschlange verfolgt den Kanalstatus.

IBM MQ-Kanäle werden nicht erfolgreich initialisiert, wenn bei der Kanalfestlegung erkannt wird, dass die Nachrichtenfolgennummern an den beiden Enden unterschiedlich sind. Unter Umständen müssen Sie diese manuell zurücksetzen.

## **Beispiel für die Konfiguration gemeinsamer Kanäle**

Zur Konfiguration eines gemeinsamen Kanals sind eine Reihe von Schritten erforderlich.

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte für den z/OS-Warteschlangenmanager zur Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for z/OS und Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Werte im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern,

müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

*Tabelle 10. Arbeitsblatt zur Konfiguration für IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange*

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>QSG</b>	
B	Name der lokalen Warteschlange		<b>QSG.SHAREDQ</b>	
<b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Windows“ auf Seite 58 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (LU 6.2)		<b>QSG.WINNT.SNA</b>	
H	Name des Senderkanals (TCP)		<b>QSG.WINNT.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>WINNT.QSG.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>WINNT.QSG.TCP</b>	
<b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für AIX“ auf Seite 11 verwendeten Werten wie angegeben entsprechen.				
C	Name des fernen Warteschlangenmanagers		<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (LU 6.2)		<b>QSG.AIX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>QSG.AIX.TCP</b>	
I	Name des Empfängerkanals (LU 6.2)	G	<b>AIX.QSG.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>AIX.QSG.TCP</b>	

**z/OS Definitionen gemeinsamer Senderkanäle in IBM MQ for z/OS**

Eine Beispielformatdefinition für gemeinsame Senderkanäle für LU 6.2 und TCP.

## LU 6.2 verwenden

```
Local Queue
  Object type : QLOCAL
  Name       : WINNT           F
  Usage      : X (XmitQ)
  Disposition : SHARED

  Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ   D
  Name on remote system : WINNT.LOCALQ   E
  Remote system name : WINNT     C
  Transmission queue : WINNT     F
  Disposition : GROUP

  Sender Channel
  Channel name : MVS.WINNT.SNA   G
  Transport type : L (LU6.2)
  Transmission queue name : WINNT   F
  Connection name : M3           13
  Disposition : GROUP
```

## TCP verwenden

```
Local Queue
  Object type : QLOCAL
  Name       : WINNT           F
  Usage      : X (XmitQ)
  Disposition : SHARED

  Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ   D
  Name on remote system : WINNT.LOCALQ   E
  Remote system name : WINNT     C
  Transmission queue : WINNT     F
  Disposition : GROUP

  Sender Channel
  Channel name : QSG.WINNT.TCP   H
  Transport type : T (TCP)
  Transmission queue name : WINNT   F
  Connection name : winnt.tcpip.hostname
  Disposition : GROUP
```

## Definitionen gemeinsamer Empfängerkanäle in IBM MQ for z/OS

Eine Beispielformatdefinition gemeinsamer Empfängerkanäle für LU 6.2 und TCP.

## LU 6.2 verwenden

```
Local Queue
  Object type : QLOCAL
  Name       : QSG.SHAREDQ     B
  Usage      : N (Normal)
  Disposition : SHARED

  Receiver Channel
  Channel name : WINNT.QSG.SNA   I
  Disposition : GROUP
```

## TCP verwenden

```
Local Queue
  Object type : QLOCAL
  Name       : QSG.SHAREDQ     B
  Usage      : N (Normal)
  Disposition : SHARED
```

```
Receiver Channel
Channel name : WINNT.QSG.TCP      J
Disposition : GROUP
```

## z/OS MQ-Beispielkonfiguration für z/OS unter Verwendung der gruppeninternen Warteschlangensteuerung

In diesem Abschnitt wird beschrieben, wie eine typische Anwendung zur Lohnbuchhaltungsabfrage, die derzeit zur Übertragung kurzer Nachrichten zwischen Warteschlangenmanagern die verteilte Steuerung von Warteschlangen verwendet, für die Nutzung von Gruppen mit gemeinsamen Warteschlangen und gemeinsam genutzten Warteschlangen migriert werden kann.

Zur Veranschaulichung der Verwendung der verteilten Steuerung von Warteschlangen, gruppeninternen Warteschlangensteuerung mit gemeinsam genutzten Warteschlangen und gemeinsam genutzten Warteschlangen werden drei Konfigurationen beschrieben. Die zugehörigen Diagramme zeigen nur den Datenfluss in eine Richtung, und zwar vom Warteschlangenmanager QMG1 zum Warteschlangenmanager QMG3.

### z/OS Konfiguration 1

Konfiguration 1 beschreibt, wie derzeit mit der verteilten Steuerung von Warteschlangen Nachrichten zwischen den Warteschlangenmanagern QMG1 und QMG3 übertragen werden.

Konfiguration 1 zeigt ein System für die verteilte Warteschlangensteuerung, mit dem Nachrichten, die der Warteschlangenmanager QMG1 von der Lohnbuchhaltungsabfrage empfängt, an den Warteschlangenmanager QMG2 und schließlich an den Warteschlangenmanager QMG3 übertragen werden, von dem aus sie dann an den Lohnbuchhaltungsserver gesendet werden.

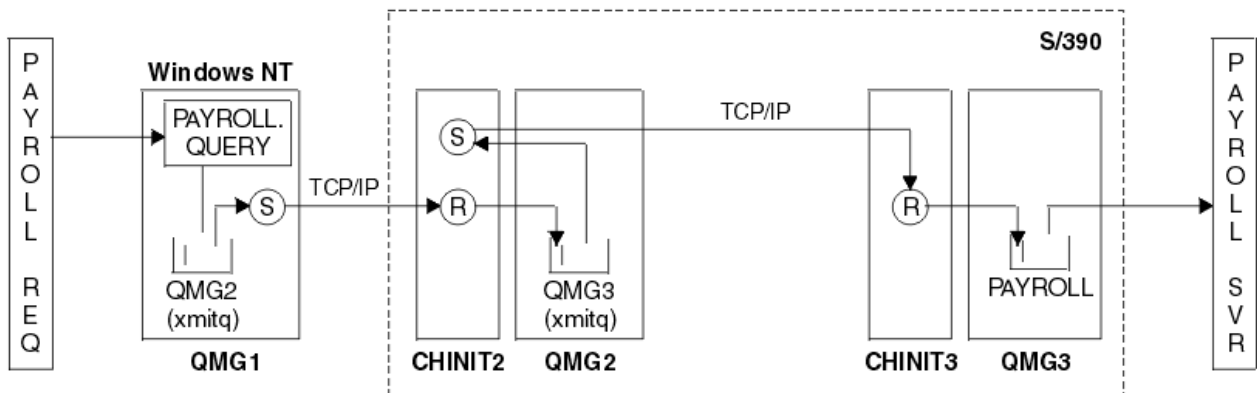


Abbildung 2. Konfiguration 1: z/OS unter Verwendung der gruppeninternen Warteschlangensteuerung

Der Ablauf der Operationen lautet wie folgt:

1. Eine Abfrage wird unter Verwendung der Anwendung zur Lohnbuchhaltungsabfrage eingegeben, die mit dem Warteschlangenmanager QMG1 verbunden ist.
2. Die Anwendung zur Lohnbuchhaltungsabfrage reiht die Abfrage in die ferne Warteschlange PAYROLL.QUERY ein. Die Warteschlange PAYROLL.QUERY wird in die Übertragungswarteschlange QMG2 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG2 eingereiht.
3. Der Senderkanal (S) auf dem Warteschlangenmanager QMG1 übergibt die Abfrage an den Partnerempfängerkanal (R) auf dem Warteschlangenmanager QMG2.
4. Der Empfängerkanal (R) auf dem Warteschlangenmanager QMG2 reiht die Abfrage in die Warteschlange PAYROLL auf dem Warteschlangenmanager QMG3 ein. Die Warteschlange PAYROLL auf dem QMG3 wird in die Übertragungswarteschlange QMG3 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG3 eingereiht.

5. Der Senderkanal (S) auf Warteschlangenmanager QMG2 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG3.
6. Der Empfängerkanal (R) auf Warteschlangenmanager QMG3 reißt die Abfrage in die lokale Warteschlange PAYROLL ein.
7. Die Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist, ruft die Abfrage aus der lokalen Warteschlange PAYROLL ab, verarbeitet sie und generiert eine passende Antwort.

## Definitionen für Konfiguration 1

Konfiguration 1 erfordert die nachfolgenden Definitionen (beachten Sie, dass die Definitionen kein Auslösen berücksichtigen und dass nur Kanaldefinitionen zur Kommunikation mittels TCP/IP angegeben werden).

### Auf QMG1

Definition der fernen Warteschlange:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

Definition der Empfangswarteschlange für Antworten:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

### Auf QMG2

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)

DEFINE QLOCAL(QMG3) DESCR('Transmission queue to QMG3') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

Hier ersetzen Sie WINTQMG1(1414) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG3') XMITQ(QMG3) CONNAME('MVSQMG3(1416)')
```

Hier ersetzen Sie MVSQMG3(1416) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.  
Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')  
  
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG3')
```

## Auf QMG3

Definition der lokalen Warteschlange:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +  
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE  
  
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.  
Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG2')
```

## Konfiguration 2

Konfiguration 2 beschreibt, wie Nachrichten mittels Gruppen mit gemeinsamer Warteschlange und der gruppeninternen Warteschlangensteuerung ohne Auswirkung auf die Lohnbuchhaltungsserveranwendung auf dem Back-End zwischen den Warteschlangenmanagern QMG1 und QMG3 übertragen werden können.

Konfiguration 2 zeigt ein System für die verteilte Warteschlangensteuerung, mit dem Nachrichten unter Verwendung von Gruppen mit gemeinsamer Warteschlange und gruppeninterner Warteschlangensteuerung von der Anwendung zur Lohnbuchhaltungsabfrage an den Lohnbuchhaltungsserver übertragen werden. Dank dieser Konfiguration entfallen Kanaldefinitionen zwischen den Warteschlangenmanagern QMG2 und QMG3, da die Nachrichten mittels der gruppeninternen Warteschlangensteuerung zwischen diesen beiden Warteschlangenmanagern übertragen werden.

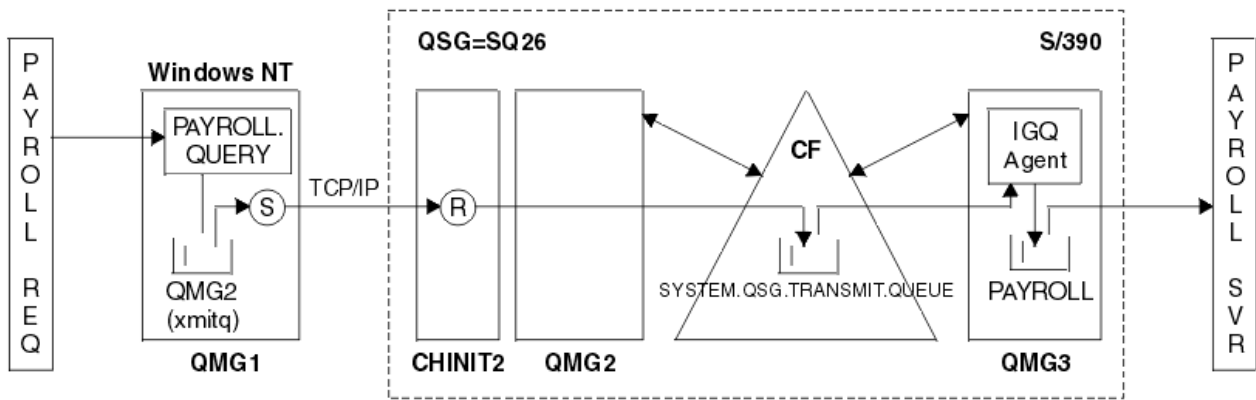


Abbildung 3. Konfiguration 2

Der Ablauf der Operationen lautet wie folgt:

1. Eine Abfrage wird unter Verwendung der Anwendung zu Lohnbuchhaltungsabfrage eingegeben, die mit dem Warteschlangenmanager QMG1 verbunden ist.
2. Die Anwendung zur Lohnbuchhaltungsabfrage reiht die Abfrage in die ferne Warteschlange PAYROLL.QUERY ein. Die Warteschlange PAYROLL.QUERY wird in die Übertragungswarteschlange QMG2 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG2 eingereiht.
3. Der Senderkanal (S) auf Warteschlangenmanager QMG1 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG2.
4. Der Empfängerkanal (R) auf Warteschlangenmanager QMG2 reiht die Abfrage in die Warteschlange PAYROLL auf Warteschlangenmanager QMG3 ein. Die Warteschlange PAYROLL auf QMG3 wird in die gemeinsame Übertragungswarteschlange YSTEM.QSG.TRANSMIT.QUEUE aufgelöst und die Abfrage wird in die gemeinsame Übertragungswarteschlange SYSYSTEM.QSG.TRANSMIT.QUEUE eingereiht.
5. Der IGQ-Agent auf dem Warteschlangenmanager QMG3 ruft die Abfrage aus der gemeinsamen Übertragungswarteschlange SYSTEM.QSG.TRANSMIT.QUEUE ab und stellt sie in die lokale Warteschlange PAYROLL auf dem Warteschlangenmanager QMG3.
6. Die Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist, ruft die Abfrage aus der lokalen Warteschlange PAYROLL ab, verarbeitet sie und generiert eine passende Antwort.

**Anmerkung:** Im Beispiel zur Lohnbuchhaltungsabfrage werden nur kompakte Nachrichten übertragen. Wenn Sie sowohl persistente als auch nicht persistente Nachrichten übertragen müssen, kann eine Kombination aus Konfiguration 1 und Konfiguration 2 eingerichtet werden. Auf diese Weise lassen sich lange Nachrichten unter Verwendung der Route der verteilten Warteschlangensteuerung übertragen, während kompakte Nachrichten über die potenziell schnellere Route der Warteschlangensteuerung übertragen werden.

## **z/OS** Definitionen für Konfiguration 2

Konfiguration 2 erfordert die nachfolgenden Definitionen (beachten Sie, dass die Definitionen kein Auslösen berücksichtigen und dass nur Kanaldefinitionen zur Kommunikation mittels TCP/IP angegeben werden).

Es wird angenommen, dass die Warteschlangenmanager QMG2 und QMG3 bereits als Mitglieder derselben Gruppe mit gemeinsamer Warteschlange konfiguriert wurden.

### Auf QMG1

Definition der fernen Warteschlange:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```



Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

Definition der Empfangswarteschlange für Antworten:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

## Auf QMG2

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)

DEFINE QLOCAL(SYSTEM.QSG.TRANSMIT.QUEUE) QSGDISP(SHARED) +
DESCR('IGQ Transmission queue') REPLACE PUT(ENABLED) USAGE(XMITQ) +
GET(ENABLED) INDXTYPE(CORRELID) CFSTRUCT('APPLICATION1') +
DEFSOPT(SHARED) DEFPSIST(NO)
```

Hier ersetzen Sie APPLICATION1 mit dem Namen Ihrer definierten CF-Struktur. Beachten Sie auch, dass diese Warteschlange, da es eine gemeinsam genutzte Warteschlange ist, nur auf einem der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange definiert werden muss.

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

Hier ersetzen Sie WINTQMG1(1414) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG1')
```

Warteschlangenmanagerdefinition:

```
ALTER QMGR IGQ(ENABLED)
```

## Auf QMG3

Definition der lokalen Warteschlange:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +  
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

Warteschlangenmanagerdefinition:

```
ALTER QMGR IGQ(ENABLED)
```

### Konfiguration 3

Konfiguration 3 beschreibt, wie Nachrichten mittels Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzten Warteschlangen ohne Auswirkung auf die Lohnbuchhaltungsserveranwendung auf dem Back-End zwischen den Warteschlangenmanagern QMG1 und QMG3 übertragen werden können.

Konfiguration 3 zeigt ein System für die verteilte Warteschlangensteuerung, mit dem Nachrichten unter Verwendung von Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzten Warteschlangen zwischen den Warteschlangenmanagern QMG1 und QMG3 ausgetauscht werden.

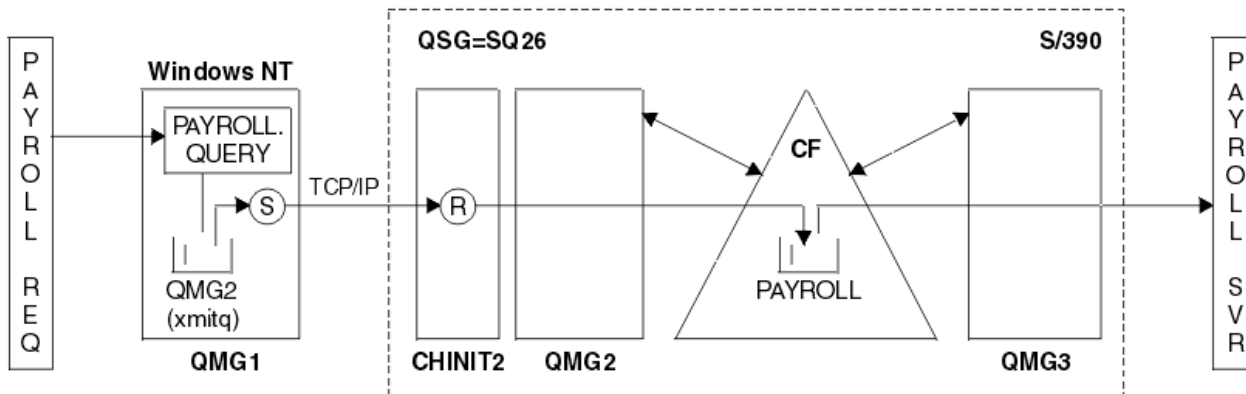


Abbildung 4. Konfiguration 3

Der Verfahrensablauf gliedert sich wie folgt:

1. Eine Abfrage wird unter Verwendung der Anwendung zur Lohnbuchhaltungsabfrage eingegeben, die mit dem Warteschlangenmanager QMG1 verbunden ist.
2. Die Anwendung zur Lohnbuchhaltungsabfrage reiht die Abfrage in die ferne Warteschlange PAYROLL.QUERY ein. Die Warteschlange PAYROLL.QUERY wird in die Übertragungswarteschlange QMG2 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG2 eingereicht.
3. Der Senderkanal (S) auf Warteschlangenmanager QMG1 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG2.
4. Der Empfängerkanal (R) auf Warteschlangenmanager QMG2 reiht die Abfrage in die gemeinsam genutzte Warteschlange PAYROLL ein.
5. Die Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist, ruft die Abfrage aus der gemeinsam genutzten Warteschlange PAYROLL ab, verarbeitet sie und generiert eine passende Antwort.

Diese Konfiguration ist zweifellos am einfachsten zu konfigurieren. Um Antworten (generiert von der Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist) von Warteschlangenmanager QMG3 an Warteschlangenmanager QMG2 und anschließend an Warteschlangenmanager QMG1 zu übertragen, muss jedoch eine verteilte Steuerung von Warteschlangen oder eine gruppeninterne Warteschlangensteuerung konfiguriert werden. (Informationen zur Konfiguration, die beim

Rücksenden von Antworten an die Anwendung zur Lohnbuchhaltungsabfrage verwendet wird, finden Sie unter „Erläuterung des Beispiels der Gruppe mit gemeinsamer Warteschlange für z/OS“ auf Seite 203.)

Auf QMG3 sind keine Definitionen erforderlich.

### **z/OS** Definitionen für Konfiguration 3

Konfiguration 3 erfordert die nachfolgenden Definitionen (beachten Sie, dass die Definitionen kein Auslösen berücksichtigen und dass nur Kanaldefinitionen zur Kommunikation mittels TCP/IP angegeben werden).

Es wird angenommen, dass die Warteschlangenmanager QMG2 und QMG3 bereits als Mitglieder derselben Gruppe mit gemeinsamer Warteschlange konfiguriert wurden.

#### **Auf QMG1**

Definition der fernen Warteschlange:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

Definition der Empfangswarteschlange für Antworten:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

#### **Auf QMG2**

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

Hier ersetzen Sie WINTQMG1(1414) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')
```

Definition der lokalen Warteschlange:

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) DESCR('Payroll query request queue') +  
REPLACE PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE +  
DEFSOPT(SHARED) DEFPSIST(NO) CFSTRUCT(APPLICATION1)
```

Hier ersetzen Sie APPLICATION1 mit dem Namen Ihrer definierten CF-Struktur. Beachten Sie auch, dass diese Warteschlange, da es eine gemeinsam genutzte Warteschlange ist, nur auf einem der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange definiert werden muss.

## Auf QMG3

Auf QMG3 sind keine Definitionen erforderlich.

### Beispiel ausführen

Nachdem Sie das Beispiel eingerichtet haben, können Sie es ausführen.

Für Konfiguration 1:

1. Starten Sie die Warteschlangenmanager QMG1, QMG2 und QMG3.
2. Starten Sie die Kanalinitiatoren für QMG2 und QMG3.
3. Starten Sie die Empfangsprogramme auf QMG1, um an Port 1414 empfangsbereit zu sein, auf QMG2, um an Port 1415 empfangsbereit zu sein und auf QMG3, um an Port 1416 empfangsbereit zu sein.
4. Starten Sie die Senderkanäle auf QMG1, QMG2 und QMG3.
5. Starten Sie die mit QMG1 verbundene Anwendung zur Lohnbuchhaltungsabfrage.
6. Starten Sie die mit QMG3 verbundene Lohnbuchhaltungsserveranwendung.
7. Übergeben Sie eine Lohnbuchhaltungsabfrage an QMG3 und warten Sie auf die Antwort der Lohnbuchhaltung.

Für Konfiguration 2:

1. Starten Sie die Warteschlangenmanager QMG1, QMG2 und QMG3.
2. Starten Sie den Kanalinitiator für QMG2.
3. Starten Sie die Empfangsprogramme auf QMG1, um an Port 1414 empfangsbereit zu sein, und auf QMG2, um an Port 1415 empfangsbereit zu sein.
4. Starten Sie den Senderkanal auf QMG1 und QMG2.
5. Starten Sie die mit QMG1 verbundene Anwendung zur Lohnbuchhaltungsabfrage.
6. Starten Sie die mit QMG3 verbundene Lohnbuchhaltungsserveranwendung.
7. Übergeben Sie eine Lohnbuchhaltungsabfrage an QMG3 und warten Sie auf die Antwort der Lohnbuchhaltung.

Für Konfiguration 3:

1. Starten Sie die Warteschlangenmanager QMG1, QMG2 und QMG3.
2. Starten Sie den Kanalinitiator für QMG2.
3. Starten Sie die Empfangsprogramme auf QMG1, um an Port 1414 empfangsbereit zu sein, und auf QMG2, um an Port 1415 empfangsbereit zu sein.
4. Starten Sie die Senderkanäle auf QMG1 und QMG2.
5. Starten Sie die mit QMG1 verbundene Anwendung zur Lohnbuchhaltungsabfrage.

6. Starten Sie die mit QMG3 verbundene Lohnbuchhaltungsserveranwendung.
7. Übergeben Sie eine Lohnbuchhaltungsabfrage an QMG3 und warten Sie auf die Antwort der Lohnbuchhaltung.

## **Beispiel erweitern**

Das Beispiel kann auf zahlreiche Art erweitert werden.

Sie können das Beispiel:

- Zur Verwendung von Kanal-Triggering und Anwendungs-Triggering (Warteschlange PAYROLL und PAYROLL.REPLY) erweitern
- Zur Kommunikation unter Verwendung von LU6.2 konfigurieren
- Zur Konfiguration von weiteren Warteschlangenmanagern für die Gruppe für gemeinsame Warteschlange erweitern. Danach können Sie die Serveranwendung klonen, sodass sie auf anderen Warteschlangenmanagerinstanzen ausgeführt werden kann, um der Abfragewarteschlange PAYROLL mehrere Server bereitzustellen.
- Zur Erhöhung der Instanzen der Anwendung zur Lohnbuchhaltungsabfrage erweitern, um die Verarbeitung von Abfragen aus mehreren Clients zu veranschaulichen
- Zur Verwendung von Sicherheit (IGQAUT und IGQUSER) erweitern

## **Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen**

In den folgenden Informationen wird die Sicherheit beschrieben, die auf die Dateien und Verzeichnisse unter /var/mqm/ angewendet wird. Außerdem wird erläutert, weshalb die Dateisystemberechtigungen auf die jeweiligen Werte gesetzt sind. Um sicherzustellen, dass IBM MQ ordnungsgemäß funktioniert, sollten Sie die von IBM MQ festgelegten Dateisystemberechtigungen nicht ändern.

### **Befehl `crtmqdir`**

Ab IBM MQ 9.0.3 gilt Folgendes: Wenn Ihr Unternehmen, aus welchem Grund auch immer, eine der /var/mqm-Dateiberechtigungen geändert hat, können Sie die Berechtigungen aktualisieren oder mit dem Befehl `crtmqdir` Verzeichnisse hinzufügen.

### **IBM MQ-Dateisystemsicherheit unter UNIX, Linux und IBM i**

In den Dateien unter dem IBM MQ-Datenverzeichnis (/var/mqm) werden folgende Daten gespeichert:

- IBM MQ-Konfigurationsdaten
- Anwendungsdaten (IBM MQ-Objekte und die in IBM MQ-Nachrichten enthaltenen Daten)
- Steuerinformationen der Laufzeit
- Überwachungsdaten (Nachrichten und FFST-Dateien)

Der Zugriff auf diese Daten wird über Dateisystemberechtigungen gesteuert, wobei einige der Daten für alle Benutzer zugänglich sind, während andere Daten auf Mitglieder der IBM MQAdministratorgruppe `mqm` (oder `QMQM` unter IBM i) beschränkt sind.

Die Erteilung des Zugriffs erfolgt in den folgenden drei Kategorien:

#### **Nur Gruppe 'mqm'**

Auf die Dateien und Verzeichnisse in dieser Kategorie können nur IBM MQ-Administratoren (Mitglieder der Gruppe `mqm`) und die IBM MQ-Warteschlangenmanagerprozesse zugreifen.

Die Dateiberechtigungen für diese Dateien und Verzeichnisse lauten wie folgt:

```
-IWXIXW--- mqm:mqm      (UNIX and Linux)
-IWXIXW--- QMQMADM:QMQM (IBM i)
```

Es folgt ein Beispiel der Dateien und Verzeichnisse in dieser Kategorie:

```
/var/mqm/qmgrs/QMGR/qm.ini
/var/mqm/qmgrs/QMGR/channel/
/var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
/var/mqm/qmgrs/QMGR/queues/
/var/mqm/qmgrs/QMGR/queues/SYSTEM!DEFAULT!LOCAL!QUEUES/
/var/mqm/qmgrs/QMGR/errors/
/var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
/var/mqm/qmgrs/QMGR/ssl/
/var/mqm/qmgrs/QMGR/@qmgr/
/var/mqm/qmgrs/QMGR/@qmpersist/
...
```

## Lesezugriff für alle Benutzer - Lese- und Schreibzugriff für Mitglieder der Gruppe 'mqm'

Die Dateien und Verzeichnisse in dieser Kategorie können von allen Benutzern gelesen werden, aber nur Mitglieder der Gruppe 'mqm' können diese Dateien ändern und diese Verzeichnisse bearbeiten.

Die Dateiberechtigungen für diese Dateien und Verzeichnisse lauten wie folgt:

```
-IWXIXWXI-X mqm:mqm      (UNIX and Linux)
-IWXIXWXI-X QMQMADM:QMQM (IBM i)
```

Es folgt ein Beispiel der Dateien und Verzeichnisse in dieser Kategorie:

```
/var/mqm/mqs.ini
/var/mqm/exits/
/var/mqm/qmgrs/
/var/mqm/qmgrs/QMGR/
/var/mqm/qmgrs/QMGR/@app/
/var/mqm/qmgrs/QMGR/@ipcc/
```



**Achtung:** Sie sollten Ausführungsberechtigungen nur für ausführbare Dateien und Scripts festlegen. Beispiel: Wenn der Befehl **crtmqm** unter Linux ausgeführt wird, werden die folgenden Dateiberechtigungen festgelegt:

```
-rW-rW---- mqm mqm /var/mqm/qmgrs/QMGR/qm.ini
-rW-rW---- mqm mqm /var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
-rW-rW---- mqm mqm /var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
-rW-rW-r-- mqm mqm /var/mqm/mqs.ini
```

IBM MQ 8.0:

```
/var/mqm/sockets/@SYSTEM
/var/mqm/sockets/QMGR/@app/hostname
/var/mqm/sockets/QMGR/@ipcc/hostname
```

## Lese- und Schreibzugriff für alle Benutzer Dateien mit Lese- und Schreibzugriff für alle Benutzer

IBM MQ hat keine *Regulär*-Dateien mit globalem Schreibzugriff (777). Es gibt jedoch eine Reihe von *Spezial*-Dateien, die global modifizierbare Dateiberechtigungen haben.

Diese speziellen Dateien bedeuten keine Sicherheitslücke. Obwohl die Berechtigungen als 777 angezeigt werden, handelt es sich nicht um *reguläre* Dateien. Sie können also keine Schreibvorgänge direkt in den Dateien vornehmen.

Es handelt sich hierbei um die folgenden speziellen Dateien:

### Symbolische Links

Symbolische Verbindungen lassen sich an dem Buchstaben 'l' am Anfang ihrer Berechtigungen erkennen. Die Berechtigungen für die symbolische Verbindung haben keinen Einfluss darauf, wer auf die Zieldatei zugreifen kann. Der Grund hierfür ist, dass der Zugriff auf den Befehl durch die Berechtigungen auf dem Ziel der symbolischen Verbindung gesteuert wird.

Da die Berechtigungen für symbolische Verbindungen auf den meisten UNIX- und Linux-Systemen nicht geändert werden können, werden sie stets als `lrwxrwxrwx` angezeigt.

### Socketdateien

Socketdateien sind spezielle Dateien, die vom Betriebssystem als Folge eines Prozesses erstellt werden, der ein UNIX-Domänensocket erstellt. Diese Dateien lassen sich am Buchstaben 's' am Anfang der Dateiberechtigungen erkennen (`srxwxrwxrwx`).

Die Berechtigungen für die Datei bedeuten nicht, dass der Zugriff auf die Datei selbst erteilt wird, sondern sie definieren, wer sich mit dem UNIX-Domänensocket verbinden kann.

IBM MQ verwendet mehrere dieser Socketdateien und die Festlegung der Berechtigungen richtet sich immer danach, wem die Kommunikation mit dem Socket erlaubt wird.

Die folgenden Verzeichnisse enthalten Socketdateien mit Schreib-/Leseberechtigung für alle Benutzer (`srxwxrwxrwx`).

IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketEC/hostname/Zsocket_*
```

Socketdateien, die von Anwendungen verwendet werden, welche sich unter Verwendung von isolierten Bindungen mit IBM MQ verbinden.

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/*
```

### Verzeichnisse mit Lese- und Schreibzugriff für alle Benutzer

In manchen Fällen müssen IBM MQ-Anwendungen Dateien unter dem IBM MQ-Datenverzeichnis erstellen. Um sicherzustellen, dass Anwendungen bei Bedarf Dateien erstellen können, wird für verschiedene Verzeichnisse ein globaler Schreibzugriff erteilt. Dies bedeutet, dass jeder Benutzer auf dem System Dateien in diesem Verzeichnis erstellen kann.

Mit Ausnahme der Fehlerprotokolldateien, die von jedem Mitglied der Gruppe 'mqm' beschrieben werden können, werden alle Dateien in diesen Verzeichnissen mit eingeschränkten Berechtigungen erstellt, die nur dem Dateiersteller Schreibzugriff erteilen. Dadurch kann der Systemadministrator die Benutzer-ID aller Daten verfolgen, die in Dateien dieser Verzeichnisse geschrieben werden.

#### **`/var/mqm/errors/`**

Dieses Verzeichnis enthält die Systemfehlerprotokolldateien und FFST-Dateien. Die Berechtigung für dieses Verzeichnis lautet `drwxrwsrwt`. Dies bedeutet, dass alle Benutzer auf dem System Dateien in diesem Verzeichnis erstellen können.

Das Bit 's' von SetGroupId zeigt an, dass alle der in diesem Verzeichnis erstellten Dateien die Zugehörigkeit zur Gruppe 'mqm' aufweisen.

Das Sticky Bit `t` ist für dieses Verzeichnis standardmäßig nicht festgelegt, aber ein IBM MQ-Administrator kann dies explizit festlegen, damit Benutzer nur die Dateien löschen können, die sie erstellen.

**Anmerkung:**  Diese Funktion ist unter IBM i nicht verfügbar.

#### **AMQERR0\*.LOG**

Diese Fehlerprotokolldateien können nur durch Mitglieder der entsprechenden Gruppe (group) direkt beschrieben werden, aber jeder Benutzer kann die Nachrichten, welche in diese Dateien geschrieben wurden, lesen (Berechtigung: `-rw-rw-r--`).

### AMQnnnnn.\*.FDC

Diese Dateien enthalten FFST-Informationen, die geschrieben werden, wenn im Warteschlangenmanager oder in einer von einem Benutzer geschriebenen Anwendung ein Fehler auftritt. Diese Dateien werden mit den Berechtigungen `-rw-r-----` erstellt.


#### `/var/mqm/trace/`

Wenn der IBM MQ-Trace aktiviert ist, werden Tracedateien in dieses Verzeichnis geschrieben. Der IBM MQ-Trace wird von allen Prozessen geschrieben, die einem Warteschlangenmanager mit aktiviertem Trace zugeordnet sind.

Die Berechtigungen für dieses Verzeichnis lauten `d1wx1ws1wt`. Dies bedeutet, dass alle Benutzer auf dem System Dateien in diesem Verzeichnis erstellen können.

Das Bit 's' von SetGroupId zeigt an, dass alle der in diesem Verzeichnis erstellten Dateien die Zugehörigkeit zur Gruppe 'mqm' aufweisen.

Das Sticky Bit `t` ist für dieses Verzeichnis standardmäßig nicht festgelegt, aber ein IBM MQ-Administrator kann dies explizit festlegen, damit Benutzer nur die Dateien löschen können, die sie erstellen.

**Anmerkung:**  Diese Funktion ist unter IBM i nicht verfügbar.

### AMQnnnnn.\*.TRC

Diese Dateien enthalten die Tracedaten, die von jedem Traceerstellungsprozess geschrieben und mit den Berechtigungen `-rw-r-----` erstellt werden.

Die Berechtigungen für dieses Verzeichnis lauten `d1wx1ws1wt`. Die Berechtigungen für die in diesem Verzeichnis erstellten Socketdateien lauten `s1wx-----`.

### IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketapp/hostname/
```


Dieses Verzeichnis wird von Anwendungen verwendet, die sich unter Verwendung von *isolierten* Bindungen mit IBM MQ verbinden. Während der Verbindungsverarbeitung wird von der verbindenden Anwendung in diesem Verzeichnis eine Socketdatei erstellt. Nachdem die Verbindung zu dem Warteschlangenmanager hergestellt wurde, wird die Socketdatei entfernt.

Die Berechtigungen für dieses Verzeichnis lauten `d1wx1ws1wt`. Die Berechtigungen für die in diesem Verzeichnis erstellten Socketdateien lauten `s1wx-----`.

Das SetGroupId-Bit `s` für dieses Verzeichnis stellt sicher, dass alle Dateien, die in diesem Verzeichnis erstellt werden, das Gruppeneigentumsrecht von 'mqm' haben.

Auf allen Plattformen außer IBM i ist für diese Verzeichnisse auch das Sticky Bit `t` gesetzt, das verhindert, dass ein Benutzer Dateien mit Ausnahme der Dateien, deren Eigner er ist, löschen kann. So können nicht berechtigte Benutzer keine Dateien löschen, die ihnen nicht gehören.

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/  
/var/mqm/sockets/QMGR/@app/ssem/hostname/
```

 Für Prozesse, die sich unter Verwendung von *gemeinsam genutzten* Bindungen mit IBM MQ verbinden, kann mithilfe von UNIX-Domänensockets eine Synchronisation zwischen der Anwendung und dem Warteschlangenmanager erfolgen. Wenn UNIX-Domänensockets verwendet werden, wird die zugehörige Socketdatei in diesen Verzeichnissen erstellt.

Die Berechtigungen für diese Verzeichnisse lauten `d1wx1ws1wt`. Die Berechtigungen für die in diesen Verzeichnissen erstellten Socketdateien lauten `s1wx1wx1wx`.

Das Bit 's' von SetGroupId bei diesen Verzeichnissen stellt sicher, dass alle der in diesen Verzeichnissen erstellten Dateien die Zugehörigkeit zur Gruppe 'mqm' aufweisen.

Auf allen Plattformen außer IBM i ist für diese Verzeichnisse auch das Sticky Bit `t` festgelegt, das verhindert, dass ein Benutzer Dateien mit Ausnahme der Dateien, deren Eigner er ist, löscht. So können nicht berechtigte Benutzer keine Dateien löschen, die ihnen nicht gehören.



## Verwendung von System V-IPC-Ressourcen durch IBM MQ

IBM MQ verwendet gemeinsam genutzten System V-Speicher und Semaphoren für die Kommunikation zwischen Prozessen. Die Gruppierung dieser Ressourcen richtet sich danach, wie sie in den einzelnen Gruppen mit der entsprechenden Eigentümerschaft und den jeweiligen Zugriffsberechtigungen verwendet werden.

Für die Prüfung, welche System V-IPC-Ressourcen auf einem System zu IBM MQ gehören, haben Sie folgende Möglichkeiten:

- Prüfen Sie die Eigentümerschaft.

Der Eigner von IBM MQ System V IPC-Ressourcen ist immer der Benutzer 'mqm' auf UNIX -Plattformen und Linux. Unter IBM i ist der Benutzer 'QMQM' der Eigner.

- Verwenden Sie in IBM MQ 8.0 und höher das Dienstprogramm amqspdbg.

Das mit IBM MQ ausgelieferte Dienstprogramm amqspdbg kann für die Anzeige des gemeinsam genutzten Speichers und der Semaphoren-IDs eines bestimmten Warteschlangenmanagers verwendet werden.

Sie müssen den Befehl einmalig für die Gruppe 'system' der von IBM MQ erstellten System V-Ressourcen ausgeben:

```
# amqspbg -z -I
```

Anschließend müssen Sie ihn vier Mal für jeden Warteschlangenmanager auf dem System ausgeben, um die vollständige Liste der von IBM MQ verwendeten System V-Ressourcen zu erhalten. In den folgenden Beispielen wird vorausgesetzt, dass als Warteschlangenmanagername QMGR1 festgelegt ist:

```
# amqspdbg -i QMGR1 -I
# amqspdbg -q QMGR1 -I
# amqspdbg -p QMGR1 -I
# amqspdbg -a QMGR1 -I
```

Die Zugriffsberechtigungen für die von IBM MQ erstellten System V-Ressourcen werden so festgelegt, dass berechtigten Benutzern immer die jeweils richtige Zugriffsstufe erteilt wird. Eine Reihe von System V-IPC-Ressourcen, die von IBM MQ erstellt wurden, sind für alle Benutzer auf der Maschine zugänglich und haben die Berechtigungen `-rw-rw-rw-`.

Der Parameter `-g Anwendungsgruppe` im Befehl `crtmqmk` kann verwendet werden, um den Zugriff auf einen Warteschlangenmanager auf die Zugehörigkeit zu einer bestimmten Betriebssystemgruppe zu beschränken. Die Verwendung der Funktionalität einer Gruppe mit eingeschränkten Rechten beschränkt die für die System V-IPC-Ressourcen erteilten Berechtigungen noch detaillierter.

Linux

UNIX

## IBM MQ-Dateiberechtigungen in /opt/mqm mit setuid für mqm

Die folgenden Informationen beziehen sich auf die Situation, dass das Sicherheitsteam einige der ausführbaren IBM MQ-Dateien in der Verzeichnisstruktur `$MQ_INSTALLATION_PATH` markiert hat, weil sie lokale Sicherheitsrichtlinien verletzen. Die Standardposition unter AIX ist `/usr/mqm` und für die anderen UNIX-Betriebssysteme ist es `/opt/mqm`. Die Informationen in diesem Abschnitt gelten auch dann, wenn Sie IBM MQ in einem anderen als dem Standardverzeichnis installiert haben (z. B. in `/opt/mqm90`) oder es in Ihrer Umgebung mehrere Installationen gibt.

### Problemursache

Das Sicherheitsteam hat unter `$MQ_INSTALLATION_PATH` die folgenden Problembereiche erkannt:

1. Dateien im Verzeichnis /opt/mqm/bin sind setuid für den Eigner der Verzeichnisstruktur, in der sie sich befinden. Beispiel:

```
dr-xr-xr-x   mqm mqm   ${MQ_INSTALLATION_PATH}/bin
-r-sr-s---   mqm mqm   ${MQ_INSTALLATION_PATH}/bin/addmqinf
-r-sr-s---   mqm mqm   ${MQ_INSTALLATION_PATH}/bin/amqcrsta
-r-sr-s---   mqm mqm   ${MQ_INSTALLATION_PATH}/bin/amqfcxba
...
```

2. Eigner praktisch aller Verzeichnisse und Dateien ist "mqm:mqm", außer der folgenden, deren Eigner der Root-Benutzer ist:

```
dr-xr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security
-r-sr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security/amqoamax
-r-sr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security/amqoampx
```

Eigner dieses Unterverzeichnisses muss der Root-Benutzer sein, da es sich um die ausführbaren Dateien handelt, die mit dem Betriebssystem interagieren, wenn der Benutzer über einen IBM MQ-Client ein Kennwort angibt. Das Kennwort wird vom IBM MQ-Warteschlangenmanager an das Betriebssystem übergeben, um zu bestätigen, ob das Kennwort gültig oder ungültig ist.

3. Der Benutzer ist nicht Eigner von Dateien im Verzeichnis /opt/mqm/lib/iconv (dieses Verzeichnis ist unter AIX nicht vorhanden). Beispiel:

```
dr-xr-xr-x   mqm mqm   ${MQ_INSTALLATION_PATH}/lib/iconv
-r--r--r--   bin bin   ${MQ_INSTALLATION_PATH}/lib/iconv/002501B5.tbl
-r--r--r--   bin bin   ${MQ_INSTALLATION_PATH}/lib/iconv/002501F4.tbl
-r--r--r--   bin bin   ${MQ_INSTALLATION_PATH}/lib/iconv/00250333.tbl
...
```

4. Das Fixpack-Wartungsverzeichnis auf RPM-basierten Linux-Systemen. Wenn Fixpacks installiert werden, werden die vorhandenen Dateien unter diesem Verzeichnis in einer Struktur gespeichert, die der im folgenden Beispiel ähnelt, mit der Ausnahme, dass in diesem Beispiel V.R die IBM MQ -Versions- und Releasenummer und die Unterverzeichnisse darstellt, die von den installierten Fixpacks abhängig sind:

```
drwx-----   root root   ${MQ_INSTALLATION_PATH}/maintenance
drwxr-xr-x   root root   ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.1
drwxr-xr-x   root root   ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.3
drwxr-xr-x   root root   ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.4
...
```

## Lösung des Problems

Eines der Probleme auf UNIX -Systemen in Bezug auf setuid-Programme war, dass die Systemsicherheit durch die Bearbeitung von Umgebungsvariablen wie LD\* (LD\_LIBRARY\_PATH, LIBPATH unter AIX usw.) beeinträchtigt werden könnte. Dies ist kein Problem mehr, weil verschiedene UNIX-Betriebssysteme diese LD\*-Umgebungsvariablen jetzt beim Laden von setuid-Programmen ignorieren.

1. Warum sind einige der IBM MQ-Programme mqm-setuid oder mqm-setgid?

In IBM MQ sind die Benutzer-ID "mqm" und alle IDs, die zur Gruppe "mqm" gehören, die IBM MQ -Benutzer mit Verwaltungsaufgaben.

IBM MQ-Warteschlangenmanagerressourcen werden geschützt, indem eine Authentifizierung dieser Benutzer durchgeführt wird. Da die Warteschlangenmanagerprozesse diese Warteschlangenmanagerressourcen verwenden und ändern, benötigen die Warteschlangenmanagerprozesse die Berechtigung "mqm", um auf die Ressourcen zugreifen zu können. Daher sind IBM MQ -Warteschlangenmanagerunterstützungsprozesse so konzipiert, dass sie mit der effektiven Benutzer-ID "mqm" ausgeführt werden können.

Damit auch Benutzer ohne Verwaltungsaufgaben auf IBM MQ-Objekte zugreifen können, stellt IBM MQ den Objektberechtigungsmanager (OAM) bereit, mit dem Berechtigungen erteilt und entzogen werden können, falls dies für die Anwendung, die vom Benutzer ohne Verwaltungsaufgaben ausgeführt wird, notwendig ist.

Mit der Möglichkeit, verschiedenen Ebenen von Authentifizierungen für Benutzer zu erteilen, und der Tatsache, dass **setuid**- und **setgid**-Programme LD\* -Variablen ignorieren, beeinträchtigen die Binär- und Bibliotheksdateien von IBM MQ die Sicherheit Ihres Systems in keiner Weise.

2. Ist es möglich, die Berechtigungen zu ändern, um die Sicherheitsrichtlinien des Unternehmens zu erfüllen, ohne die IBM MQ-Funktionalität zu gefährden?

Sie dürfen die Berechtigungen und Eigentumsrechte von IBM MQ -Binärdateien und -Bibliotheken nicht ändern. Die IBM MQ-Funktionalität kann unter solchen Änderungen leiden, sodass Warteschlangenmanagerprozesse möglicherweise nicht auf einige der Ressourcen zugreifen können.

Die Berechtigungen und die Eigentümerschaften stellen keinesfalls eine Sicherheitsbedrohung für das System dar.

Linux-Festplattenlaufwerke/Platten, auf denen IBM MQ installiert ist oder sich IBM MQ-Daten befinden, dürfen nicht mit der Option `nosuid` angehängt werden. Diese Konfiguration kann die Funktionalität von IBM MQ beeinträchtigen.

Weitere Informationen finden Sie im Abschnitt „Auf `/var/mqm` angewendete IBM MQ-Dateisystemberechtigungen“ auf Seite 85.

## Zugehörige Informationen

[Dateisystem](#)

## Windows IBM MQ-Dateisystemberechtigungen unter Windows

In den folgenden Informationen wird die Sicherheit beschrieben, die auf die Dateien und Verzeichnisse unter Windows angewendet wird. Um sicherzustellen, dass IBM MQ ordnungsgemäß funktioniert, sollten Sie die von IBM MQ festgelegten Dateisystemberechtigungen nicht ändern.

### Datenverzeichnis

**Anmerkung:** Die Berechtigungen, die für das Stammelement dieses Verzeichnisses festgelegt sind, werden in der gesamten Verzeichnisstruktur nach unten übernommen.

Für die Verzeichnisse unter dem Datenverzeichnis (DATADIR) sind die folgenden Berechtigungen festgelegt, abgesehen von den Ausnahmen, die im folgenden Text ausführlich beschrieben werden.

#### Administrators

Uneingeschränkter Zugriff

#### Gruppe 'mqm'

Uneingeschränkter Zugriff

#### SYSTEM

Uneingeschränkter Zugriff

#### Jeder

Lese- und Ausführungszugriff

Die Ausnahmen sind:

#### DATADIR\errors

Alle uneingeschränkten Zugriff

#### DATADIR\trace

Alle uneingeschränkten Zugriff

#### DATADIR\log

##### Administrators

Uneingeschränkter Zugriff

##### Gruppe 'mqm'

Uneingeschränkter Zugriff

##### SYSTEM

Uneingeschränkter Zugriff

**Jeder**  
Lesen

**DATADIR\log\\active**

**Administrators**  
Uneingeschränkter Zugriff

**Gruppe 'mqm'**  
Uneingeschränkter Zugriff

**SYSTEM**  
Uneingeschränkter Zugriff  
Kein Zugriff erteilt für 'Alle'.

## Frühere Releases des Produkts

In Releases des Produkts vor IBM MQ 8.0 befanden sich das Standardprogramm und die Standarddatenverzeichnisse an einer gemeinsamen Position.

In jeder Installation, die ursprünglich vor IBM MQ 8.0 installiert wurde. Die Daten- und Programmverzeichnisse, die an den Standardpositionen installiert und anschließend von dort aktualisiert wurden, bleiben an derselben Position (in C:\Program Files\IBM\WebSphere MQ).

Falls sich die Daten- und Programmverzeichnisse an einer gemeinsamen Position befinden, gelten die obigen Informationen nur für die Verzeichnisse, die zum Datenverzeichnis gehören, und nicht für die Verzeichnisse, die Teil des Programmverzeichnisses sind.

## Benennungseinschränkungen für Warteschlangen

Es gibt Einschränkungen bei der Länge von Warteschlangennamen. Einige Warteschlangennamen sind für Warteschlangen reserviert, die vom Warteschlangenmanager definiert werden.

### Einschränkungen bei der Länge von Namen

Die Namen von Warteschlangen können bis zu 48 Zeichen lang sein.

### Reservierte Warteschlangennamen

Namen, die mit "SYSTEM." beginnen, sind für Warteschlangen reserviert, die vom Warteschlangenmanager definiert werden. Mit dem Befehl **ALTER** oder **DEFINE REPLACE** können Sie diese Warteschlangendefinitionen an Ihre Installation anpassen. Folgende Namen sind für IBM MQ definiert:

Name der Warteschlange	Beschreibung
SYSTEM.ADMIN.ACTIVITY.QUEUE	Warteschlange für Aktivitätsberichte
SYSTEM.ADMIN.CHANNEL.EVENT	Warteschlange für Kanalereignisse
SYSTEM.ADMIN.COMMAND.EVENT	Warteschlange für Befehlsereignisse
SYSTEM.ADMIN.COMMAND.QUEUE	Warteschlange, an die PCF-Befehlsnachrichten gesendet werden
SYSTEM.ADMIN.CONFIG.EVENT	Warteschlange für Konfigurationsereignisse
SYSTEM.ADMIN.PERFM.EVENT	Warteschlange für Leistungsereignisse
SYSTEM.ADMIN.PUBSUB.EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM.ADMIN.QMGR.EVENT	Warteschlange für Warteschlangenmanagerereignisse
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Warteschlange für Traceroute-Antwortnachrichten

<b>Name der Warteschlange</b>	<b>Beschreibung</b>
SYSTEM.AUTH.DATA.QUEUE	Warteschlange mit den Zugriffskontrolllisten für den Warteschlangenmanager (nicht für z/OS)
SYSTEM.CHANNEL.INITQ	Initialisierungswarteschlange für Kanäle
SYSTEM.CHANNEL.SYNCQ	Warteschlange mit den Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	Warteschlange zum Auslösen (nicht unter z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange zur Kommunikation von Repository-Änderungen zwischen Warteschlangenmanagern
SYSTEM.CLUSTER.HISTORY.QUEUE	Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange mit Informationen zum Repository
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Mit der Warteschlange werden einzelne Übertragungswarteschlangen für jeden Clustersenderkanal erstellt.
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle von der Clusterunterstützung verwalteten Ziele
SYSTEM.COMMAND.INPUT	Warteschlange, an die Befehlsnachrichten unter z/OS gesendet werden
SYSTEM.COMMAND.REPLY.MODEL	Definition der Modellwarteschlange für Antworten auf Befehle (unter z/OS)
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten (nicht unter z/OS)
SYSTEM.DEFAULT.ALIAS.QUEUE	Standarddefinition der Aliaswarteschlange
SYSTEM.DEFAULT.INITIATION.QUEUE	Warteschlange für die Auslösung eines angegebenen Prozesses (nicht unter z/OS)
SYSTEM.DEFAULT.LOCAL.QUEUE	Standarddefinition der lokalen Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standarddefinition der Modellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standarddefinition der fernen Warteschlange
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	Lokale Warteschlange mit einer persistenten Kopie der permanenten Subskriptionen des Warteschlangenmanagers
SYSTEM.HIERARCHY.STATE	Warteschlange mit Informationen zum Status der Beziehungen zwischen Warteschlangenmanagern in einer Publish/Subscribe-Hierarchie
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen
SYSTEM.INTERNAL.REPLY.QUEUE	IBM MQ-interne Antwortwarteschlange (nicht unter z/OS)
SYSTEM.INTER.QMGR.CONTROL	Warteschlange in einer Publish/Subscribe-Hierarchie für den Empfang von Anforderungen von einem fernen Warteschlangenmanager zur Erstellung einer Proxy-Subskription
SYSTEM.INTER.QMGR.PUBS	Warteschlange in einer Publish/Subscribe-Hierarchie für den Empfang von Veröffentlichungen von einem fernen Warteschlangenmanager

Name der Warteschlange	Beschreibung
SYSTEM.INTER.QMGR.FANREQ	Warteschlange in einer Publish/Subscribe-Hierarchie für die Verarbeitung von Anforderungen zur Erstellung einer Proxy-Subskription auf einem fernen Warteschlangenmanager
SYSTEM.MQEXPLORER.REPLY.MODEL	Definition der Modellwarteschlange für Antworten auf IBM MQ Explorer
SYSTEM.MQSC.REPLY.QUEUE	Modellwarteschlangendefinition für Antworten auf WebSphere MQ-Scriptbefehle (nicht unter z/OS)
SYSTEM.QSG.CHANNEL.SYNCQ	Gemeinsam genutzte lokale Warteschlange zum Speichern von Nachrichten mit den Synchronisationsdaten für gemeinsame Kanäle (nur unter z/OS)
SYSTEM.QSG.TRANSMIT.QUEUE	Gemeinsam genutzte lokale Warteschlange, die vom Agenten zur gruppeninternen Warteschlangensteuerung für die Übertragung von Nachrichten zwischen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange verwendet wird (nur unter z/OS)
SYSTEM.RETAINED.PUB.QUEUE	Lokale Warteschlange mit Kopien aller ständigen Veröffentlichungen des Warteschlangenmanagers
SYSTEM.SELECTION.EVALUATION.QUEUE	IBM MQ-interne Warteschlange für Auswahlbewertungen (nicht unter z/OS)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM MQ-interne Warteschlange für Auswahlprüfungen (nicht unter z/OS)

## Benennungseinschränkungen für andere Objekte

Es gibt Einschränkungen bei der Länge von Objektnamen. Einige Objektnamen sind für Objekte reserviert, die vom Warteschlangenmanager definiert werden.

### Einschränkungen bei der Länge von Namen

Die Namen von Prozessen, Namenslisten, Clustern, Themen, Services und Authentifizierungsdatenobjekten können eine Länge von maximal 48 Zeichen haben.

Die Namen von Kanälen können eine Länge von maximal 20 Zeichen haben.

Die Namen von Speicherklassen können eine Länge von maximal 8 Zeichen haben.

Die Namen von CF-Strukturen können eine Länge von maximal 12 Zeichen haben.

### Reservierte Objektnamen

Namen, die mit SYSTEM beginnen, sind für vom Warteschlangenmanager definierte Objekte reserviert. Mit den Befehlen **ALTER** oder **DEFINE REPLACE** können Sie diese Objektdefinitionen an Ihre Installation anpassen. Folgende Namen sind für IBM MQ definiert:

Objektname	Beschreibung
SYSTEM.ADMIN.SVRCONN	Serververbindungskanal für die ferne Verwaltung eines Warteschlangenmanagers
SYSTEM.AUTO.RECEIVER	Standardempfängerkanal für automatische Definition (nur unter UNIX, Linux, and Windows)

Objektname	Beschreibung
SYSTEM.AUTO.SVRCONN	Standardserververbindungskanal für automatische Definition (nur Multiplatforms)
SYSTEM.BASE.TOPIC	Basisartikel zur ASPARENT-Auflösung. Wenn ein besonderes administratives Artikelobjekt keine übergeordneten administrativen Artikelobjekte hat, werden alle ASPARENT-Attribute von diesem Objekt übernommen.
SYSTEM.DEF.CLNTCONN	Standarddefinition für Clientverbindungskanal
SYSTEM.DEF.CLUSRCVR	Standarddefinition für Clusterempfängerkanal
SYSTEM.DEF.CLUSSDR	Standarddefinition für Clustersenderkanal
SYSTEM.DEF.RECEIVER	Standarddefinition für Empfängerkanal
SYSTEM.DEF.REQUESTER	Standarddefinition für Requesterkanal
SYSTEM.DEF.SENDER	Standarddefinition für Senderkanal
SYSTEM.DEF.SERVER	Standarddefinition für Serverkanal
SYSTEM.DEF.SVRCONN	Standarddefinition für Serververbindungskanal
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Definition eines Standardauthentifizierungsdatenobjekts zur Definition von Authentifizierungsdatenobjekten vom Typ CRLLDAP
SYSTEM.DEFAULT.AUTHINFO.OCSP	Definition eines Standardauthentifizierungsdatenobjekts zur Definition von Authentifizierungsdatenobjekten vom Typ OCSP
SYSTEM.DEFAULT.LISTENER.LU62	Standardmäßiges SNA-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.NETBIOS	Standardmäßiges NetBIOS-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.SPX	Standardmäßiges SPX-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.TCP	Standardmäßiges TCP/IP-Empfangsprogramm (nur Multiplatforms)
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.DEFAULT.PROCESS	Standarddefinition für Prozesse
SYSTEM.DEFAULT.SERVICE	Standardservice (nur Multiplatforms)
SYSTEM.DEFAULT.TOPIC	Standarddefinition für Prozesse
SYSTEM.QPUBSUB.QUEUE.NAMELIST	Eine Liste von Warteschlangen für das zu überwachende Queued Publish/Subscribe Interface
  SYSTEMST	Standardmäßige Speicherklassendefinition (nur unter z/OS)

## Auflösung des Warteschlangennamens

Dieser Abschnitt enthält Informationen zur Auflösung eines Warteschlangennamens, die von Warteschlangenmanagern auf der Sende- und Empfangsseite eines Kanals ausgeführt wird.

In größeren Netzen bietet die Verwendung von Warteschlangenmanagern gegenüber anderen Kommunikationsformen einige Vorteile. Diese Vorteile ergeben sich aus der Funktion der Namensauflösung im verteilten Warteschlangenmanagement. Die wichtigsten Vorteile lauten wie folgt:

- Anwendungen müssen keine Routing-Entscheidungen treffen.
- Anwendungen müssen die Netzstruktur nicht kennen.
- Netzverknüpfungen werden von Systemadministratoren erstellt.
- Die Netzstruktur wird von Netzplanern gesteuert.
- Zur Partitionierung des Datenverkehrs können mehrere Kanäle verwendet werden.

In der folgenden Abbildung wird ein Beispiel für die Auflösung von Warteschlangennamen gezeigt. Zu sehen sind zwei Maschinen in einem Netz; auf einer Maschine ist eine PUT-Anwendung aktiv, auf der anderen Maschine eine GET-Anwendung. Die Anwendungen kommunizieren miteinander über den IBM MQ-Kanal, der von den MCAs gesteuert wird. Für Anwendungen entspricht dieser Prozess dem Einreihen von Nachrichten in eine lokale Warteschlange.

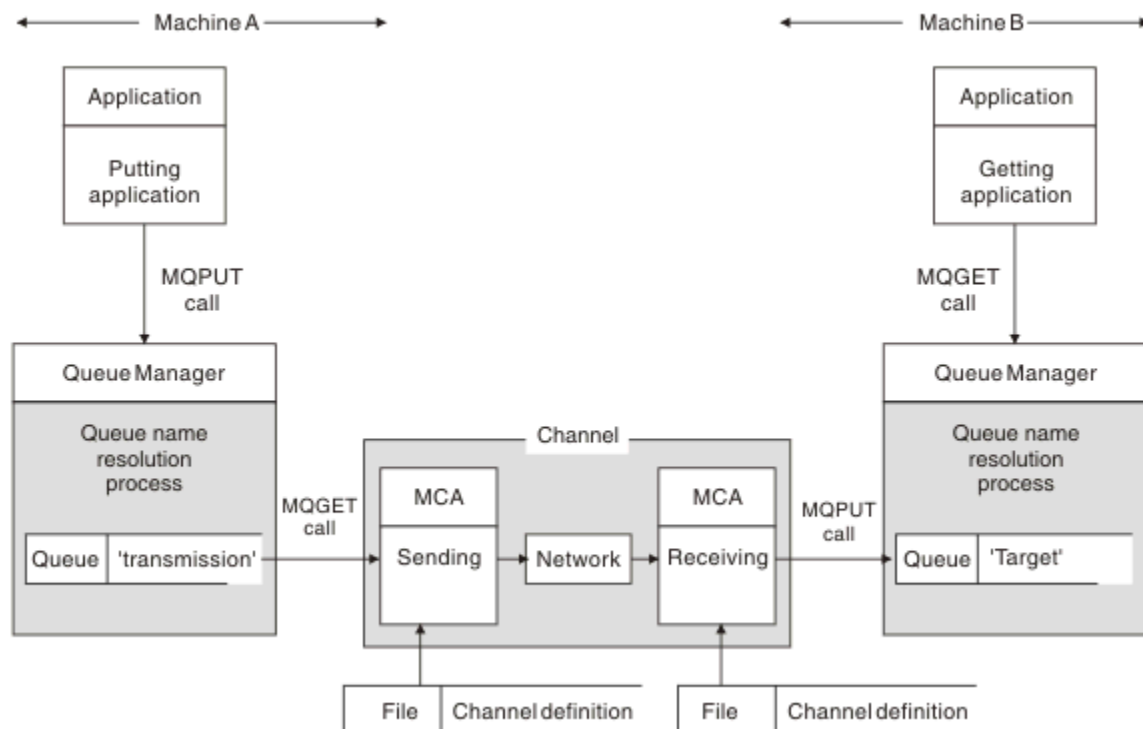


Abbildung 5. Namensauflösung

Wie in [Abbildung 5 auf Seite 96](#) beschrieben, ist der grundlegende Mechanismus für die Einreihung von Nachrichten in eine ferne Warteschlange in Bezug auf die Anwendung mit dem Prozess identisch, der für die Einreihung von Nachrichten in eine lokale Warteschlange genutzt wird:

- Die Anwendung, von der die Nachricht eingereicht wird, gibt zur Einreihung der Nachrichten in die Zielwarteschlange MQOPEN- und MQPUT-Aufrufe aus.
- Die Anwendung, von der die Nachricht abgerufen wird, gibt zum Abruf der Nachrichten aus der Zielwarteschlange MQOPEN- und MQGET-Aufrufe aus.

Sind beide Anwendungen mit demselben Warteschlangenmanager verbunden, ist keine Kommunikation zwischen Warteschlangenmanagern erforderlich, und die Zielwarteschlange wird beiden Anwendungen als *lokal* beschrieben.

Sind die Anwendungen jedoch mit unterschiedlichen Warteschlangenmanagern verbunden, werden zwei Nachrichtenkanalagenten mit ihrer zugehörigen Netzverbindung wie in der Abbildung dargestellt in die Übertragung einbezogen. In diesem Fall gilt die Zielwarteschlange bei der einreihenden Anwendung als *ferne Warteschlange*.



Der Ablauf der Ereignisse lautet wie folgt:

1. Die Anwendung, von der die Nachricht eingereicht wird, gibt zur Einreihung der Nachrichten in die Zielwarteschlange MQOPEN- und MQPUT-Aufrufe aus.
2. Während des MQOPEN-Aufrufs stellt die *Namensauflösung* fest, dass die Zielwarteschlange keine lokale Warteschlange ist, und entscheidet, welche Übertragungswarteschlange geeignet ist. Anschließend werden bei den MQPUT-Aufrufen, die dem MQOPEN-Aufruf zugeordnet sind, alle Nachrichten in diese Übertragungswarteschlange eingereicht.
3. Der sendende Nachrichtenkanalagent ruft die Nachrichten aus der Übertragungswarteschlange ab und übergibt sie beim fernen Computer an den empfangenden Nachrichtenkanalagenten.
4. Der empfangende Nachrichtenkanalagent reiht die Nachrichten in die Zielwarteschlange(n) ein.
5. Die Anwendung, von der die Nachricht abgerufen wird, gibt zum Abruf der Nachrichten aus der Zielwarteschlange MQOPEN- und MQGET-Aufrufe aus.

**Anmerkung:** Nur Schritt 1 und 5 enthalten Anwendungscode, während die Schritte 2 bis 4 von den lokalen Warteschlangenmanagern und Nachrichtenkanalagentenprogrammen ausgeführt werden. Die einreihende Anwendung kennt die Position der Zielwarteschlange nicht; diese kann sich in demselben Prozessor oder in einem anderen Prozessor auf einem anderen Kontinent befinden.

Die Kombination aus einem sendenden Nachrichtenkanalagenten, der Netzverbindung und dem empfangenden Nachrichtenkanalagenten wird als *Nachrichtenkanal* bezeichnet und ist grundsätzlich eine unidirektionale Einheit. Da Nachrichten für gewöhnlich in beide Richtungen fließen müssen, werden für diese Bewegung zwei Kanäle eingerichtet, und zwar einer für jeweils eine Richtung.

## Was ist die Auflösung von Warteschlangennamen?

Die Auflösung des Warteschlangennamens ist notwendig für das verteilte Warteschlangenmanagement (DQM). Anwendungen müssen die physische Adresse von Warteschlangen nicht mehr kennen und sie werden von den Netzwerkdetails isoliert.

Der Systemadministrator kann Warteschlangen von einem Warteschlangenmanager zu einem anderen verschieben und das Routing zwischen Warteschlangenmanagern ändern, ohne dass Anwendungen darüber informiert werden müssen.

Um den genauen Pfad, über den Daten weitergeleitet werden, vom Anwendungsdesign zu entkoppeln, ist es erforderlich, eine Zwischenstufe zwischen dem Namen, der von der Anwendung für die Zielwarteschlange verwendet wird, und der Benennung des Kanals einzuführen, über den der Datenfluss stattfindet. Diese Zwischenstufe wird durch die Funktion zur Auflösung des Warteschlangennamens erreicht.

Wenn eine Anwendung auf einen Warteschlangennamen verweist, wird der Name durch den Auflösungsmechanismus entweder einer Übertragungswarteschlange oder einer lokalen Warteschlange zugeordnet, die keine Übertragungswarteschlange ist. Für das Zuordnen zu einer Übertragungswarteschlange ist am Ziel eine zweite Namensauflösung erforderlich, dann wird die empfangene Nachricht in der Zielwarteschlange platziert, wie vom Anwendungsentwickler geplant. Die Anwendung hat keine Informationen zur Übertragungswarteschlange und zum Kanal, der zum Verschieben der Nachricht verwendet wurde.

**Anmerkung:** Für die Definition der Warteschlange und des Kanals ist das Systemmanagement zuständig. Sie können durch einen Operator oder ein Dienstprogramm für das Systemmanagement geändert werden, ohne dass die Anwendungen geändert werden müssen.

Eine wichtige Anforderung für das Systemmanagement von Nachrichtenflüssen besteht darin, zwischen Warteschlangenmanagern alternative Pfade bereitzustellen. So kann beispielsweise durch Geschäftsanforderungen festgelegt werden, dass unterschiedliche *Serviceklassen* über verschiedene Kanäle an dasselbe Ziel gesendet werden sollen. Dies ist eine Entscheidung des Systemmanagements, für die der Mechanismus zur Auflösung des Warteschlangennamens flexible Möglichkeiten bietet. Dies wird im Application Programming Guide detailliert erläutert. Die grundlegende Idee besteht jedoch darin, beim sendenden Warteschlangenmanager die Auflösung von Warteschlangennamen zu verwenden, um den von der Anwendung angegebenen Warteschlangennamen der für den jeweiligen Datenverkehrstyp geeigneten Übertragungswarteschlange zuzuordnen. Auf der Empfangsseite wird analog dazu durch die Auflösung von Warteschlangennamen der Name im Nachrichtendeskriptor einer lokalen Warteschlange

(keiner Übertragungswarteschlange) zugeordnet oder erneut einer geeigneten Übertragungswarteschlange zugeordnet.

Nicht nur der Weiterleitungspfad von einem Warteschlangenmanager zu einem anderen kann in verschiedene Datenverkehrstypen partitioniert werden, sondern auch die Antwortnachricht, die in der abgehenden Nachricht an die Definition der Empfangswarteschlange für Antworten gesendet wird, kann dieselbe Datenverkehrspartitionierung verwenden. Die Auflösung des Warteschlangennamens erfüllt diese Anforderung und der Anwendungsentwickler muss in diese Entscheidungen zur Partitionierung des Datenverkehrs nicht einbezogen werden.

Die Tatsache, dass die Zuordnung vom sendenden und empfangenden Warteschlangenmanager ausgeführt wird, ist ein wichtiger Aspekt bei der Namensauflösung. Durch diese Zuordnung kann der Warteschlangename, der durch die Anwendung angegeben wird, die den PUT-Vorgang durchgeführt hat, beim sendenden Warteschlangenmanager einer lokalen Warteschlange oder einer Übertragungswarteschlange und anschließend beim empfangenden Warteschlangenmanager erneut einer lokalen oder einer Übertragungswarteschlange zugeordnet werden.

Für Antwortnachrichten von empfangenden Anwendungen oder MCAs wird die Auflösung auf dieselbe Weise ausgeführt, sodass das Rückgaberrouting über bestimmte Pfade mit Warteschlangendefinitionen bei allen Warteschlangenmanagern entlang der Route ermöglicht wird.

## System- und Standardobjekte

Hier sind die über den Befehl **crtmqm** erstellten System- und Standardobjekte aufgeführt.

Wenn Sie mit dem Steuerbefehl **crtmqm** einen Warteschlangenmanager erstellen, werden die System- und Standardobjekte automatisch erstellt.

- Systemobjekte sind die für den Betrieb eines Warteschlangenmanagers oder Kanals erforderlichen IBM MQ-Objekte.
- In den Standardobjekten sind alle Attribute eines Objekts definiert. Wenn Sie ein Objekt erstellen (z. B. eine lokale Warteschlange), werden alle Attribute, die Sie nicht explizit angeben, aus dem Standardobjekt übernommen.

In den folgenden Tabellen sind die über den Befehl **crtmqm** erstellten System- und Standardobjekte aufgeführt:

- [Tabelle 11 auf Seite 98](#) enthält die System- und Standardwarteschlangenobjekte.
- In [Tabelle 12 auf Seite 100](#) sind die System- und Standardthemenobjekte aufgeführt.
- In [Tabelle 13 auf Seite 100](#) sind die System- und Standardkanalobjekte aufgeführt.
- In [Tabelle 14 auf Seite 101](#) sind die System- und Standardauthentifizierungsdatenobjekte aufgeführt.
- In [Tabelle 15 auf Seite 101](#) sind die System- und Standardempfangsprogrammobjekte aufgeführt.
- In [Tabelle 16 auf Seite 101](#) sind die System- und Standardnamenslistenobjekte aufgeführt.
- In [Tabelle 17 auf Seite 101](#) sind die System- und Standardprozessobjekte aufgeführt.
- In [Tabelle 18 auf Seite 101](#) sind die System- und Standardserviceobjekte aufgeführt.

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Die Warteschlange mit Berechnungsüberwachungsdaten.
SYSTEM.ADMIN.ACTIVITY.QUEUE	Die Warteschlange mit zurückgegebenen Aktivitätsberichten.
SYSTEM.ADMIN.CHANNEL.EVENT	Ereigniswarteschlange für Kanäle
SYSTEM.ADMIN.COMMAND.EVENT	Ereigniswarteschlange für Befehlsereignisse.

Tabelle 11. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.ADMIN.COMMAND.QUEUE	Warteschlange für Verwaltungsbefehle. Wird für ferne MQSC-Befehle und PCF-Befehle verwendet.
SYSTEM.ADMIN.CONFIG.EVENT	Ereigniswarteschlange für Konfigurationsereignisse.
SYSTEM.ADMIN.PERFM.EVENT	Ereigniswarteschlange für Leistungsereignisse
SYSTEM.ADMIN.PUBSUB.EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM.ADMIN.QMGR.EVENT	Ereigniswarteschlange für Warteschlangenmanagereignisse.
SYSTEM.ADMIN.STATISTICS.QUEUE	Die Warteschlange mit Statistiküberwachungsdaten.
SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE	Die Warteschlange für die Anzeige der Trace-Aktivitäten.
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Die Warteschlange für zurückgegebene Trace-Route-Antwortnachrichten.
SYSTEM.AUTH.DATA.QUEUE	Warteschlange mit den Zugriffskontrolllisten für den Warteschlangenmanager
SYSTEM.CHANNEL.INITQ	Kanalinitialisierungswarteschlange
SYSTEM.CHANNEL.SYNCQ	Warteschlange für die Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	CICS-Standardinitialisierungswarteschlange.
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange für die Übertragung von Nachrichten an den Repository-Warteschlangenmanager
SYSTEM.CLUSTER.HISTORY.QUEUE	Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Mit der Warteschlange werden einzelne Übertragungswarteschlangen für jeden Clustersenderkanal erstellt.
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange für die Speicherung aller Repositoryinformationen
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle Nachrichten an alle Cluster
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten.
SYSTEM.DEFAULT.ALIAS.QUEUE	Standardaliaswarteschlange
SYSTEM.DEFAULT.INITIATION.QUEUE	Standardinitialisierungswarteschlange
SYSTEM.DEFAULT.LOCAL.QUEUE	Standardmäßige lokale Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standardmodellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standardmäßige ferne Warteschlange
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen

Tabelle 11. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.MQEXPLORER.REPLY.MODEL	Die IBM MQ Explorer-Warteschlange für zu beantwortende Nachrichten. Hierbei handelt es sich um eine Modellwarteschlange, mit der eine temporäre dynamische Warteschlange für Antworten an den IBM MQ Explorer erstellt wird.
SYSTEM.MQSC.REPLY.QUEUE	Empfangswarteschlange für Antworten für MQSC-Befehle. Dies ist eine Modellwarteschlange, aus der eine temporäre dynamische Warteschlange für Antworten an ferne MQSC-Befehle erstellt wird.
SYSTEM.PENDING.DATA.QUEUE	Unterstützt verzögerte Nachrichten in JMS.

Tabelle 12. System- und Standardobjekte: Themen

Objektname	Beschreibung
SYSTEM.BASE.TOPIC	Basisthema für die ASPARENT-Auflösung. Gibt es zu einem bestimmten Thema keine übergeordneten Verwaltungsthemenobjekte oder weisen diese übergeordneten Objekte auch den Wert ASPARENT auf, werden alle verbleibenden ASPARENT-Attribute von diesem Objekt übernommen.
SYSTEM.DEFAULT.TOPIC	Standardthemendefinition.

Tabelle 13. System- und Standardobjekte: Kanäle

Objektname	Beschreibung
SYSTEM.AUTO.RECEIVER	Dynamischer Empfängerkanal
SYSTEM.AUTO.SVRCONN	Dynamischer Serververbindungskanal
SYSTEM.DEF.CLUSRCVR	Standardmäßiger Empfängerkanal für den Cluster zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLUSRCVR-Kanals auf einem Warteschlangenmanager im Cluster nicht angegeben werden.
SYSTEM.DEF.CLUSSDR	Standardmäßiger Senderkanal für den Cluster zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLUSSDR-Kanals auf einem Warteschlangenmanager im Cluster nicht angegeben werden.
SYSTEM.DEF.RECEIVER	Standardmäßiger Empfängerkanal
SYSTEM.DEF.REQUESTER	Standardmäßiger Requesterkanal
SYSTEM.DEF.SENDER	Standardmäßiger Senderkanal
SYSTEM.DEF.SERVER	Standardmäßiger Serverkanal
SYSTEM.DEF.SVRCONN	Standardmäßiger Serververbindungskanal
SYSTEM.DEF.CLNTCONN	Standardclientverbindungskanal.
V 9.0.0 V 9.0.0 SYSTEM.DEF.AMQP	AMQP-Standardkanal.

<i>Tabelle 14. System- und Standardobjekte: Authentifizierungsdatenobjekte</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Standardauthentifizierungsdatenobjekt zur Definition von Authentifizierungsdatenobjekten vom Typ CRLLDAP.
SYSTEM.DEFAULT.AUTHINFO.OCSP	Standardauthentifizierungsdatenobjekt zur Definition von Authentifizierungsdatenobjekten vom Typ OCSP.

<i>Tabelle 15. System- und Standardobjekte: Empfangsprogramme</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.LISTENER.TCP	TCP-Standardempfangsprogramm.
SYSTEM.DEFAULT.LISTENER.LU62 <sup>1</sup>	LU62-Standardempfangsprogramm.
SYSTEM.DEFAULT.LISTENER.NETBIOS <sup>1</sup>	NETBIOS-Standardempfangsprogramm.
SYSTEM.DEFAULT.LISTENER.SPX <sup>1</sup>	SPX-Standardempfangsprogramm.

1. Nur Windows

<i>Tabelle 16. System- und Standardobjekte: Namenslisten</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.NAMELIST	Standardnamensliste.

<i>Tabelle 17. System- und Standardobjekte: Prozesse</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.PROCESS	Standardprozessdefinition.

<i>Tabelle 18. System- und Standardobjekte: Services</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.SERVICE	Standardservice.
SYSTEM.BROKER	Publish/Subscribe-Broker

## **Windows** Objekte der Windows-Standardkonfiguration

Auf Windows-Systemen können Sie mithilfe der Anwendung IBM MQ Postcard eine Standardkonfiguration einrichten.

**Anmerkung:** Wenn auf Ihrem Computer andere Warteschlangenmanager vorhanden sind, können Sie keine Standardkonfiguration einrichten.

Für viele Namen, die für Objekte der Windows-Standardkonfiguration verwendet werden, wird ein TCP/IP-Kurzname verwendet. Dies ist der TCP/IP-Name des Computers ohne die Domänenkomponente. Der TCP/IP-Kurzname des Computers `mycomputer.hursley.ibm.com` beispielsweise lautet `mycomputer`. Immer wenn dieser Name abgeschnitten werden muss und das letzte Zeichen ein Punkt (.) ist, wird dieser entfernt.

Alle Zeichen in dem TCP/IP-Kurznamen, die für IBM MQ-Objektnamen nicht gültig sind (beispielsweise Bindestriche) werden durch ein Unterstreichungszeichen ersetzt.

Gültige Zeichen für IBM MQ -Objektnamen sind: a bis z, A bis Z, 0 bis 9 und die vier Sonderzeichen/%, und \_.

Der Clustername für die Windows-Standardkonfiguration ist DEFAULT\_CLUSTER.

Wenn es sich bei dem Warteschlangenmanager nicht um einen Repository-Warteschlangenmanager handelt, werden die in [Tabelle 19 auf Seite 102](#) aufgeführten Objekte erstellt.

<i>Tabelle 19. Von der Anwendung für die Windows-Standardkonfiguration erstellte Objekte</i>	
<b>Objekt</b>	<b>Name</b>
Warteschlangenmanager	<p>Der TCP/IP-Kurzname, dem die Zeichen QM_ vorangestellt sind. Die maximale Länge des Namens des Warteschlangenmanagers beträgt 48 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 48 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.</p> <p>Dem Warteschlangenmanager sind ein Befehlsserver, ein Kanalempfangsprogramm und ein Kanalinitiator zugeordnet. Das Kanalempfangsprogramm überwacht den IBM MQ-Standardport mit der Portnummer 1414. Alle anderen auf diesem System erstellten Warteschlangenmanager dürfen Port 1414 nicht verwenden, solange der Warteschlangenmanager der Standardkonfiguration vorhanden ist.</p>
Generischer Clusterempfängerkanal	<p>Der TCP/IP-Kurzname mit dem Präfix TO_QM_. Die maximale Länge des generischen Clusterempfängernamens beträgt 20 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 20 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.</p>
Clustersenderkanal	<p>Der Clustersenderkanal wird anfänglich mit dem Namen TO_ + QMNAME + erstellt. Wenn IBM MQ eine Verbindung zum Repository-Warteschlangenmanager für den Standardkonfigurationscluster hergestellt hat, wird dieser Name durch den Namen des Repository-Warteschlangenmanagers für den Standardkonfigurationscluster mit dem Präfix TO_ ersetzt. Die maximale Länge des Clustersenderkanalnamens beträgt 20 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 20 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.</p>
Lokale Nachrichtenwarteschlange	<p>Die lokale Nachrichtenwarteschlange hat die Bezeichnung default.</p>
Lokale Nachrichtenwarteschlange zur Verwendung durch die Anwendung IBM MQ Postcard	<p>Die lokale Nachrichtenwarteschlange zur Verwendung durch die Anwendung IBM MQ Postcard hat die Bezeichnung postcard.</p>
Serververbindungskanal	<p>Mithilfe des Serververbindungskanals können Clients eine Verbindung zum Warteschlangenmanager herstellen. Der Name ist der TCP/IP-Kurzname, dem die Zeichen S_ vorangestellt sind. Die maximale Länge des Serververbindungskanalnamens beträgt 20 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 20 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.</p>

Wenn es sich bei dem Warteschlangenmanager um einen Repository-Warteschlangenmanager handelt, ähnelt die Standardkonfiguration der in [Tabelle 19 auf Seite 102](#) beschriebenen Konfiguration, weist jedoch folgende Unterschiede auf:



- Der Warteschlangenmanager wird als Repository-Warteschlangenmanager für den Cluster der Standardkonfiguration definiert.
- Es wird kein Clustersenderkanal definiert.

- Es wird eine lokale Clusterwarteschlange erstellt, deren Name der TCP/IP-Kurzname mit den vorangestellten Zeichen clq\_default\_ ist. Die maximale Länge dieses Namens liegt bei 48 Zeichen. Namen, die diese Länge überschreiten, werden nach 48 Zeichen abgeschnitten.

Wenn Sie Einrichtungen zur Fernverwaltung anfordern, wird außerdem der Serververbindungskanal SYSTEM.ADMIN.SVRCONN erstellt.

## SYSTEM.BASE.TOPIC

Basisthema für die ASPARENT-Auflösung. Gibt es zu einem bestimmten Thema keine übergeordneten Verwaltungsthemenobjekte oder weisen diese übergeordneten Objekte auch den Wert ASPARENT auf, werden alle verbleibenden ASPARENT-Attribute von diesem Objekt übernommen.

<i>Tabelle 20. Standardwerte von SYSTEM.BASE.TOPIC</i>	
<b>Parameter</b>	<b>Wert</b>
TOPICSTR	"
CLROUTE	DIRECT
CLUSTER	Der Standardwert ist eine leere Zeichenfolge.
COMMINFO	SYSTEM.DEFAULT.COMMINFO.MULTICAST
DEFPRESP	SYNC
DEFPRTY	0
DEFPSIST	NO
DESCR	'Base topic for resolving attributes'
DURSUB	YES
MCAST	DISABLED
MDURMDL	SYSTEM.DURABLE.MODEL.QUEUE
MNDURMDL	SYSTEM.NDURABLE.MODEL.QUEUE
NPMSGDLV	ALLAVAIL
PMSGDLV	ALLDUR
PROXYSUB	FIRSTUSE
PUB	ENABLED
PUBSCOPE	ALL
  QSGDISP (nur z/OS-Plattform)	QMGR
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

Falls dieses Objekt nicht vorhanden ist, werden seine Standardwerte von IBM MQ dennoch für ASPARENT-Attribute verwendet, die nicht über übergeordnete Themen in der Themenstruktur aufgelöst werden.

Durch Einstellen des Attributs PUB oder SUB von SYSTEM.BASE.TOPIC auf DISABLED wird verhindert, dass Anwendungen Themen in der Themenstruktur publizieren oder subscribieren, mit zwei Ausnahmen:

1. Themenobjekte in der Themenstruktur, deren Attribut PUB oder SUB explizit auf ENABLE eingestellt ist. Anwendungen können diese Themen und die ihnen untergeordneten Elemente publizieren und abonnieren.
2. Die Publizierung und Subskription von SYSTEM . BROKER . ADMIN . STREAM wird durch die Einstellung des Attributs PUB oder SUB von SYSTEM . BASE . TOPIC auf DISABLED nicht inaktiviert.

Weitere Informationen finden Sie unter [Besondere Handhabung des Parameters PUB](#).

## IBM i System- und Standardobjekte für IBM i

Wird mit dem Befehl CRTMQM ein Warteschlangenmanager erstellt, werden die Systemobjekte und die Standardobjekte automatisch erstellt.

- Bei den Systemobjekten handelt es sich um die IBM MQ-Objekte, die für den Betrieb eines Warteschlangenmanagers oder Kanals erforderlich sind.
- In den Standardobjekten sind alle Attribute eines Objekts definiert. Wenn Sie ein Objekt erstellen (z. B. eine lokale Warteschlange), werden alle Attribute, die Sie nicht explizit angeben, aus dem Standardobjekt übernommen.

In den folgenden Tabellen werden die System- und Standardobjekte, die mit dem Befehl **CRTMQM** erstellt werden, aufgelistet:

- [Tabelle 21 auf Seite 104](#) enthält die System- und Standardwarteschlangenobjekte.
- In [Tabelle 22 auf Seite 106](#) sind die System- und Standardkanalobjekte aufgeführt.
- [Tabelle 23 auf Seite 107](#) enthält die System- und Standardauthentifizierungsdatenobjekte.
- [Tabelle 24 auf Seite 107](#) enthält das System- und Standardempfangsprogrammobjekt.
- [Tabelle 25 auf Seite 107](#) enthält die System- und Standardnamenslistenobjekte.
- [Tabelle 26 auf Seite 107](#) enthält das System- und Standardprozessobjekt.
- [Tabelle 27 auf Seite 107](#) enthält das System- und Standardserviceobjekt.

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM . ADMIN . ACCOUNTING . QUEUE	Abrechnungsnachrichtendaten, die bei der Trennung der Verbindung zwischen einer Anwendung und dem Warteschlangenmanager generiert werden
SYSTEM . ADMIN . ACTIVITY . QUEUE	Nachrichtendaten des Aktivitätsberichts.
SYSTEM . ADMIN . CHANNEL . EVENT	Ereigniswarteschlange für Kanäle
SYSTEM . ADMIN . COMMAND . QUEUE	Warteschlange für Verwaltungsbefehle. Wird für ferne MQSC-Befehle und PCF-Befehle verwendet.
SYSTEM . ADMIN . LOGGERS . EVENT	Nachrichtendaten für Protokollierungsereignis (Journalempfänger)
SYSTEM . ADMIN . PERFM . EVENT	Ereigniswarteschlange für Leistungsereignisse
SYSTEM . ADMIN . PUBSUB . EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM . ADMIN . QMGR . EVENT	Ereigniswarteschlange für Warteschlangenmanagerereignisse.
SYSTEM . ADMIN . STATISTICS . QUEUE	Warteschlange für MQI-, Warteschlangen- und Kanalstatistiknachrichtendaten



Tabelle 21. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Warteschlange für Traceroute-Antwortnachrichtendaten
SYSTEM.AUTH.DATA.QUEUE	Wird vom Objektberechtigungsmanager verwendet
SYSTEM.BROKER.ADMIN.STREAM	Verwaltungsdatenstrom, der von der Schnittstelle für eingereichtes Publish/Subscribe verwendet wird
SYSTEM.BROKER.CONTROL.QUEUE	Steuerwarteschlange für Publish/Subscribe-Schnittstelle
SYSTEM.BROKER.DEFAULT.STREAM	Standarddatenstrom, der von der Schnittstelle für eingereichtes Publish/Subscribe verwendet wird
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	Warteschlange für Kommunikation zwischen Brokern
SYSTEM.CHANNEL.INITQ	Kanalinitialisierungswarteschlange
SYSTEM.CHANNEL.SYNCQ	Warteschlange für die Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.DURABLE.MODEL.QUEUE	Warteschlange, die als Modell für verwaltete permanente Subskriptionen verwendet wird
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	Warteschlange für eine persistente Kopie der permanenten Subskriptionen im Warteschlangenmanager
SYSTEM.CICS.INITIATION.QUEUE	CICS-Standardinitialisierungswarteschlange.
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange für die Übertragung von Nachrichten an den Repository-Warteschlangenmanager
SYSTEM.CLUSTER.HISTORY.QUEUE	Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange für die Speicherung aller Repositoryinformationen
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Mit der Warteschlange werden einzelne Übertragungswarteschlangen für jeden Clustersenderkanal erstellt.
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle Nachrichten an alle Cluster
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten
SYSTEM.DEFAULT.ALIAS.QUEUE	Standardaliaswarteschlange
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Standardauthentifizierungsdatendefinition
SYSTEM.DEFAULT.INITIATION.QUEUE	Standardinitialisierungswarteschlange
SYSTEM.DEFAULT.LOCAL.QUEUE	Standardmäßige lokale Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standardmodellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standardmäßige ferne Warteschlange

*Tabelle 21. System- und Standardobjekte: Warteschlangen (Forts.)*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen
SYSTEM.HIERARCHY.STATE	Hierarchiebeziehungsstatus für verteiltes Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.CONTROL	Steuerwarteschlange für verteiltes Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.FANREQ	Eingabewarteschlange für Ausgabefächerungsprozesse der internen Proxy-Subskription des verteilten Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.PUBS	Veröffentlichungen des verteilten Publish/Subscribe von IBM MQ.
SYSTEM.MQEXPLORER.REPLY.MODEL	Die IBM MQ Explorer-Warteschlange für zu beantwortende Nachrichten. Hierbei handelt es sich um eine Modellwarteschlange, mit der eine temporäre dynamische Warteschlange für Antworten an den IBM MQ Explorer erstellt wird.
SYSTEM.MQSC.REPLY.QUEUE	Empfangswarteschlange für Antworten für MQSC-Befehle. Dies ist eine Modellwarteschlange, aus der eine temporäre dynamische Warteschlange für Antworten an ferne MQSC-Befehle erstellt wird.
SYSTEM.NDURABLE.MODEL.QUEUE	Warteschlange, die als Modell für verwaltete, nicht permanente Subskriptionen verwendet wird
SYSTEM.PENDING.DATA.QUEUE	Unterstützt verzögerte Nachrichten in JMS.
SYSTEM.RETAINED.PUB.QUEUE	Warteschlange für eine Kopie jeder ständigen Veröffentlichung im Warteschlangenmanager

*Tabelle 22. System- und Standardobjekte: Kanäle*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.AUTO.RECEIVER	Dynamischer Empfängerkanal
SYSTEM.AUTO.SVRCONN	Dynamischer Serververbindungskanal
SYSTEM.DEF.CLNTCONN	Standardmäßiger Clientverbindungskanal zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLNTCONN-Kanals auf einem Warteschlangenmanager nicht angegeben werden.
SYSTEM.DEF.CLUSRCVR	Standardmäßiger Empfängerkanal für den Cluster zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLUSRCVR-Kanals auf einem Warteschlangenmanager im Cluster nicht angegeben werden.
SYSTEM.DEF.CLUSSDR	Standardmäßiger Senderkanal für den Cluster zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLUSSDR-Kanals auf einem Warteschlangenmanager im Cluster nicht angegeben werden.

*Tabelle 22. System- und Standardobjekte: Kanäle (Forts.)*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEF.RECEIVER	Standardmäßiger Empfängerkanal
SYSTEM.DEF.REQUESTER	Standardmäßiger Requesterkanal
SYSTEM.DEF.SENDER	Standardmäßiger Senderkanal
SYSTEM.DEF.SERVER	Standardmäßiger Serverkanal
SYSTEM.DEF.SVRCONN	Standardmäßiger Serververbindungskanal

*Tabelle 23. System- und Standardobjekte: Authentifizierungsdatenobjekte*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Standardauthentifizierungsdatenobjekt für Authentifizierungstyp CRLLDAP.
SYSTEM.DEFAULT.AUTHINFO.OCSP	Standardauthentifizierungsdatenobjekt für Authentifizierungstyp OCSP.

*Tabelle 24. System- und Standardobjekte: Empfangsprogramme*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.LISTENER.TCP	Standardempfangsprogramm für TCP-Transport

*Tabelle 25. System- und Standardobjekte: Namenslisten*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.QPUBSUB.QUEUE.NAMELIST	Liste mit Warteschlangennamen, die von der Schnittstelle für eingereichtes Publish/Subscribe überwacht werden
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	Liste mit Themenobjekten, die von der Schnittstelle für eingereichtes Publish/Subscribe zum Abgleichen von Themenobjekten mit Subskriptionspunkten verwendet werden

*Tabelle 26. System- und Standardobjekte: Prozesse*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.PROCESS	Standardprozessdefinition.

*Tabelle 27. System- und Standardobjekte: Services*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.SERVICE	Standardservice.

## Informationen zu Zeilengruppen

Mithilfe der folgenden Informationen können Sie die Angaben innerhalb von Zeilengruppen konfigurieren. Zudem finden Sie hier eine Auflistung des Inhalts der Dateien `mqc.ini`, `qm.ini` und `mqclient.ini`.

## Zeilengruppen konfigurieren

Rufen Sie über die folgenden Links hilfreiche Informationen zur Konfiguration des Systems bzw. der Systeme in Ihrem Unternehmen auf:

- [IBM MQ-Konfigurationsdaten ändern](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *AllQueueManagers*
  - Zeilengruppe *DefaultQueueManager*
  - Zeilengruppe *ExitProperties*
  - Zeilengruppe *LogDefaults*
  - Zeilengruppe *Security* in der Datei *qm.ini*
- [Konfigurationsdaten von Warteschlangenmanagern ändern](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - **Windows** Zeilengruppe *AccessMode* (nur unter Windows)
  - Zeilengruppe *Service* - für installierbare Services
  - Zeilengruppe *Log*
  - **Linux** **UNIX** Zeilengruppe *RestrictedMode* (nur auf Systemen mit UNIX and Linux)
  - Zeilengruppe *XAResourceManager*
  - Zeilengruppen *TCP*, *LU62* und *NETBIOS*
  - Zeilengruppe *ExitPath*
  - Zeilengruppe *QMErrorLog*
  - Zeilengruppe *SSL*
  - Zeilengruppe *ExitPropertiesLocal*
- [Services und Komponenten konfigurieren](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *Service*
  - Zeilengruppe *ServiceComponent*

Außerdem finden Sie dort Links zu Informationen über deren Verwendung für die verschiedenen Services unter UNIX and Linux und auf Windows-Plattformen.
- [API-Exits konfigurieren](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *AllActivityTrace*
  - Zeilengruppe *ApplicationTrace*
- [Verhalten des Aktivitätstrace konfigurieren](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *ApiExitCommon*
  - Zeilengruppe *ApiExitTemplate*
  - Zeilengruppe *APIExitLocal*
- [Konfigurationsdaten für Clients](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *CHANNELS*
  - Zeilengruppe *ClientExitPath*
  - **Windows** Zeilengruppen *LU62*, *NETBIOS* und *SPX* (nur unter Windows)
  - Zeilengruppe *MessageBuffer*
  - Zeilengruppe *SSL*
  - Zeilengruppe *TCP*

- „Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen“ auf Seite 110 enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *CHANNELS*
  - Zeilengruppe *TCP*
  - Zeilengruppe *LU62*
  - *NETBIOS*
  - Zeilengruppe *ExitPath*
- Attribute von eingereichten Publish/Subscribe-Nachrichten festlegen enthält Informationen zur Konfiguration der folgenden Elemente:
  - Attribut *PersistentPublishRetry*
  - Attribut *NonPersistentPublishRetry*
  - Attribut *PublishBatchSize*
  - Attribut *PublishRetryInterval*
 in der Zeilengruppe *Broker*.



**Achtung:** Falls Sie eine *Broker*-Zeilengruppe benötigen, müssen Sie diese erstellen.

## Konfigurationsdateien

Unter

- Datei [mq<sub>s</sub>.ini](#)
- Datei [qm.ini](#)
- Datei [mqclient.ini](#)

Dort finden Sie eine Liste der möglichen Zeilengruppen in den einzelnen Konfigurationsdateien.

Linux

UNIX

### mq<sub>s</sub>.ini-Datei

Unter Beispiel einer IBM MQ-Konfigurationsdatei für Systeme mit UNIX and Linux finden Sie ein Beispiel für die Datei `mqs.ini`.

Eine `mqs.ini`-Datei kann die folgenden Zeilengruppen enthalten:

- [AllQueueManagers](#)
- [DefaultQueueManager](#)
- [ExitProperties](#)
- [LogDefaults](#)

Außerdem gibt es für jeden Warteschlangenmanager eine [QueueManager](#)-Zeilengruppe.


### Datei 'qm.ini'

Unter Beispiel einer Konfigurationsdatei für einen Warteschlangenmanager in IBM MQ für Systeme mit UNIX and Linux finden Sie ein Beispiel für die Datei `qm.ini`.

Eine `qm.ini`-Datei kann die folgenden Zeilengruppen enthalten:

- [ExitPath](#)
- [Protokoll](#)
- [QMErrorLog](#)
- [QueueManager](#)
- [Sicherheit](#)

- Service und ServiceComponent

 Verwenden Sie zum Konfigurieren von InstallableServices die Zeilengruppen Service und ServiceComponent.




- Connection für DefaultBindType



**Achtung:** Falls Sie eine Connection-Zeilengruppe benötigen, müssen Sie diese erstellen.

- SSL und TLS
- TCP, LU62 und NETBIOS
- XAResourceManager

Zusätzlich können Sie Folgendes ändern:

-  AccessMode (nur unter Windows)
-   RestrictedMode (nur auf Systemen mit UNIX and Linux)

Hierfür verwenden Sie den Befehl crtmqm.

### Datei 'mqclient.ini'

Eine mqclient.ini-Datei kann die folgenden Zeilengruppen enthalten:

- CHANNELS
- ClientExitPath
- LU62, NETBIOS und SPX
- MessageBuffer
- SSL
- TCP

Darüber hinaus wird möglicherweise eine Zeilengruppe PreConnect zur Konfiguration eines PreConnect-Exits benötigt.

## Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen

Beschreibung der Zeilengruppen der Warteschlangenmanager-Konfigurationsdatei qm.ini für die verteilte Steuerung von Warteschlangen

Dieser Abschnitt enthält die Zeilengruppen in der Warteschlangenmanager-Konfigurationsdatei für die verteilte Steuerung von Warteschlangen. Sie gelten für die Warteschlangenmanager-Konfigurationsdatei für IBM MQ for Multiplatforms. Der Name der Datei lautet auf allen Plattformen qm.ini.

Zeilengruppen für die verteilte Steuerung von Warteschlangen sind:

- Kanäle
- TCP
- LU62
- NETBIOS
- EXITPATH

In Abbildung 6 auf Seite 111 sind die Werte aufgeführt, die Sie mit diesen Zeilengruppen festlegen können. Wenn Sie eine dieser Zeilengruppen definieren, müssen Sie nicht jedes Element in einer neuen Zeile beginnen. Sie können entweder ein Semikolon (;) oder ein Nummernzeichen (#) verwenden, um einen Kommentar anzugeben.

```

CHANNELS:
MAXCHANNELS=n           ; Maximum number of channels allowed, the
                        ; default value is 100.
MAXACTIVECHANNELS=n    ; Maximum number of channels allowed to be active at
                        ; any time, the default is the value of MaxChannels.
MAXINITIATORS=n        ; Maximum number of initiators allowed, the default
                        ; and maximum value is 3.
MQIBINDTYPE=type1      ; Whether the binding for applications is to be
                        ; "fastpath" or "standard".
                        ; The default is "standard".
PIPELINELENGTH=n       ; The maximum number of concurrent threads a channel will use.
                        ; The default is 1. Any value greater than 1 is treated as 2.
ADOPTNEWMCA=chltype    ; Stops previous process if channel fails to start.
                        ; The default is "NO".
ADOPTNEWMCATIMEOUT=n   ; Specifies the amount of time that the new
                        ; process should wait for the old process to end.
                        ; The default is 60.
ADOPTNEWMCACHECHECK=   ; Specifies the type checking required.
  typecheck             ; The default is "NAME", "ADDRESS", and "QM".
CHLAUTHEARLYADOPT=Y/N ; The order in which connection authentication and channel authentica
tion rules are         ; processed. If not present in the qm.ini file the default is "N".




From MQ9.0.4 all
                        ; queue managers are created with a default of "Y"
PASSWORDPROTECTION=   ; From MQ8.0, set protected passwords in the MQCSP structure, rather
than using TLS.
  options              ; The options are "compatible", "always", "optional" and "warn"
                        ; The default is "compatible".
CHLAUTHISSUEWARN=Y    ; If you want message AMQ9787 to be generated when you set the WARN=YES
attribute
                        ; on the SET CHLAUTH command.
TCP:                  ; TCP entries
  PORT=n              ; Port number, the default is 1414
  KEEPALIVE=Yes       ; Switch TCP/IP KeepAlive on
LU62:
  LIBRARY2=DLLName2   ; Used if code is in two libraries
  EXITPATH:2          Location of user exits
  EXITPATHS=          ; String of directory paths.

```





Abbildung 6. Zeilengruppen der Datei qm.ini für die verteilte Steuerung von Warteschlangen

## Anmerkungen:

1. MQIBINDTYPE gilt nur auf den folgenden Plattformen:

-  AIX
-  HP-UX
-  IBM i
-  Solaris

2. EXITPATH gilt nur auf den folgenden Plattformen:

-  AIX
-  HP-UX
-  Solaris
-  Windows

## Zugehörige Informationen

### Konfiguration

 [z/OS konfigurieren](#)

[Konfigurationsdaten auf Windows-, UNIX- und Linux-Systemen ändern.](#)

 [Konfigurationsdaten unter IBM i ändern](#)

## Kanalattribute

In diesem Abschnitt werden die in den Kanaldefinitionen angegebenen Kanalattribute beschrieben.

Sie wählen die Attribute eines Kanals so aus, dass sie unter bestimmten Bedingungen für jeden Kanal optimal sind. Wenn der Kanal dann ausgeführt wird, wurden die tatsächlichen Werte jedoch möglicherweise im Rahmen von Startvereinbarungen geändert. Weitere Informationen finden Sie im Abschnitt [Kanäle vorbereiten](#).

Für viele Attribute gibt es Standardwerte, die für die meisten Kanäle verwendet werden können. Für Bedingungen, unter denen die Standardwerte nicht optimal sind, finden Sie in diesem Abschnitt eine Anleitung zur Auswahl der richtigen Werte.

Für Clusterkanäle geben Sie die Clusterkanalattribute auf den Ziel-Warteschlangenmanagern auf den Clusterempfängerkanälen an. Attribute, die Sie auf den zugehörigen Clustersenderkanälen angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie unter [Clusterkanäle](#).

**Anmerkung:** In IBM MQ for IBM i können die meisten Attribute als \*SYSDFTCHL angegeben werden, d. h., es wird der Wert des Standardkanals Ihres Systems übernommen.

## Kanalattribute und Kanaltypen

Unterschiedliche Kanaltypen unterstützen unterschiedliche Kanalattribute.

Die Kanaltypen für IBM MQ-Kanalattribute werden in der folgenden Tabelle aufgelistet.

**Anmerkung:** Wenn bei Clusterkanälen (die Tabellenspalten CLUSSDR und CLUSRCVR) ein Attribut für beide Kanäle gesetzt werden kann, definieren Sie das Attribut für beide Kanäle und achten Sie darauf, dass für beide Kanäle dieselbe Attributeinstellung verwendet wird. Bei unterschiedlichen Einstellungen werden in der Regel die Einstellungen verwendet, die Sie für den Kanal CLUSRCVR angegeben haben. Dies wird im Abschnitt [Clusterkanäle](#) erläutert.





Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
<a href="#">Datum ändern</a>	ALTDATE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Yes (Ja)
<a href="#">Zeit ändern</a>	ALTTIME	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Yes (Ja)
    AMQP keep alive	AMQPKA									Yes (Ja)
<a href="#">Intervall der Überwachungssignale für Stapel</a>	BATCHHB	Ja	Ja					Ja	Ja	
<a href="#">Stapelintervall</a>	BATCHINT	Ja	Ja					Ja	Ja	
<a href="#">Batch limit (Stapeldatengrenzwert)</a>	BATCHLIM	Ja	Ja					Ja	Ja	



Tabelle 28. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Be- fehlspara- meter	SDR	SVR	RCV R	RQST R	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR	<small>VS00</small> AM QP
<u>Stapelgröße</u>	BATCHSZ	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Zertifikatsbe- zeichnung</u>	CERTLABL	Ja	Ja	Ja	Ja	Ja	Ja	Yes (Ja) „1“ auf Seite 117	Ja	<small>VS00</small> Yes (Ja)
<u>Kanalname</u>	CHANNEL	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	<small>VS00</small> Yes (Ja)
<u>Kanalstatistik</u>	STATCHL	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Kanaltyp</u>	CHLTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	<small>VS00</small> Yes (Ja)
<u>Clientkanalge- wichtung</u>	CLNTWGH T					Ja				
<u>Cluster</u>	CLUSTER							Ja	Ja	
<u>Clusternamenslis- te</u>	CLUSNL							Ja	Ja	
<u>Clusterauslas- tungspriorität</u>	CLWLPRY							Ja	Ja	
<u>Clusterauslas- tungsrangordnung</u>	CLWLRAK							Ja	Ja	
<u>Clusterauslas- tungsgewichtung</u>	CLWLWGH T							Ja	Ja	
<u>Verbindungsaffini- tät</u>	AFFINITY					Ja				
<u>Verbindungsname</u>	CONNNAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Nachricht umwan- deln</u>	CONVERT	Ja	Ja					Ja	Ja	
<u>Datenkomprimie- rung</u>	COMPMSG	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Beschreibung</u>	DESCR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	<small>VS00</small> Yes (Ja)
<u>Unterbrechungs- intervall</u>	DISCINT	Ja	Ja				Yes (Ja) „2“ auf Seite 117	Ja	Ja	
<u>Disposition „2“ auf Seite 117</u>	QSGDISP	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	

Tabelle 28. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	<small>VS00</small> AM QP
<u>Header-Kompri- mierung</u>	COMPHDR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Intervall der Über- wachungssignale</u>	HBINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Keepalive-Inter- vall</u>	KAINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Lokale Adresse</u>	LOC- LADDR	Ja	Ja		Ja	Ja		Ja	Ja	<small>VS00</small> Yes (Ja)
<u>Zähler für Wieder- holungsversuche nach langem In- tervall</u>	LONGRTY	Ja	Ja					Ja	Ja	
<u>Langes Wiederho- lungsintervall</u>	LONGTMR	Ja	Ja					Ja	Ja	
<u>Gibt den LU 6.2- Modusnamen an.</u>	MODENA- ME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Gibt das LU 6.2-Transakti- onsprogramm an.</u>	TPNAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Maximale Instan- zen</u>	MAXINST						Ja			<small>VS00</small> Yes (Ja)
<u>Maximale Instan- zen pro Client</u>	MA- XINSTC						Ja			
<u>Maximale Nach- richtenlänge</u>	MAXMSGL	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	<small>VS00</small> Yes (Ja)
<u>Name des Nachrichten- kanalagenten</u>	MCANAME	Ja	Ja		Ja			Ja	Ja	
<u>Typ des Nachrich- tenkanalagenten</u>	MCATYPE	Ja	Ja		Ja			Ja	Ja	
<u>Benutzer des Nachrichten- kanalagenten</u>	MCAUSER	Ja	Ja	Ja	Ja		Ja	Ja	Ja	<small>VS00</small> Yes (Ja)
<u>Name des Nach- richtenexits</u>	MSGEXIT	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Benutzerdaten des Nachrichten- exits</u>	MSGDATA	Ja	Ja	Ja	Ja			Ja	Ja	

Tabelle 28. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Be- fehlspar- ameter	SDR	SVR	RCV R	RQST R	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR	<small>V9.0.0</small> AM QP
Name des Exits für Nachrichten- wiederholung	MREXIT			Ja	Ja				Ja	
Benutzerdaten des Exits für Nachrichten- wiederholungen	MRDATA			Ja	Ja				Ja	
Zähler für Nachrichten- wiederholungen	MRRTY			Ja	Ja				Ja	
Intervall für Nachrichten- wiederholungen	MRTMR			Ja	Ja				Ja	
Monitoring	MONCHL	Ja	Ja	Ja	Ja		Ja	Ja	Ja	
Netzverbin- dungspriorität	NETPRTY								Ja	
Gibt die Ge- schwindigkeit nicht permanenter Nachrichten an.	NPMSPEE D	Ja	Ja	Ja	Ja			Ja	Ja	
Password	KENN- WORT	Ja	Ja		Ja	Ja		Ja		
<small>V9.0.0</small> <small>V9.0.0</small> Port- nummer	PORT									<small>V9.0.0</small> Yes (Ja)
Eigenschaften- steuerung	PROPCTL	Ja	Ja					Ja	Ja	
PUT-Berechtigung	PUTAUT			Ja	Ja		Yes (Ja) „2“ auf Seite 117		Ja	
Warteschlangen- managername	QMNAME					Ja				
Name des Emp- fangsexits	RCVEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
Benutzerdaten des Empfangs- exits	RCVDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
Name des Sicher- heitsexits	SCYEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	


Tabelle 28. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	<small>V9.0.0</small> AM QP
<u>Benutzerdaten des Sicherheits-exits</u>	SCYDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Name des Sendexits</u>	SENDEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Sendexits</u>	SENDDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Folgenummernserie</u>	SEQWRAP	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Gemeinsam genutzte Verbindungen</u>	SHA-RECNV					Ja	Ja			
<u>Zähler für Wiederholungsversuche nach kurzem Intervall</u>	SHORTRTY	Ja	Ja					Ja	Ja	
<u>Kurzes Wiederholungsintervall</u>	SHORTTMR	Ja	Ja					Ja	Ja	
<u>SSL-Verschlüsselungsspezifikation</u>	SSLCIPH	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	<small>V9.0.0</small> Yes (Ja)
<u>SSL-Clientauthentifizierung</u>	SSLCAUTH		Ja	Ja	Ja		Ja		Ja	<small>V9.0.0</small> Yes (Ja)
<u>SSL-Peer</u>	SSLPEER	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	<small>V9.0.0</small> Yes (Ja)
<small>V9.0.0</small> <small>V9.0.0</small> <u>Themen-Root</u>	TPROOT									<small>V9.0.0</small> Yes (Ja)
<u>Name der Übertragungswarteschlange</u>	XMITQ	Ja	Ja							
<u>Transporttyp</u>	TRPTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<small>V9.0.0</small> <small>V9.0.0</small> <u>Client-ID verwenden</u>	USECLTID									<small>V9.0.0</small> Yes (Ja)
<u>Warteschlange für nicht zustellbare Nachrichten verwenden</u>	USEDLQ	Ja	Ja	Ja	Ja			Ja	Ja	

Tabelle 28. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Be- fehlspar- parameter	SDR	SVR	RCV R	RQST R	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR	<small>VS00</small> AM QP
Benutzer-ID	USERID	Ja	Ja		Ja	Ja		Ja		

**Anmerkungen:**

- Dieses Attribut kann durch keine Verwaltungsschnittstelle abgefragt oder für CLUSSDR-Kanäle festgelegt werden. Sie erhalten eine MQRCCF\_WRONG\_CHANNEL\_TYPE-Nachricht. Das Attribut ist allerdings in CLUSSDR-Kanalobjekten vorhanden (einschließlich MQCD-Strukturen) und kann bei Bedarf durch einen CHAD-Exit programmgesteuert festgelegt werden.
-  Nur gültig unter z/OS.

### Zugehörige Konzepte

„Kanalattribute in alphabetischer Reihenfolge“ auf Seite 117

In diesem Abschnitt werden alle Attribute eines Kanalobjekts mit ihren gültigen Werten und ggf. Hinweisen zu ihrer Verwendung beschrieben.

### Zugehörige Informationen

[Referenzinformationen zu MQSC](#)

## Kanalattribute in alphabetischer Reihenfolge

In diesem Abschnitt werden alle Attribute eines Kanalobjekts mit ihren gültigen Werten und ggf. Hinweisen zu ihrer Verwendung beschrieben.

IBM MQ implementiert für bestimmte Plattformen unter Umständen nicht alle in diesem Abschnitt aufgeführten Attribute. Auf Ausnahmen und Plattformunterschiede wird ggf. in den einzelnen Attributbeschreibungen hingewiesen.

Das Schlüsselwort, das Sie in MQSC angeben können, wird bei jedem Attribut in eckigen Klammern angegeben.

Die Attribute sind in alphabetischer Reihenfolge geordnet.

### Datum ändern (ALTDATE)

Dieses Attribut ist das Datum, an dem die Definition zuletzt geändert wurde, im Format JJJJ-MM-TT.

Dieses Attribut ist für alle Kanaltypen gültig.

### Zeit ändern (ALTTIME)

Dieses Attribut ist die Zeit, zu der die Definition zuletzt geändert wurde, im Format hh:mm:ss.

Dieses Attribut ist für alle Kanaltypen gültig.

### AMQP keep alive (AMQPKA)

Mit dem Attribut **AMQPKA** können Sie eine Keepalive-Zeit für die AMQP-Clientverbindung angeben. Wenn der AMQP-Client innerhalb des Keepalive-Intervalls keine Frames gesendet hat, wird die Verbindung geschlossen.

Das Attribut **AMQPKA** bestimmt den Wert des Attributs "idle-timeout", das von IBM MQ an einen AMQP-Client gesendet wurde. Das Attribut gibt einen Zeitraum in Millisekunden an.

Wenn für **AMQPKA** ein Wert > 0 festgelegt ist, übergibt IBM MQ die Hälfte dieses Werts als Attribut für das Inaktivitätszeitlimit. Der Wert 10000 bewirkt beispielsweise, dass der Warteschlangenmanager einen Wert von 5000 für das Inaktivitätszeitlimit sendet. Der Client muss sicherstellen, dass mindestens alle

10.000 Millisekunden Daten an IBM MQ gesendet werden. Wenn in dieser Zeit keine Daten von IBM MQ empfangen werden, geht IBM MQ davon aus, dass die Verbindung zum Client abgebrochen wurde und es wird das Schließen der Verbindung mit der Fehlerbedingung `amqp:resource-limit-exceeded` erzwungen.

Der Wert AUTO oder 0 gibt an, dass IBM MQ kein Attribut für das Inaktivitätszeitlimit an den AMQP-Client übergibt.

Ein AMQP-Client kann jedoch einen eigenen Wert für das Inaktivitätszeitlimit übergeben. In diesem Fall übergibt IBM MQ Daten (oder einen leeren AMQP-Frame) mit mindestens dieser Häufigkeit, um den Client darüber zu informieren, dass er verfügbar ist.

## Stapelintervall der Überwachungssignale (BATCHHB)

Mit diesem Attribut können sendende Kanäle vor dem Festschreiben eines Nachrichtenstapels prüfen, ob der empfangende Kanal noch aktiv ist.

Damit ermöglicht das Stapelintervall der Überwachungssignale, dass der Stapel zurückgesetzt wird und nicht unbestätigt bleibt, wenn der empfangende Kanal inaktiv ist. Durch das Zurücksetzen des Stapels bleiben die Nachrichten für die Verarbeitung verfügbar und können zum Beispiel an einen anderen Kanal umgeleitet werden.

Wenn der sendende Kanal vom empfangenden Kanal innerhalb des Überwachungsintervalls für den Stapelbetrieb ein Meldung erhalten hat, wird davon ausgegangen, dass der empfangende Kanal immer noch aktiv ist. Andernfalls wird zur Überprüfung ein Überwachungssignal an den empfangenden Kanal gesendet. Der sendende Kanal wartet die durch das Kanalattribut 'Intervall der Überwachungssignale' (HBINT) angegebene Anzahl an Sekunden auf eine Antwort vom empfangenden Kanal.

Der Wert wird in Millisekunden angegeben und muss im Bereich 0 bis 999999 liegen. Beim Wert 0 werden keine Überwachungssignale für Stapel verwendet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Stapelintervall (BATCHINT)

Dieses Attribut gibt an, wie lange (in Millisekunden) der Kanal einen Stapel geöffnet hält, auch wenn sich keine Nachrichten in der Übertragungswarteschlange befinden.

Sie können einen beliebigen Wert von 0 bis 999.999.999 angeben. Der Standardwert ist null.

Wenn Sie kein Stapelintervall angeben, wird der Stapel bei Eintreten einer der folgenden Bedingungen geschlossen:

- Die in BATCHSZ angegebene Anzahl von Nachrichten wurde gesendet.
- Die in BATCLIM angegebene Anzahl an Byte wurde gesendet.
- Die Übertragungswarteschlange ist leer.

Bei Kanälen mit geringer Auslastung und häufig leerer Übertragungswarteschlange kann die effektive Stapelgröße deutlich unter BATCHSZ liegen.

Mit dem Attribut BATCHINT können Sie die Effizienz der Kanäle steigern, indem Sie die Anzahl kurzer Stapel verringern. Beachten Sie jedoch, dass sich die Reaktionszeit verlängert, weil Stapel länger dauern und Nachrichten länger nicht festgeschrieben werden.

Wenn Sie BATCHINT angeben, werden Stapel nur bei einer der folgenden Bedingungen geschlossen:

- Die in BATCHSZ angegebene Anzahl von Nachrichten wurde gesendet.
- Die in BATCLIM angegebene Anzahl an Byte wurde gesendet.

- Es gibt keine weiteren Nachrichten in der Übertragungswarteschlange und das Zeitintervall von BATCHINT ist abgelaufen, während auf Nachrichten gewartet wurde (seit Abruf der ersten Nachricht des Stapels).

**Anmerkung:** BATCHINT gibt die gesamte Zeit an, in der auf Nachrichten gewartet wird. Darin ist nicht die Zeit für den Abruf von Nachrichten enthalten, die bereits in der Übertragungswarteschlange verfügbar sind, und nicht die Zeit, die für das Übertragen von Nachrichten benötigt wird.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Stapelatengrenzwert (BATCLIM)

Dieses Attribut gibt (in Kilobyte) das Datenvolumen an, das vor dem nächsten Synchronisationspunkt maximal über einen Kanal gesendet werden kann.

Ein Synchronisationspunkt wird erreicht, nachdem die Nachricht, mit der dieser Grenzwert erreicht wurde, vollständig über den Kanal übertragen wurde.

Der Wert muss zwischen 0 und 999999 liegen. Der Standardwert ist 5000.

Der Wert null für dieses Attribut bedeutet, dass es für Stapel auf diesem Kanal keinen Datengrenzwert gibt.

Der Stapel wird beendet, wenn eine der folgenden Bedingungen eintritt:

- BATCHSZ-Nachrichten wurden gesendet.
- BATCLIM-Bytes wurden gesendet.
- Die Übertragungswarteschlange ist leer und BATCHINT wurde überschritten.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Dieser Parameter wird auf allen Plattformen unterstützt.

## Stapelgröße (BATCSZ)

Dieses Attribut ist die maximale Anzahl der zu sendenden Nachrichten, bevor ein Synchronisationspunkt erstellt wird.

Die Stapelgröße wirkt sich nicht auf die Nachrichtenübertragungsmethode des Kanals aus. Nachrichten werden immer einzeln übertragen, aber als Stapel festgeschrieben oder zurückgesetzt.

Zur Leistungssteigerung können Sie eine Stapelgröße einstellen, die die maximale Anzahl der zwischen zwei *Synchronisationspunkten* zu übertragenden Nachrichten definiert. Die zu verwendende Stapelgröße wird beim Start eines Kanals vereinbart. Die kleinere der beiden Kanaldefinitionen wird verwendet. Bei einigen Implementierungen wird die Stapelgröße aus der kleinsten der beiden Kanaldefinitionen und den beiden Warteschlangenmanager-MAXUMSGS-Werten berechnet. Die tatsächliche Größe eines Stapels kann geringer ausfallen. Ein Stapel wird beispielsweise beendet, wenn es keine Nachrichten mehr in der Übertragungswarteschlange gibt oder das Stapelintervall abläuft.

Ein großer Wert für die Stapelgröße erhöht den Durchsatz, verlängert aber auch die Wiederherstellungszeiten, weil mehr Nachrichten zurückgesetzt und erneut gesendet werden müssen. Der Standardwert für BATCHSZ beträgt 50. Probieren Sie zuerst diesen Wert aus. Sie können einen niedrigeren Wert für

BATCHSZ wählen, wenn die Kommunikation störanfällig ist und damit die Wahrscheinlichkeit von Wiederherstellungen steigt.

Das Synchronisationspunktverfahren erfordert, dass beim Erstellen eines Synchronisationspunkts eine eindeutige ID der logischen Arbeitseinheit über die Verbindung ausgetauscht wird, um die Stapelfestschreibungsprozeduren zu koordinieren.

Wenn die synchronisierte Stapelfestschreibungsprozedur unterbrochen wird, kann eine *unbestätigte* Situation entstehen. Unbestätigte Situationen werden beim Starten eines Nachrichtenkanals automatisch aufgelöst. Wenn diese Auflösung nicht erfolgreich ist, kann ein manueller Eingriff mit dem Befehl RESOLVE erforderlich sein.

Einige Überlegungen zur Wahl der Stapelgröße:

- Wenn der Wert zu groß ist, wird auf beiden Seiten der Verbindung zu viel Warteschlangenspeicherplatz belegt. Nachrichten belegen Warteschlangenspeicherplatz, wenn sie nicht festgeschrieben werden, und können erst aus den Warteschlangen entfernt werden, wenn sie festgeschrieben wurden.
- Wenn ein kontinuierlicher Nachrichtenfluss zu erwarten ist, können Sie die Leistung eines Kanals durch Erhöhen der Stapelgröße verbessern, da dadurch weniger Bestätigungsflüsse für die Übertragung derselben Anzahl von Bytes erforderlich sind.
- Wenn die Merkmale des Nachrichtenflusses anzeigen, dass Nachrichten unregelmäßig eintreffen, kann die Stapelgröße 1 mit einem relativ großen Unterbrechungsintervall eine bessere Leistung erzielen.
- Der Wert kann im Bereich 1 bis 9999 liegen. Aus Gründen der Datenintegrität muss jedoch für Kanäle, die mit einer der aktuellen Plattformen verbunden sind, eine Stapelgröße von mehr als 1 angegeben werden. Der Wert 1 ist für Produkte der Version 1 vorgesehen (mit Ausnahme von IBM MQ for MVS).
- Auch wenn nicht persistente Nachrichten auf einem schnellen Kanal nicht auf einen Synchronisationspunkt warten, werden sie in die Stapelgröße eingerechnet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Zertifikatsbezeichnung (CERTLABL)

Dieses Attribut gibt die Zertifikatsbezeichnung der Kanaldefinition an.

Die Bezeichnung ermittelt, welches persönliche Zertifikat im Schlüsselrepository an den fernen Peer gesendet wird. Das Zertifikat wird wie unter [Digitale Zertifikatsbeschreibungen](#) beschrieben definiert.

Eingehende Kanäle (einschließlich RCVR-, RQSTR-, CLUSRCVR-, nicht qualifizierte SERVER- und SVRCONN-Kanäle) senden das konfigurierte Zertifikat nur, wenn die IBM MQ-Version des fernen Peers die Konfiguration der Zertifikatsbezeichnung vollständig unterstützt und der Kanal ein TLS-CipherSpec verwendet. Ist dies nicht der Fall, ermittelt das Attribut **CERTLABL** des Warteschlangenmanagers das gesendete Zertifikat. Diese Einschränkung besteht, da der Auswahlmechanismus für die Zertifikatsbezeichnung für eingehende Kanäle von einer TLS-Protokollerweiterung abhängt, die nicht in allen Fällen unterstützt wird. Insbesondere Java-Clients, JMS-Clients und alle Versionen von IBM MQ vor IBM MQ 8.0 unterstützen die erforderliche Protokollerweiterung nicht und empfangen unabhängig von der kanalspezifischen Bezeichnungseinstellung nur das vom Warteschlangenmanagerattribut **CERTLABL** konfigurierte Zertifikat.

Ein nicht qualifizierter Serverkanal ist ein Kanal, für den das Feld CONNAME nicht festgelegt wurde.

Dieses Attribut kann durch keine Verwaltungsschnittstelle abgefragt oder für CLUSSDR-Kanäle festgelegt werden. Sie erhalten eine MQRCCF\_WRONG\_CHANNEL\_TYPE-Nachricht. Das Attribut ist allerdings in CLUSSDR-Kanalobjekten vorhanden (einschließlich MQCD-Strukturen) und kann bei Bedarf durch einen CHAD-Exit programmgesteuert festgelegt werden.



Weitere Informationen zu den möglichen Inhalten von Zertifikatsbezeichnungen finden Sie im Abschnitt Digitale Zertifikatsbezeichnungen - Anforderungen.

Dieses Attribut ist für alle Kanaltypen gültig.

**Anmerkung:** Für SSL/TLS muss in der QMGR-Definition ein CERTLABL festgelegt sein. Sie können optional ein CERTLABL in der CHANNEL-Definition festlegen. Kanäle verwenden jedoch weiterhin den Warteschlangenmanagerstandard CERTLABL, auch wenn Sie CERTLABL als Kanalattribut definiert haben.

Das CERTLABL des Warteschlangenmanagers wird überprüft. Es muss ein gültiges persönliches Zertifikat sein, selbst wenn in der CHANNEL-Definition ein CERTLABL festgelegt ist.

Kanäle verwenden weiterhin den Standardwarteschlangenmanager CERTLABL, auch wenn Sie OPMODE im Modul CSQ6SYSP angegeben haben.

## Kanalname (CHANNEL)

Dieses Attribut gibt den Namen der Kanaldefinition an.

Der Name kann bis zu 20 Zeichen lang sein. Da jedoch beide Seiten eines Nachrichtenkanals denselben Namen haben müssen und bei anderen Implementierungen Größenbeschränkungen gelten können, muss die tatsächliche Zeichenzahl unter Umständen darunter liegen.

Kanalnamen dürfen auf zwei Warteschlangenmanagern in einem Netz miteinander verbundener Warteschlangenmanager nie doppelt vorkommen.

Der Name muss aus Zeichen der folgenden Liste bestehen:

Alphabetisch	(A-Z, a-z; Groß-/Kleinschreibung wird beachtet)
Ziffern	(0-9)
Punkt	(.)
Schrägstrich	(/)
Unterstrich	(_)
Prozentzeichen	(%)

### Anmerkung:

1. Eingebettete Leerzeichen sind nicht zulässig. Führende Leerzeichen werden ignoriert.
2. Auf Systemen, die EBCDIC Katakana verwenden, können keine Kleinbuchstaben verwendet werden.

Dieses Attribut ist für alle Kanaltypen gültig.

## Kanalstatistik (STATCHL) auf n Multiplatforms

Auf Multiplatforms steuert dieses Attribut die Erfassung von statistischen Daten für Kanäle.

Folgende Werte sind möglich:

### QMGR

Die Erfassung statistischer Daten für diesen Kanal basiert auf der Einstellung des Warteschlangenmanagerattributs STATCHL. Dies ist der Standardwert.

### OFF

Die Erfassung statistischer Daten für diesen Kanal wird inaktiviert.

### LOW


Die Erfassung statistischer Daten für diesen Kanal wird mit einer niedrigen Datenerfassungsrate aktiviert.

### MITTEL

Die Erfassung statistischer Daten für diesen Kanal wird mit einer mittleren Datenerfassungsrate aktiviert.

## HIGH

Die Erfassung statistischer Daten für diesen Kanal wird mit einer hohen Datenerfassungsrate aktiviert. Weitere Informationen zur Kanalstatistik finden Sie im Abschnitt [Referenzinformationen zur Überwachung](#).

 Auf z/OS -Systemen wird durch die Aktivierung dieses Parameters einfach die Erfassung statistischer Daten aktiviert, unabhängig vom ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Kanaltyp (CHLTYPE)

Dieses Attribut gibt den Typ des definierten Kanals an.

Mögliche Kanaltypen:

### Nachrichtenskanaltypen:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

### MQI-Kanaltypen:

- Clientverbindung (nur Windows und UNIX)

**Anmerkung:** Sie können Clientverbindungskanäle unter z/OS auch für die Verwendung auf anderen Plattformen definieren.

- Serververbindung
- AMQP

Beide Kanalseiten müssen denselben Namen und kompatible Typen haben:

- Sender mit Empfänger
- Requester mit Server
- Requester mit Sender (für Callback)
- Server mit Empfänger (Server wird als Sender verwendet)
- Clientverbindung mit Serververbindung
- Clustersender mit Clusterempfänger
- AMQP mit AMQP

## Clientkanalgewichtung (CLNTWGHT)

Dieses Attribut gibt eine Gewichtung an, durch die gesteuert wird, welche Clientverbindungskanaldefinition verwendet wird.

Mit dem Attribut CLNTWGHT können Clientkanaldefinitionen auf der Basis ihrer Gewichtung ausgewählt werden, wenn mehrere geeignete Definitionen zur Verfügung stehen.

Wenn ein Client MQCONN ausgibt, um eine Verbindung zu einer Warteschlangenmanagergruppe anzufordern, und dabei einen mit einem Stern beginnenden Warteschlangenmanagernamen angibt, wodurch die Clientgewichtung für mehrere Warteschlangenmanager ermöglicht wird, wird die zu verwendende Definition auf der Basis der Gewichtung ausgewählt, wenn die Definitionstabelle für Clientkanäle (CCDT - Client Channel Definition Table) mehrere geeignete Kanaldefinitionen enthält. Dabei werden gültige Definitionen des Typs CLNTWGHT(0) in alphabetischer Reihenfolge zuerst ausgewählt.

Geben Sie einen Wert im Bereich von 0 bis 99 an. Der Standardwert ist 0.

Der Wert 0 gibt an, dass kein Lastausgleich erfolgt und gültige Definitionen in alphabetischer Reihenfolge ausgewählt werden. Wenn der Lastausgleich aktiviert werden soll, wählen Sie einen Wert im Bereich von 1 bis 99 aus, wobei 1 der niedrigsten und 99 der höchsten Gewichtung entspricht. Die Aufteilung der Verbindungen zwischen zwei oder mehreren Kanälen mit einer Gewichtung ungleich null erfolgt in etwa proportional zum Verhältnis dieser Gewichtungen. Es werden beispielsweise drei Kanäle mit den CLNTWGHT-Werten 2, 4 und 14 zu rund 10 %, 20 % und 70 % der Zeit ausgewählt. Diese Verteilung ist nicht garantiert. Wenn das Attribut AFFINITY der Verbindung auf PREFERRED gesetzt ist, wird für die erste Verbindung eine Kanaldefinition entsprechend den Clientgewichtungen ausgewählt und für die nachfolgenden Verbindungen wird dieselbe Kanaldefinition verwendet.

Dieses Attribut ist nur für den Kanaltyp Clientverbindungskanal gültig.

## Cluster (CLUSTER)

Dieses Attribut ist der Name des Clusters, dem der Kanal zugeordnet ist.

Die maximale Länge beträgt 48 Zeichen gemäß den Regeln für die Benennung von IBM MQ-Objekten.

Maximal einer der resultierenden Werte von CLUSTER und CLUSNL kann andere Zeichen als nur Leerzeichen enthalten. Wenn einer der Werte andere Zeichen als nur Leerzeichen enthält, darf der andere Wert nur aus Leerzeichen bestehen.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

## Clusternamensliste (CLUSNL)

Dieses Attribut ist der Name der Namensliste mit den Clustern, zu denen der Kanal gehört.

Maximal einer der resultierenden Werte von CLUSTER und CLUSNL kann andere Zeichen als nur Leerzeichen enthalten. Wenn einer der Werte andere Zeichen als nur Leerzeichen enthält, darf der andere Wert nur aus Leerzeichen bestehen.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

## Kanalattribut CLWLPRTY

Das Kanalattribut CLWLPRTY legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Kanalattribut CLWLPRTY legen Sie die Reihenfolge der Priorität der verfügbaren Clusterziele fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Nachrichten gehen an den Warteschlangenmanager mit dem Kanal mit der höchsten Priorität. Ist dieser Kanal nicht

verfügbar, gehen die Nachrichten an den Warteschlangenmanager mit der nächsthöchsten Priorität. Warteschlangenmanager mit niedrigerer Priorität fungieren als Reserve.

Vor der Priorisierung der Kanäle prüft IBM MQ den Kanalstatus. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

#### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- 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 (Sicherheit) 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 Sicherheit.

### **Kanalattribut CLWLRANK**

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Verwenden Sie das Kanalattribut **CLWLRANK**, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Sie steuern die Auswahl des Zielorts, indem Sie den Rang der Kanäle festlegen, die einen Warteschlangenmanager am Schnittpunkt der Cluster mit den Gateway-Warteschlangenmanagern verbinden.

Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen. Ein Beispiel: Nachrichten kommen an einem Gateway-Warteschlangenmanager an, der sie an einen der beiden Warteschlangenmanager weiterleiten kann, die Kanäle mit dem Rang 1 und 2 verwenden. Die Nachrichten werden automatisch an den Warteschlangenmanager gesendet, der durch einen Kanal mit dem höchsten Rang verbunden ist; in diesem Fall ist dies der Warteschlangenmanagerkanal mit dem Rang 2.

IBM MQ ruft den Rang von Kanälen noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Kanäle zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

#### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Wenn Sie auch das Prioritätsattribut **CLWLPRTY** verwenden, wählt IBM MQ zwischen verfügbaren Zielen aus. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

### **Kanalattribut CLWLWGHT**

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von `CLUSDDR`- und `CLUSRCVR`-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

Verwenden Sie **CLWLWGHT**, um mehr Nachrichten an Server mit einer größeren Verarbeitungskapazität zu senden. Je stärker ein Kanal gewichtet ist, desto mehr Nachrichten werden über diesen Kanal versendet.

#### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Standardeinstellung des Parameters CLLWLGHT ist 50. Wird diese Einstellung für einen Kanal geändert, so wird der Lastausgleich abhängig von der Häufigkeit, die die einzelnen Kanäle für eine an eine Clusterwarteschlange gesendete Nachricht ausgewählt wurden. Weitere Informationen finden Sie unter „Algorithmus für das Clusterauslastungsmanagement“ auf Seite 176.

## Verbindungsaffinität (AFFINITY)

Dieses Attribut gibt an, ob Clientanwendungen, bei denen mehrfach Verbindungen mit dem gleichen Warteschlangenmanageramen hergestellt werden, denselben Clientkanal verwenden.

Verwenden Sie dieses Attribut, wenn mehrere gültige Kanaldefinitionen verfügbar sind.

Folgende Werte sind möglich:

### PREFERRED

Die erste Verbindung eines Prozesses, der eine Definitionstabelle für Clientkanäle (CCDT) liest, erstellt basierend auf der Clientkanalgewichtung eine Liste gültiger Definitionen, in der die Definitionen mit der Gewichtung 0 jeweils in alphabetischer Reihenfolge zuerst aufgeführt sind. Bei jeder Verbindung des Prozesses wird versucht, die Verbindung über die erste Definition der Liste herzustellen. Wenn eine Verbindung nicht erfolgreich ist, wird die nächste Definition verwendet. Nicht erfolgreiche Definitionen mit Clientkanalgewichtungswerten ungleich null werden an das Ende der Liste verschoben. Definitionen mit einer Clientkanalgewichtung von 0 verbleiben am Anfang der Liste und werden für jede Verbindung zuerst ausgewählt.

Jeder Clientprozess mit demselben Hostnamen erstellt immer dieselbe Liste.

Bei Clientanwendungen, die in C, C++ oder im .NET-Programmierframework geschrieben wurden (einschließlich vollständig verwalteter .NET-Clientanwendungen) und für Anwendungen, die IBM MQ classes for Java und IBM MQ classes for JMS verwenden, wird die Liste aktualisiert, wenn die Definitionstabelle für Clientkanäle (CCDT) seit der Erstellung der Liste geändert wurde.

Dies ist der Standardwert.

### KEINE

Die erste Verbindung eines Prozesses, die eine CCDT liest, erstellt eine Liste gültiger Definitionen. Alle Verbindungen eines Prozesses wählen eine gültige Definition basierend auf der Clientkanalgewichtung aus, wobei Definitionen mit der Gewichtung 0 in alphabetischer Reihenfolge zuerst ausgewählt werden.

Bei Clientanwendungen, die in C, C++ oder im .NET-Programmierframework geschrieben wurden (einschließlich vollständig verwalteter .NET-Clientanwendungen) und für Anwendungen, die IBM MQ classes for Java und IBM MQ classes for JMS verwenden, wird die Liste aktualisiert, wenn die Definitionstabelle für Clientkanäle (CCDT) seit der Erstellung der Liste geändert wurde.

Dieses Attribut ist nur für den Kanaltyp Clientverbindungskanal gültig.

## Verbindungsname (CONNAME)

Dieses Attribut ist die Kommunikationsverbindungs-ID. Es gibt die Kommunikationsverbindungen an, die von diesem Kanal verwendet werden sollen.

Es ist optional für Serverkanäle, sofern der Serverkanal nicht ausgelöst wird. In diesem Fall muss es einen Verbindungsnamen angeben.

Geben Sie für **CONNAME** eine durch Kommas getrennte Liste mit Computernamen für den angegebenen **TRPTYPE** ein. In der Regel ist nur ein Systemname erforderlich. Sie können mehrere Systemnamen angeben, um mehrere Verbindungen mit denselben Eigenschaften zu konfigurieren. Die Verbindungen werden normalerweise in der Reihenfolge getestet, in der sie in der Verbindungsliste angegeben sind, bis eine Verbindung erfolgreich eingerichtet werden konnte. Die Reihenfolge wird für Clients geändert, wenn das **CLNTWGH**-Attribut bereitgestellt wird. Falls keine Verbindung hergestellt werden kann, versucht der

Kanal, wie durch die Kanalattribute festgelegt, die Verarbeitung erneut. Bei Clientkanälen stellen Verbindungslisten eine Alternative zur Konfiguration mehrerer Verbindungen mithilfe von Warteschlangenmanagergruppen dar. Bei Nachrichtenkanälen wird eine Verbindungsliste zur Konfiguration von Verbindungen mit den Alternativadressen eines Multi-Instanz-Warteschlangenmanagers verwendet.

Die Bereitstellung von Namen für Mehrfachverbindungen in einer Liste wurde zuerst in IBM WebSphere MQ 7.0.1 unterstützt. Er ändert die Syntax des Parameters **CONNNAME**. Frühere Clients und Warteschlangenmanager stellen Verbindungen unter Verwendung des ersten Verbindungsnamens in der Liste her und lesen den Rest der Verbindungsnamen in der Liste nicht. Damit die früheren Clients und Warteschlangenmanager die neue Syntax analysieren können, müssen Sie für den ersten Verbindungsnamen in der Liste eine Portnummer angeben. Durch die Angabe einer Portnummer werden Probleme bei der Verbindungsherstellung von einem Client oder Warteschlangenmanager, der unter einer früheren Version als IBM WebSphere MQ 7.0.1 ausgeführt wird, mit einem Kanal vermieden.

**Multi** Unter [Multiplatforms](#) ist die Angabe des TCP/IP-Verbindungsnamensparameters eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM MQ automatisch einen Verbindungsnamen, wobei der Standardport vorausgesetzt und die aktuelle IP-Adresse des Systems verwendet wird. Sie können die Standardportnummer überschreiben, aber weiterhin die aktuelle IP-Adresse des Systems verwenden. Lassen Sie für jeden Verbindungsnamen den IP-Namen leer und übergeben Sie die Portnummer in runden Klammern; Beispiel:

(1415)

Der generierte **CONNNAME** wird nicht als alphanumerischer DNS-Hostname angegeben, sondern immer im dezimalen Format mit Punkten als Trennzeichen (IPv4) oder im hexadezimalen Format (IPv6).

Die maximale Länge des Namens ist von der Plattform abhängig:

- **Multi** 264 Zeichen.
- **z/OS** 48 Zeichen (siehe [Anmerkung 1](#)).

### Transporttyp TCP

CONNNAME ist der Hostname oder die Netzadresse der fernen Maschine (oder der lokalen Maschine bei Clusterempfängerkanälen). Beispiel: (ABC.EXAMPLE.COM), (2001:DB8:0:0:0:0:0:0) oder (127.0.0.1). Er kann die Portnummer enthalten, z. B. (MACHINE(123)).

**z/OS** Er kann den IP-Namen (IP\_name) einer dynamischen DNS-Gruppe oder einen Network Dispatcher-Eingabeport enthalten.

Wenn Sie eine IPv6-Adresse in einem Netz verwenden, das nur IPv4 unterstützt, wird der Verbindungsname nicht aufgelöst. In einem Netz, das IPv4 und IPv6 verwendet, bestimmt der Verbindungsname zusammen mit der lokalen Adresse, welcher IP-Stack verwendet wird. Weitere Informationen hierzu finden Sie im Abschnitt „[Lokale Adresse \(LOCLADDR\)](#)“ auf Seite 132.

### Transporttyp LU 6.2

**Windows** **IBM i** **UNIX** Wenn TPNAME und MODENAME angegeben sind, verwenden Sie den vollständig qualifizierten Namen der Partner-LU.

**Multi** Wenn TPNAME und MODENAME leer sind, geben Sie den Namen des CPI-C-Nebeninformationsobjekts für die bestimmte Plattform an.

**z/OS** Der Wert kann in zwei Formen angegeben werden:

- Name der logischen Einheit

Angaben zur logischen Einheit für den Warteschlangenmanager; diese setzen sich aus dem Namen der logischen Einheit, dem TP-Namen sowie (optional) dem Modusnamen zusammen. Dieser Name kann in einem von drei Formaten angegeben werden:

Format	Beispiel
LU-Name	IGY12355
LU-Name/TP-Name	IGY12345/APING
LU-Name/TP-Name/Modusname	IGY12345/APINGD/#INTER

Beim ersten Format müssen der Name des Transaktionsprogramms und der Modusname mit den Attributen TPNAME und MODENAME angegeben werden. Bei den beiden anderen Formaten müssen diese Attribute leer sein. Für Clientverbindungskanäle ist nur das erste Format erlaubt.

- Symbolischer Name

Symbolischer Bestimmungsname für die Angaben zur logischen Einheit für den Warteschlangenmanager, wie im Datensatz mit den Nebeninformationen definiert. Die Attribute TPNAME und MODENAME müssen leer sein. Beachten Sie, dass sich bei Clusterempfängerkanälen die Nebeninformationen in den anderen Warteschlangenmanagern des Clusters befinden. In diesem Fall kann es sich auch um einen Namen handeln, den ein Exit für die automatische Kanaldefinition in die entsprechenden LU-Informationen für den lokalen Warteschlangenmanager auflösen kann.

Der angegebene oder implizierte Name der logischen Einheit (LU) kann der LU-Name einer generischen VTAM-Ressourcengruppe sein.

### Übertragungsprotokoll NetBIOS

CONNNAME ist der in der fernen Maschine definierte NetBIOS-Name.

### Übertragungsprotokoll SPX

CONNNAME ist eine SPX-artige Adresse aus einer 4-Byte-Netzadresse, einer 6-Byte-Knotenadresse und einer 2-Byte-Socket-Nummer. Geben Sie diese Werte im Hexadezimalformat ein. Trennen Sie die Netz- und Knotenadresse durch einen Punkt, und setzen Sie die Socket-Nummer in eckige Klammern. Beispiel:

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

Wenn Sie die Socket-Nummer übergehen, wird die SPX-Standardsocketnummer von IBM MQ verwendet. Der Standardwert ist X'5E86'.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Es ist optional für Serverkanäle, sofern der Serverkanal nicht ausgelöst wird. In diesem Fall muss es einen Verbindungsnamen angeben.

### Anmerkung:

1. Bei der Länge des Namens können Sie die Begrenzung auf 48 Zeichen auf eine der folgenden Arten umgehen:
  - Richten Sie die DNS-Server z. B. so ein, dass Sie als Hostnamen "myserver" anstelle von "myserver.location.company.com" verwenden können.
  - Verwenden Sie IP-Adressen.
2. Die Definition des Übertragungsprotokolls finden Sie im Abschnitt „Transporttyp (TRPTYPE)“ auf Seite [151](#).

## Nachricht umwandeln (CONVERT)

Dieses Attribut gibt an, dass die Nachricht vor der Übertragung in das vom empfangenden System benötigte Format umgewandelt werden muss.

Anwendungsnachrichtendaten werden in der Regel von der empfangenden Anwendung umgewandelt. Wenn sich der ferne Warteschlangenmanager jedoch auf einer Plattform befindet, von der die Datenumwandlung nicht unterstützt wird, verwenden Sie dieses Kanalattribut, um anzugeben, dass die Nachricht **vor** der Übertragung in das Format umgewandelt werden muss, das für das empfangende System erforderlich ist.

Die gültigen Werte sind 'Ja' (yes) und 'Nein' (no). Wenn Sie 'Ja' (yes) angeben, werden die Anwendungsdaten in der Nachricht vor dem Senden umgewandelt, sofern Sie einen Namen der integrierten Formate angegeben haben oder ein Datenumwandlungsexit für ein benutzerdefiniertes Format verfügbar ist (siehe Abschnitt [Datenumwandlungsexits schreiben](#)). Wenn Sie Nein angeben, werden die Anwendungsdaten in der Nachricht vor dem Senden nicht umgewandelt.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Datenkomprimierung (COMPMSG)

Dieses Attribut ist eine Liste mit den Komprimierungsverfahren für Nachrichtendaten, die vom Kanal unterstützt werden.

Bei Sender-, Server-, Clustersender-, Clusterempfänger- und Clientverbindungskanälen werden die angegebenen Werte in der bevorzugten Reihenfolge aufgelistet. Das erste vom fernen Ende des verwendeten Kanals unterstützte Komprimierungsverfahren wird verwendet. Die von den Kanälen unterstützten Komprimierungsverfahren werden an den Nachrichtenexit des sendenden Kanals übergeben, von dem aus das verwendete Komprimierungsverfahren für jede einzelne Nachricht geändert werden kann. Durch die Komprimierung werden die Daten geändert, die an den Sende- bzw. Empfangsexit übergeben werden. Details zur Komprimierung des Nachrichtenheaders finden Sie im Abschnitt [„Headerkomprimierung \(COMPHDR\)“](#) auf Seite 130.

Folgende Werte sind möglich:

### **KEINE**

Es werden keine Nachrichtendaten komprimiert. Dies ist der Standardwert.

### **RLE**

Nachrichtendaten werden mittels Lauflängencodierung komprimiert.

### **ZLIBFAST**

Die Komprimierung der Nachrichtendaten erfolgt unter Verwendung der ZLIB-Komprimierungstechnik. Dabei wird eine kurze Komprimierungszeit bevorzugt.

ZLIBFAST kann optional durch die zEnterprise-Datenkomprimierungsfunktion entlastet werden. Weitere Informationen finden Sie unter [zEDC Express-Funktion](#).

### **ZLIBHIGH**

Die Komprimierung der Nachrichtendaten erfolgt unter Verwendung der ZLIB-Komprimierungstechnik. Dabei wird eine hohe Komprimierungsstufe bevorzugt.

### **ANY**

Ermöglicht dem Kanal die Unterstützung aller Komprimierungstechniken, die auch der Warteschlangenmanager unterstützt. Wird nur für Empfänger-, Requester- und Serververbindungskanäle unterstützt.

Dieses Attribut ist für alle Kanaltypen gültig.



## Standardmäßige Verbindungswiederherstellung (DEFRECON)

Gibt an, ob eine Clientverbindung automatisch die Verbindung zu einer Clientanwendung wiederherstellt, wenn die Verbindung unterbrochen wird.

Folgende Werte sind möglich:

### NO

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

### YES

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

### QMGR

Wenn nicht durch **MQCONN** 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 **MQCONN** angefordert wird.

Dieses Attribut ist nur für Clientverbindungskanäle gültig.

## Beschreibung (DESCR)

Dieses Attribut beschreibt die Kanaldefinition und enthält bis zu 64 Byte Text.

**Anmerkung:** Die maximale Anzahl von Zeichen nimmt ab, wenn das System einen Doppelbytezeichensatz verwendet.

Stellen Sie mithilfe von Zeichen aus dem Zeichensatz mit der CCSID des Warteschlangenmanagers sicher, dass der Text ordnungsgemäß umgesetzt wird, wenn er an einen anderen Warteschlangenmanager gesendet wird.

Dieses Attribut ist für alle Kanaltypen gültig.

## Unterbrechungsintervall (DISCINT)

Dieses Attribut definiert die Zeit, nach der ein Kanal geschlossen wird, wenn in diesem Zeitraum keine Nachricht mehr eintrifft.

Dieses Attribut ist ein Zeitlimit (in Sekunden) für die Server, Clustersender-, Sender- und Clusterempfängerkanäle. Das Intervall setzt ein, wenn ein Stapel endet, d. h., wenn die Stapelgröße erreicht wird oder das Stapelintervall abläuft und die Übertragungswarteschlange leer ist. Wenn während des angegebenen Zeitintervalls keine Nachrichten in der Übertragungswarteschlange eintreffen, wird der Kanal geschlossen. (Die Zeitangabe ist ein Näherungswert.)

Der Austausch von Steuerdaten zwischen den beiden Kanalseiten im Rahmen des Systemabschlusses beinhaltet eine Angabe über den Grund für die Schließung. Damit wird sichergestellt, dass die entsprechende Seite des Kanals für einen Neustart verfügbar bleibt.

Sie können eine beliebige Zeit (in Sekunden) von 0 bis 999.999 festlegen. Der Wert 0 bedeutet, dass es keine Unterbrechung gibt und unendlich lange gewartet wird.

Bei Serververbindungskanälen, die das TCP-Protokoll verwenden, gibt das Intervall die Zeit in Sekunden an, bis die Verbindung bei Inaktivität des Clients getrennt wird. Wenn eine Serververbindung für diese Dauer keine Kommunikation von ihrem Partnerclient erhalten hat, beendet sie die Verbindung.

Das Inaktivitätsintervall für Serververbindung gilt zwischen IBM MQ -API-Aufrufen von einem Client.

**Anmerkung:** Ein MQGET-Aufruf mit Wartezeit mit potenziell langer Laufzeit wird nicht als Inaktivität klassifiziert und überschreitet daher nie das Zeitlimit, da DISCINT abläuft.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Serververbindung
- Clustersender
- Clusterempfänger

Bei Serververbindungskanälen, die ein anderes Protokoll als TCP verwenden, wird dieses Attribut ignoriert.

**Anmerkung:** Der Wert für das Unterbrechungsintervall wirkt sich auf die Leistung aus.

Ein niedriger Wert (z. B. wenige Sekunden) kann der Systemleistung abträglich sein, weil der Kanal immer wieder gestartet wird. Ein großer Wert (mehr als eine Stunde) kann bedeuten, dass Systemressourcen unnötig lange belegt werden. Sie können auch ein Intervall der Überwachungssignale angeben. Wenn sich keine Nachrichten in der Übertragungswarteschlange befinden, sendet der sendende MCA einen Austausch von Überwachungssignalen an den empfangenden MCA. Dann kann der empfangende MCA den Kanal in den Quiescemodus versetzen, ohne auf den Ablauf des Unterbrechungsintervalls warten zu müssen. Damit diese beiden Werte effektiv zusammenarbeiten können, muss der Wert des Intervalls der Überwachungssignale deutlich niedriger als der Wert des Unterbrechungsintervalls sein.

Der Standardwert für DISCINT beträgt 100 Minuten. Oft ist jedoch ein Wert von wenigen Minuten sinnvoll, um die Leistung nicht zu beeinträchtigen oder um zu verhindern, dass Kanäle über unnötig lange Zeiträume aktiv bleiben. Sofern für Ihre Umgebung geeignet, können Sie diesen Wert ändern, entweder für jeden einzelnen Kanal oder durch Änderung des Werts in den Standardkanaldefinitionen, z. B. SYSTEM.DEF.SENDER.

Weitere Informationen finden Sie im Abschnitt [Kanäle anhalten und stilllegen](#).

## Disposition (QSGDISP)

Dieses Attribut gibt die Disposition des Kanals in einer Gruppe mit gemeinsamer Warteschlange an. Dieser Parameter ist nur unter z/OS gültig.

Folgende Werte sind möglich:

### QMGR

Der Kanal wird in der Seitengruppe des Warteschlangenmanagers definiert, der den Befehl ausführt. Dieser Wert ist der Standardwert.

### GRUPPE

Der Kanal befindet sich im gemeinsamen Repository. Dieser Wert ist nur in einer Umgebung mit gemeinsamer Warteschlange zulässig. Wenn ein Kanal mit QSGDISP(GROUP) definiert ist, wird der Befehl DEFINE CHANNEL(name) NOREPLACE QSGDISP(COPY) automatisch generiert und an alle aktiven Warteschlangenmanager gesendet, damit diese lokale Kopien in der Seitengruppe 0 erstellen. Bei Warteschlangenmanagern, die nicht aktiv sind oder die der Gruppe mit gemeinsamer Warteschlange zu einem späteren Zeitpunkt beitreten, wird der Befehl beim Starten des Warteschlangenmanagers generiert.

### KOPIEREN

Der Kanal wird in der Seitengruppe des Warteschlangenmanagers definiert, der den Befehl ausführt. Dabei wird die Definition vom Kanal QSGDISP(GROUP) mit demselben Namen kopiert. Dieser Wert ist nur in einer Umgebung mit gemeinsamer Warteschlange zulässig.

Dieses Attribut ist für alle Kanaltypen gültig.

## Headerkomprimierung (COMPHDR)

Dieses Attribut ist eine Liste mit den Komprimierungsverfahren für Headerdaten, die vom Kanal unterstützt werden.

Für Sender-, Server-, Clustersender-, Clusterempfänger- und Clientverbindungskanäle werden die Werte in der bevorzugten Reihenfolge angegeben. Dabei wird das erste Komprimierungsverfahren verwendet, das vom fernen Ende des verwendeten Kanals unterstützt wird. Die von den Kanälen unterstützten

Komprimierungsverfahren werden an den Nachrichtenexit des sendenden Kanals übergeben, von dem aus das verwendete Komprimierungsverfahren für jede einzelne Nachricht geändert werden kann. Durch die Komprimierung werden die Daten geändert, die an den Sende- bzw. Empfangsexit übergeben werden.

Mögliche Werte:

**KEINE**

Es werden keine Headerdaten komprimiert. Dies ist der Standardwert.

**SYSTEM**

Headerdaten werden komprimiert.

Dieses Attribut ist für alle Kanaltypen gültig.

## **Intervall der Überwachungssignale (HBINT)**

Dieses Attribut gibt die geschätzte Zeit zwischen zwei Überwachungssignalen an, die der sendende Nachrichtenkanalagent (Message Channel Agent, MCA) überbrücken muss, wenn keine Nachrichten in der Übertragungswarteschlange vorhanden sind.

Der Austausch von Überwachungssignalen hebt die Blockierung des empfangenden Nachrichtenkanalagenten auf, der auf eingehende Nachrichten wartet, bzw. darauf, dass das Unterbrechungsintervall abläuft. Wenn der empfangende MCA entblockt ist, kann er die Kanalverbindung trennen, ohne den Ablauf des Unterbrechungsintervalls abzuwarten. Durch den Austausch von Überwachungssignalen werden auch die für große Nachrichten zugewiesenen Speicherpuffer freigegeben und Warteschlangen geschlossen, die auf der Empfangsseite des Kanals offen gelassen wurden.

Der Wert wird in Sekunden angegeben und muss zwischen 0 und 999999 liegen. Bei Angabe von Null werden keine Überwachungssignale gesendet. Der Standardwert ist 300. Dieser Wert ist nur sinnvoll, wenn er deutlich kleiner ist als das Unterbrechungsintervall.

Bei Anwendungen, die IBM MQ classes for Java, JMS oder .NET-APIs verwenden, wird der HBINT-Wert auf eine der folgenden Arten bestimmt:

- Entweder durch den Wert im SVRCONN-Kanal, der von der Anwendung verwendet wird.
- Oder durch den Wert im CLNTCONN-Kanal, wenn die Anwendung so konfiguriert wurde, dass sie eine CCDT verwendet.

Bei Serververbindungs- und Clientverbindungskanälen können Überwachungssignale unabhängig voneinander sowohl vom Server als auch vom Client gesendet werden. Wenn innerhalb des Überwachungssignalintervalls keine Daten über den Kanal übertragen wurden, sendet der MQI-Agent der Clientverbindung ein Überwachungssignal, das vom MQI-Agenten der Serververbindung durch ein weiteres Überwachungssignal beantwortet wird. Dies geschieht unabhängig vom Status des Kanals, also zum Beispiel unabhängig davon, ob der Kanal während eines API-Aufrufs oder während des Wartens auf Benutzereingaben vom Client inaktiv ist. Zudem kann der MQI-Agent der Serververbindung den Austausch von Überwachungssignalen mit dem Client initialisieren, und zwar auch hier unabhängig vom Status des Kanals. Um zu verhindern, dass der MQI-Agent der Serververbindung und der MQI-Agent der Clientverbindung gleichzeitig an die jeweils andere Seite Überwachungssignale abgeben, wird das Überwachungssignal des Servers erst ausgegeben, wenn über den Kanal für die Länge des Überwachungssignalintervalls plus 5 Sekunden keine Daten mehr übertragen wurden.

Bei Serververbindungs- und Clientverbindungskanälen im Kanalmodus einer früheren Version von IBM WebSphere MQ 7.0 findet der Austausch von Überwachungssignalen nur statt, wenn ein Servernachrichtenkanalagent auf einen MQGET-Befehl mit angegebener WAIT-Option wartet, den er für eine Clientanwendung ausgegeben hat.

Weitere Informationen dazu, wie MQI-Kanäle in beiden Modi arbeiten, finden Sie im Abschnitt [Sharing-Conversations \(MQLONG\)](#).

### **Zugehörige Informationen**

[CHANNEL DEFINE CHANNEL](#)

[ALTER CHANNEL](#)

## Keepalive-Intervall (KAINT)

Mit diesem Attribut wird ein Zeitlimit für einen Kanal angegeben.

Dieser Wert wird an den Kommunikationsstack übermittelt und gibt das Keepalive-Timing für den Kanal an. Das Attribut ermöglicht die Angabe unterschiedlicher Keepalive-Werte für jeden Kanal.

Sie können das Attribut KAINTE für jeden Kanal einzeln angeben.

**Multi** Auf Multiplatforms können Sie auf den Parameter zugreifen und ihn ändern. Er wird jedoch nur gespeichert und weitergeleitet, eine funktionale Implementierung des Parameters findet nicht statt. Wenn Sie die Funktionalität des Parameters KAINTE benötigen, verwenden Sie den Parameter für das Intervall der Überwachungssignale (HBINT) gemäß Abschnitt „Intervall der Überwachungssignale (HBINT)“ auf Seite 131.

Damit dieses Attribut wirksam wird, muss die TCP/IP-Keepalive-Funktion aktiviert sein. Unter z/OS aktivieren Sie Keepalive, indem Sie den MQSC-Befehl ALTER QMGR TCPKEEP(YES) ausgeben. Auf Multiplatforms wird die TCP/IP-Keepalive-Funktion aktiviert, indem der Parameter KEEPALIVE in der TCP-Zeitengruppe der Konfigurationsdatei für verteilte Warteschlangensteuerung (qm.ini) auf YES gesetzt wird (KEEPALIVE=YES), oder sie wird über den IBM MQ Explorer aktiviert. Keepalive muss mithilfe des Datenbestands für die TCP-Profilkonfiguration auch in TCP/IP selbst aktiviert werden.

Der Wert gibt einen Zeitraum (in Sekunden) an und muss im Bereich 0 bis 99999 liegen. Der Keepalive-Intervallwert 0 zeigt an, dass kein kanalspezifisches Keepalive für den Kanal aktiviert ist und nur der systemweite in TCP/IP eingestellte Keepalive-Wert verwendet wird. Sie können KAINTE auch auf AUTO festlegen. (Dies ist der Standardwert.) Dann basiert der Keepalive-Wert wie folgt auf dem vereinbarten Wert des Intervalls der Überwachungssignale (HBINT):

Vereinbarter HBINT-Wert	KAINTE
>0	Vereinbarter HBINT-Wert + 60 Sekunden
0	0

Dieses Attribut ist für alle Kanaltypen gültig.

Der Wert wird für alle Kanäle ignoriert, bei denen TransportType (TRPTYPE) eine andere Einstellung als TCP oder SPX besitzt.

## Lokale Adresse (LOCLADDR)

Dieses Attribut gibt die lokale Kommunikationsadresse für den Kanal an.

**Anmerkung:** AMQP-Kanäle unterstützen nicht dasselbe Format von LOCLADDR wie andere IBM MQ-Kanäle. Weitere Informationen finden Sie im Abschnitt „LOCLADDR für AMQP-Kanäle“ auf Seite 135.

## LOCLADDR für alle Kanäle außer AMQP-Kanälen

Dieses Attribut gilt nur, wenn TCP/IP als Transporttyp (TRPTYPE) verwendet wird. Bei allen anderen Transporttypen wird es ignoriert.

Wenn ein LOCLADDR-Wert angegeben wird, verwendet ein Kanal, der gestoppt und anschließend erneut gestartet wird, die in LOCLADDR angegebene TCP/IP-Adresse weiter. In Wiederherstellungsszenarios kann dieses Attribut nützlich sein, wenn der Kanal über eine Firewall kommuniziert. Es behebt Probleme, die dadurch entstehen, dass der Kanal mit der IP-Adresse des TCP/IP-Stacks erneut gestartet wird, mit dem er verbunden ist. LOCLADDR kann auch erzwingen, dass ein Kanal einen IPv4 -oder IPv6 -Stack auf einem Dual-Stack-System oder einen Dual-Mode-Stack auf einem Single-Stack-System verwendet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Wenn LOCLADDR eine Netzadresse enthält, muss diese zu einer Netzchnittstelle des Systems gehören, auf dem der Kanal ausgeführt wird. Für die Definition eines Senderkanals im Warteschlangenmanager ALPHA auf den Warteschlangenmanager BETA verwenden Sie beispielsweise den folgenden MSQC-Befehl:

```
DEFINE CHANNEL(TO.BETA) CHLTYPE(SDR) CONNAME(192.0.2.0) XMITQ(BETA) LOCLADDR(192.0.2.1)
```

Die LOCLADDR-Adresse ist die IPv4-Adresse 192.0.2.1. Dieser Senderkanal wird auf dem System des Warteschlangenmanagers ALPHA ausgeführt. Deshalb muss die IPv4-Adresse zu einer der Netzchnittstellen seines Systems gehören.

Der Wert ist die optionale IP-Adresse und der optionale Port bzw. Portbereich für die abgehende TCP/IP-Kommunikation. Diese Informationen werden im folgenden Format angegeben:

```
LOCLADDR([ip-addr][(low-port[,high-port])][, [ip-addr][(low-port[,high-port])]])
```

Die maximale Länge von **LOCLADDR**, einschließlich mehrerer Adressen, ist MQ\_LOCAL\_ADDRESS\_LENGTH.

Wenn Sie **LOCLADDR** weglassen, wird automatisch eine lokale Adresse zugeordnet.

Beachten Sie, dass Sie **LOCLADDR** für einen C-Client über die Definitionstabelle für Clientkanäle (CCDT) festlegen können.

Alle Parameter sind optional. Durch Übergehen des ip-addr-Teils der Adresse wird die Konfiguration einer festen Portnummer für eine IP-Firewall aktiviert. Das Übergehen der Portnummer ist hilfreich, um einen bestimmten Netzadapter auszuwählen, ohne eine eindeutige, lokale Portnummer identifizieren zu müssen. Der TCP/IP-Stack generiert eine eindeutige Portnummer.

Geben Sie [, [ip-addr][(low-port[,high-port])]] für jede zusätzliche lokale Adresse mehrmals an. Verwenden Sie mehrere lokale Adressen, wenn Sie eine bestimmte Untergruppe von lokalen Netzadaptern angeben möchten. Sie können auch [, [ip-addr][(low-port[,high-port])]] verwenden, um eine bestimmte lokale Netzadresse auf verschiedenen Servern darzustellen, die Teil einer Multi-Instanz-Warteschlangenmanagerkonfiguration sind.

#### ip-addr

ip-addr (IP-Adresse) wird in einem von drei Formaten angegeben:

##### IPv4-Dezimalschreibweise mit Punkten

Beispiel: 192.0.2.1

##### IPv6-Hexadezimalschreibweise

Beispiel: 2001:DB8:0:0:0:0:0:0

##### Alphanumerisches Hostnamensformat

Beispiel: WWW.EXAMPLE.COM

#### low-port und high-port

low-port (niedrigster\_Port) und high-port (höchster\_Port) sind Portnummern in runden Klammern.

Die folgende Tabelle zeigt, wie der Parameter **LOCLADDR** verwendet werden kann:

<i>Tabelle 30. Beispiele für Verwendung des Parameters <b>LOCLADDR</b></i>	
<b>LOCLADDR</b>	<b>Bedeutet</b>
9.20.4.98	Kanal wird lokal an diese Adresse gebunden.

Tabelle 30. Beispiele für Verwendung des Parameters **LOCLADDR** (Forts.)

LOCLADDR	Bedeutet
9.20.4.98, 9.20.4.99	Kanal wird an beide IP-Adressen gebunden. Bei der Adresse kann es sich um zwei Netzadapter auf einem einzigen Server oder um einen anderen Netzadapter auf zwei verschiedenen Servern in einer Mehrinstanzkonfiguration handeln.
9.20.4.98(1000)	Kanal wird lokal an diese Adresse und an Port 1000 gebunden.
9.20.4.98(1000,2000)	Lokale Kanalbindung an diese Adresse und den Portbereich 1000 bis 2000
(1000)	Kanal wird lokal an Port 1000 gebunden.
(1000,2000)	Kanal wird lokal an einen Port im Bereich von 1000 bis 2000 gebunden.

Beim Start eines Kanals bestimmen die Werte für den Verbindungsnamen (CONNAME) und die lokale Adresse (LOCLADDR), welcher IP-Stack für die Kommunikation verwendet wird. Der verwendete IP-Stack wird wie folgt bestimmt:

- Wenn für das System nur ein IPv4-Stack konfiguriert ist, wird immer der IPv4-Stack verwendet. Wenn Sie eine lokale Adresse (LOCLADDR) oder einen Verbindungsnamen (CONNAME) als IPv6-Netzadresse angeben, wird ein Fehler generiert und der Start des Kanals schlägt fehl.
- Wenn für das System nur ein IPv6-Stack konfiguriert ist, wird immer der IPv6-Stack verwendet. Wenn eine lokale Adresse (LOCLADDR) als IPv4 -Netzadresse angegeben wird, wird ein Fehler generiert und der Kanal kann nicht gestartet werden. Wenn Sie auf Plattformen, die die IPv6-Adresszuordnung unterstützen, einen Verbindungsnamen (CONNAME) als IPv4-Netzadresse angeben, wird die Adresse einer IPv6-Adresse zugeordnet. xxx . xxx . xxx . xxx wird beispielsweise : : ffff : xxx . xxx . xxx . xxx zugeordnet. Für die Verwendung zugeordneter Adressen werden unter Umständen Protokollumsetzungsprogramme benötigt. Vermeiden Sie die Verwendung zugeordneter Adressen, falls möglich.
- Wenn Sie eine lokale Adresse (LOCLADDR) als IP-Adresse für einen Kanal angeben, wird der Stack für diese IP-Adresse verwendet. Wenn Sie die lokale Adresse (LOCLADDR) als Hostname angeben, der in IPv4- und IPv6-Adressen aufgelöst wird, bestimmt der Verbindungsname (CONNAME), welcher der Stacks verwendet wird. Wenn die lokale Adresse (LOCLADDR) und der Verbindungsname (CONNAME) als Hostnamen angegeben werden, die in IPv4 -und IPv6 -Adressen aufgelöst werden, wird der verwendete Stack durch das Warteschlangenmanagerattribut IPADDRV bestimmt.
- Wenn auf dem System zwei IPv4 -und IPv6 -Stacks konfiguriert sind und keine lokale Adresse (LOCLADDR) für einen Kanal angegeben ist, bestimmt der für den Kanal angegebene Verbindungsname (CONNAME), welcher IP-Stack verwendet werden soll. Wenn der Verbindungsname (CONNAME) als Hostname angegeben wird, der in IPv4 -und IPv6 -Adressen aufgelöst wird, wird der verwendete Stack durch das Warteschlangenmanagerattribut IPADDRV bestimmt.

**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.

## Lokale Adressen für Clustersenderkanäle

Clustersenderkanäle übernehmen immer die im Zielwarteschlangenmanager definierte Konfiguration des entsprechenden Clusterempfängerkanals. Dies gilt auch dann, wenn ein lokal definierter Clustersenderkanal mit demselben Namen vorhanden ist; in diesem Fall wird die manuelle Definition nur für die Eingangskommunikation verwendet.

Aus diesem Grund sollte man sich nicht auf die im Clusterempfängerkanal definierte lokale Adresse (LOCLADDR) verlassen, da die IP-Adresse wahrscheinlich nicht dem System gehört, auf dem die Clustersender erstellt werden. Aus diesem Grund sollte die lokale Adresse (LOCLADDR) auf dem Clusterempfänger nur verwendet werden, wenn es einen Grund gibt, nur die Ports, nicht jedoch die IP-Adresse für alle potenz-

iellen Clusterabsender zu beschränken, und diese Ports bekanntermaßen auf allen Systemen verfügbar sind, auf denen möglicherweise ein Clustersenderkanal erstellt wird.

Wenn ein Cluster das Attribut LOCLADDR verwenden muss, um Kanäle für die abgehende Kommunikation an eine bestimmte IP-Adresse zu binden, sollte nach Möglichkeit der Exit für die automatische Kanaldefinition oder die standardmäßige lokale Adresse (LOCLADDR) für den Warteschlangenmanager verwendet werden. Bei Verwendung eines Kanalexits wird die Verwendung des Werts für LOCLADDR aus dem Exit in den automatisch definierten CLUSSDR-Kanälen erzwungen.

Bei Verwendung einer lokalen Adresse (LOCLADDR) für Clustersenderkanäle über einen Exit, bei der es sich nicht um einen Standardwert handelt, oder bei Verwendung eines Standardwerts muss in allen entsprechenden manuell definierten Clustersenderkanälen (beispielsweise zu einem Warteschlangenmanager mit vollständigem Repository) ebenfalls der Wert des Attributs LOCLADDR gesetzt sein, damit die Eingangskommunikation über den Kanal möglich ist.

**Anmerkung:** Wenn das Betriebssystem für den in LOCLADDR bereitgestellten Port (oder für alle Ports, falls ein Portbereich angegeben ist) einen Bindungsfehler zurückgibt, wird der Kanal nicht gestartet; das System gibt eine Fehlernachricht aus.

## LOCLADDR für AMQP-Kanäle

AMQP-Kanäle unterstützen ein anderes Format von LOCLADDR als andere IBM MQ-Kanäle:

### LOCLADDR ( *ip-addr* )

LOCLADDR ist die lokale Kommunikationsadresse für den Kanal. Verwenden Sie diesen Parameter, um das Verwenden einer bestimmten IP-Adresse durch den Client zu erzwingen. LOCLADDR ist auch nützlich, um einen Kanal zu zwingen, eine IPv4 -oder IPv6 -Adresse zu verwenden, wenn eine Auswahl verfügbar ist, oder um einen bestimmten Netzadapter auf einem System mit mehreren Netzadaptern zu verwenden.

Die maximale Länge von LOCLADDR ist MQ\_LOCAL\_ADDRESS\_LENGTH.

Wenn Sie LOCLADDR weglassen, wird automatisch eine lokale Adresse zugeordnet.

#### **ip-addr**

*ip-addr* ist eine einzelne Netzadresse, die in einem von drei Formaten angegeben wird:

#### **IPv4-Dezimalschreibweise mit Punkten**

Beispiel: 192.0.2.1

#### **IPv6-Hexadezimalschreibweise**

Beispiel: 2001:DB8:0:0:0:0:0:0

#### **Alphanumerisches Hostnamensformat**

Beispiel: WWW.EXAMPLE.COM

Bei Angabe einer IP-Adresse wird nur das Adressformat überprüft. Eine Überprüfung der eigentlichen IP-Adresse findet nicht statt.

## Zugehörige Informationen

Mit automatisch definierten Clustersenderkanälen arbeiten

## Zähler für Wiederholungsversuche nach langem Intervall (LONGRTY)

Dieses Attribut gibt die maximale Anzahl an Wiederholungsversuchen des Kanals an, seinem Partner eine Sitzung zuzuordnen.

Das Attribut **long retry count** kann zwischen 0 und 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender

- Clusterempfänger

Wenn der erste Zuordnungsversuch fehlschlägt, wird die *short retry count*-Zahl verringert und der Kanal wiederholt die verbleibende Anzahl von Versuchen. Wenn er immer noch fehlschlägt, wiederholt er *long retry count* so oft mit einem Intervall von *long retry interval* zwischen den einzelnen Versuchen. Kann die Operation weiterhin nicht ausgeführt werden, wird der Kanal geschlossen. Der Kanal muss anschließend mit einem Befehl neu gestartet werden; er wird nicht automatisch durch den Kanalinitiator gestartet.

Unter z/OS kann ein Kanal keine Wiederholungen durchführen, wenn die maximale Anzahl an Kanälen (**MAXCHL**) überschritten wurde.

Auf IBM i-, UNIX- und Windows-Systemen wird nur eine Wiederholung versucht, wenn ein Kanalinitiator ausgeführt wird. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

Wenn der Kanalinitiator (unter z/OS) bzw. der Kanal (auf Multiplatforms) während eines Wiederholungsversuchs gestoppt wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* zurückgesetzt, wenn der Kanalinitiator bzw. der Kanal neu gestartet wird oder wenn eine Nachricht erfolgreich über den Senderkanal versendet wird. Wenn jedoch der Kanalinitiator (unter z/OS) bzw. der Warteschlangenmanager (auf Multiplatforms) beendet und erneut gestartet wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht zurückgesetzt. Der Kanal behält die Werte des Zählers für Wiederholungsversuche bei, die vor dem Neustart des Warteschlangenmanagers bzw. vor dem Versenden der Nachricht galten.

Auf IBM i-Systemen, UNIXn und Windows-Systemen:

1. Wenn der Status des Kanals von RE TRYING zu RUNNING wechselt, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht unverzüglich zurückgesetzt. Die Zähler werden erst zurückgesetzt, wenn die erste Nachricht erfolgreich über den Kanal versendet wird, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., wenn der lokale Kanal die Anzahl der versendeten Nachrichten bestätigt hat.
2. Der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* werden beim Neustart des Kanals zurückgesetzt.

## Intervall für lange Wiederholungsversuche (LONGTMR)

Dieses Attribut ist die ungefähre Länge des Zeitintervalls in Sekunden, das der Kanal bis zum erneuten Versuch eines Verbindungsaufbaus abwartet, während der Modus für lange Wiederholungsversuche aktiviert ist.

Das Intervall zwischen den Verbindungsversuchen kann erhöht werden, wenn der Kanal abwarten muss, bis er aktiv ist.

Der Kanal versucht, so oft, wie durch den *Zähler für lange Wiederholungsversuche* für dieses lange Intervall festgelegt, eine Verbindung herzustellen, nachdem er so oft, wie durch den *Zähler für kurze Wiederholungsversuche* festgelegt, mit dem Warteintervall für kurze Wiederholungsversuche eine Verbindung herzustellen.

Dieses Attribut kann auf einen Wert von null bis 999 999 festgelegt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Name des LU 6.2-Modus (MODENAME)

Dieses Attribut wird bei LU 6.2-Verbindungen verwendet. Es handelt sich um eine zusätzliche Angabe bei den Sitzungsmerkmalen der Verbindung, wenn die Zuordnung einer Kommunikationssitzung erfolgt.



Bei Verwendung der Nebeninformationen für die SNA-Kommunikation ist der Modusname im Nebenobjekt der CPI-C-Kommunikation oder in den APPC-Nebeninformationen definiert. Dieses Attribut muss leer bleiben bzw. auf den SNA-Modusnamen gesetzt sein.

Der Name muss ein bis acht alphanumerische Zeichen lang sein.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Er gilt nicht für Empfänger- oder Serververbindungskanäle.

## **LU 6.2-Transaktionsprogrammname (TPNAME)**

Dieses Attribut wird bei LU 6.2-Verbindungen verwendet. Es handelt sich um den Namen bzw. den generischen Namen des Transaktionsprogramms (MCA), das am fernen Ende der Verbindung ausgeführt wird.

Bei Verwendung der Nebeninformationen für die SNA-Kommunikation ist der Transaktionsprogrammname im CPI-C-Nebenobjekt für Kommunikation oder in den APPC-Nebeninformationen definiert, und dieses Attribut muss leer bleiben. Andernfalls wird dieser Name von den Sender- und Requesterkanälen benötigt.

Der Name kann bis zu 64 Zeichen lang sein.

Der Name muss auf den SNA-Transaktionsprogrammnamen festgelegt sein, es sei denn, der CONNAME enthält einen Nebenobjektnamen, dann muss der Name leer bleiben. Der eigentliche Name wird stattdessen dem CPI-C-Nebenobjekt für Kommunikation bzw. der APPC-Datei für Nebeninformationen entnommen.

Diese Informationen werden auf verschiedenen Plattformen auf verschiedene Weisen festgelegt; weitere Informationen zum Einrichten der Kommunikation für Ihre Plattform finden Sie im Abschnitt [Verteilte Steuerung von Warteschlangen konfigurieren](#).

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

## **Maximale Anzahl Instanzen (MAXINST)**

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals oder AMQP-Kanals an, die gestartet werden können.

In den Unterabschnitten finden Sie Informationen darüber, wie das Attribut für die einzelnen Kanaltypen verwendet wird.

### **Zugehörige Informationen**

[Grenzwerte für Serververbindungskanäle](#)

[CHANNEL DEFINE CHANNEL](#)

## **Maximale Anzahl von Instanzen bei Verbindungen über Serververbindungskanäle**

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals an, die gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wird der Wert heruntersgesetzt, sodass er unter der Anzahl der momentan aktiven Instanzen des Serververbindungskanals liegt, hat dies keine Auswirkungen auf die aktiven Kanäle. Neue Instanzen können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

## **V 9.0.0 Maximale Anzahl von Instanzen bei Verbindungen über AMQP-Kanäle**

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines AMQP-Kanals an, die gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wenn ein Client versucht, eine Verbindung herzustellen, und die Anzahl der verbundenen Clients den Wert von MAXINST erreicht hat, schließt der Kanal die Verbindung mit einem Schließvorgangsframe. Der Schließvorgangsframe enthält die folgende Nachricht:

```
amqp:resource-limit-exceeded
```

Wenn sich ein Client mit einer bereits verbundenen ID verbindet (er also eine Clientübernahme vornimmt), ist die Übernahme erfolgreich, und zwar unabhängig davon, ob die Anzahl der verbundenen Clients den Wert von MAXINST bereits erreicht hat.

## **Maximale Anzahl Instanzen pro Client (MAXINSTC)**

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals an, die auf einem einzelnen Client gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wird der Wert heruntersgesetzt, sodass er unter der Anzahl der momentan auf den einzelnen Clients aktiven Instanzen des Serververbindungskanals liegt, hat dies keine Auswirkungen auf die aktiven Kanäle. Neue Instanzen auf diesen Clients können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

Dieses Attribut ist nur für Serververbindungskanäle gültig.

### **Zugehörige Informationen**

Grenzwerte für Serververbindungskanäle

CHANNEL DEFINE CHANNEL

## **Maximale Nachrichtenlänge (MAXMSGL)**

Dieses Attribut gibt die maximale Länge von Nachrichten an, die über den Kanal übertragen werden können.

**Multi** Auf IBM MQ for IBM i- Systemen, UNIXn und Windows-Systemen geben Sie einen Wert im Bereich zwischen null und der maximalen Nachrichtenlänge für den Warteschlangenmanager ein. Weitere Informationen finden Sie im Abschnitt zum MAXMSGL-Parameter des Befehls ALTER QMGR in [ALTER QMGR](#).

**z/OS** Geben Sie unter IBM MQ for z/OS einen Wert an, der größer-gleich null und kleiner-gleich 104 857 600 Byte ist (d. h. 100 MB).

Da es auf IBM MQ-Systemen unterschiedliche Implementierungsarten auf verschiedenen Plattformen gibt, ist in manchen Anwendungen die für die Nachrichtenverarbeitung verfügbare Länge möglicherweise

begrenzt. Dieser Wert muss einer Größe entsprechen, die Ihr System problemlos verarbeiten kann. Beim Kanalstart wird die niedrigere der zwei Zahlen für die beiden Kanäle übernommen.

Beim Hinzufügen der digitalen Signatur und des Schlüssels zur Nachricht erhöht Advanced Message Security die Länge der Nachricht.

#### **Anmerkung:**

1. Sie können als maximale Nachrichtengröße den Wert 0 festlegen, was so ausgelegt wird, dass die Größe auf den maximalen Wert des lokalen Warteschlangenmanagers festgelegt werden soll.

Dieses Attribut ist für alle Kanaltypen gültig.

### **Name des Nachrichtenkanalagenten (MCANAME)**

Dieses Attribut ist reserviert und muss bei Angabe auf Leerzeichen gesetzt werden.

Die maximale Länge beträgt 20 Zeichen.

### **Typ des Nachrichtenkanalagenten (MCATYPE)**

Dieses Attribut kann den Nachrichtenkanalagenten als *Prozess* oder *Thread* definieren.

Unter IBM MQ for z/OS wird er nur für Kanäle vom Typ Clusterempfänger unterstützt.

Vorteile der Ausführung als Prozess:

- Isolation der einzelnen Kanäle für mehr Integrität
- Für einzelne Kanäle spezifische Jobberechtigung
- Kontrolle über Job-Scheduling

Vorteile von Threads:

- Viel geringere Speicherbelegung
- Einfachere Konfiguration durch Eingabe über die Befehlszeile
- Schnellere Ausführung: es kostet weniger Zeit, einen Thread zu starten, als das Betriebssystem anzuweisen, einen Prozess zu starten

Für die Kanaltypen "Sender", "Server" und "Requester" gilt die Standardeinstellung `PROZESS`. Für die Kanaltypen "Clustersender" und "Clusterempfänger" gilt die Standardeinstellung `THREAD`. Diese Standardeinstellungen können bei der Installation geändert werden.

Bei Angabe von `PROZESS` in der Kanaldefinition wird ein `RUNMQCHL`-Prozess gestartet. Bei Angabe von `THREAD` wird der MCA auf einem Thread des `AMQRMPPA`-Prozesses bzw. auf einem Thread des `RUNMQCHI`-Prozesses ausgeführt, wenn `MQNOREMPOOL` angegeben ist. Auf der Maschine, die die eingehenden Zuordnungen empfängt, wird der MCA bei Verwendung von `RUNMQLSR` als Thread ausgeführt. Bei Eingabe von `inetd` wird er als Prozess ausgeführt.

Unter IBM MQ for z/OS wird dieses Attribut nur für Kanäle vom Typ Clusterempfänger unterstützt. Auf anderen Plattformen ist es für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clustersender
- Clusterempfänger

### **Benutzer-ID des Nachrichtenkanalagenten (MCAUSER)**

Dieses Attribut ist die Benutzer-ID (eine Zeichenfolge), die vom MCA zur Autorisierung für den Zugriff auf IBM MQ-Ressourcen verwendet wird.

**Anmerkung:** Alternativ dazu kann eine Benutzer-ID für einen Kanal, unter der dieser ausgeführt werden soll, über die Verwendung von Kanalauthentifizierungsdatensätzen bereitgestellt werden. Über Kanalauthentifizierungsdatensätze können verschiedene Verbindungen denselben Kanal mit unterschiedlichen Berechtigungsnachweisen verwenden. Wenn für einen Kanal sowohl MCAUSER gesetzt ist als auch Kanalauthentifizierungsdatensätze verwendet werden, haben die Kanalauthentifizierungsdatensätze Vorrang. Der Parameter MCAUSER in der Kanaldefinition wird nur verwendet, wenn der Kanalauthentifizierungsdatensatz USERSRC(CHANNEL) verwendet.

Diese Berechtigung beinhaltet (sofern die PUT-Berechtigung DEF ist) das Stellen der Nachricht in die Zielwarteschlange für Empfänger- oder Requesterkanäle.

In IBM MQ for Windows kann die Benutzer-ID domänenqualifiziert sein, indem sie im Format `user@domain` angegeben wird, wobei `domain` entweder die Windows-Systemdomäne des lokalen Systems oder eine vertrauenswürdige Domäne sein muss.

Wenn dieses Attribut leer ist, verwendet der MCA seine Standard-Benutzer-ID. Weitere Informationen finden Sie in [DEFINE CHANNEL](#).

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Serververbindung
- Clusterempfänger

### **Zugehörige Informationen**

[Kanalauthentifizierungssätze](#)

## **Name des Nachrichtensexits (MSGEXIT)**

Dieses Attribut gibt den Namen des Benutzerexitprogramms an, das vom Kanalnachrichtensexit ausgeführt werden soll.

Dieses Attribut kann aus einer Namensliste von Programmen bestehen, die nacheinander ausgeführt werden sollen. Lassen Sie das Attribut leer, wenn kein Kanalnachrichtensexit aktiv ist.

Das Format und die maximale Länge dieses Attributs sind plattformabhängig, wie bei [„Name des Empfangsexits \(RCVEXIT\)“](#) auf Seite 145.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## **Benutzerdaten des Nachrichtensexits (MSGDATA)**

Dieses Attribut gibt die Benutzerdaten an, die an die Kanalnachrichtensexits übergeben werden.

Sie können eine Sequenz von Nachrichtensexits ausführen. Die Beschränkungen für die Benutzerdatenlänge und ein Festlegungsbeispiel von MSGDATA für mehrere Exits entsprechen denen für RCVDATA. (siehe [„Benutzerdaten des Empfangsexits \(RCVDATA\)“](#) auf Seite 146).

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger

- Requester
- Clustersender
- Clusterempfänger

## **Name des Nachrichtenwiederholungsexits (MREXIT)**

Dieses Attribut gibt den Namen des Benutzerexitprogramms an, das vom Benutzerexit für Nachrichtenwiederholungen ausgeführt werden soll.

Lassen Sie dieses Attribut leer, wenn kein Exitprogramm für Nachrichtenwiederholung aktiv ist.

Das Format und die maximale Länge des Namens sind plattformabhängig, wie bei „Name des Empfangsexits (RCVEXIT)“ auf Seite 145. Es kann jedoch nur ein Exit für Nachrichtenwiederholungen angegeben werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## **Benutzerdaten des Kanalexits für Nachrichtenwiederholungen (MRDATA)**

Dieses Attribut gibt Daten an, die an den Kanalexit für Nachrichtenwiederholungen übergeben werden, wenn dieser aufgerufen wird.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## **Zähler für Nachrichtenwiederholungen (MRRTY)**

Dieses Attribut gibt die Anzahl der Versuche des Kanals an, die Nachricht erneut zu übermitteln.

Dieses Attribut steuert die Aktion des MCA nur, wenn kein Name für den Exit für Nachrichtenwiederholung angegeben ist. Wenn der Name des Exits nicht leer ist, wird der Wert von MRRTY an den Exit weitergeleitet, die Anzahl der Wiederholungsversuche (sofern überhaupt welche durchgeführt werden) wird dann aber durch den Exit, nicht durch dieses Attribut, bestimmt.

Der Wert muss im Bereich von 0 bis 999 999 999 liegen. Der Wert 0 bedeutet, dass keine weiteren Versuche ausgeführt werden. Der Standardwert ist 10.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## **Intervall für Nachrichtenwiederholungen (MRTMR)**

Dieses Attribut gibt die Mindestzeitspanne an, die vergehen muss, bevor der Kanal die MQPUT-Operation erneut versuchen kann.

Dieses Zeitintervall wird in Millisekunden angegeben.

Dieses Attribut steuert die Aktion des MCA nur, wenn kein Name für den Exit für Nachrichtenwiederholung angegeben ist. Wenn der Name des Exits nicht leer ist, wird der Wert von MRTMR zur Verwendung durch den Exit an den Exit übergeben, das Wiederholungsintervall wird jedoch durch den Exit gesteuert, nicht durch dieses Attribut.

Der Wert muss im Bereich von 0 bis 999 999 999 liegen. Der Wert 0 bedeutet, dass eine Wiederholung so schnell wie möglich ausgeführt wird (wenn der Wert für MRRTY größer als null ist). Der Standardwert ist 1000.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## Überwachung (MONCHL)

Dieses Attribut steuert die Erfassung von Online-Überwachungsdaten.

Mögliche Werte:

### QMGR

Die Erfassung der Daten aus der Online-Überwachung wird aus der Einstellung des Attributs MONCHL des Warteschlangenmanagerobjekts übernommen. Dies ist der Standardwert.

### OFF

Die Erfassung von Onlineüberwachungsdaten wird für diesen Kanal inaktiviert.

### LOW

Niedrige Datenerfassungsrate mit minimalen Auswirkungen auf die Systemleistung. Die abgebildeten Überwachungsergebnisse sind möglicherweise nicht aktuell.

### MITTEL

Mittlere Datenerfassungsrate mit begrenzten Auswirkungen auf die Leistung des Systems.

### HIGH

Hohe Datenerfassungsrate mit möglichen Auswirkungen auf die Leistung des Systems. Die abgebildeten Überwachungsergebnisse sind jedoch sehr aktuell.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Serververbindung
- Clustersender
- Clusterempfänger

Weitere Informationen zu Überwachungsdaten finden Sie im Abschnitt [Überwachungsdaten von Warteschlangen und Kanälen anzeigen](#).

## Kanalattribut NETPRTY

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Verwenden Sie das Attribut NETPRTY, um ein Netz als primäres Netz und ein weiteres Netz als Sicherungsnetz festzulegen. Bei einer Gruppe von Kanälen mit gleichem Rang wird durch Clustering der Pfad mit der höchsten Priorität ausgewählt, wenn mehrere Pfade zur Verfügung stehen.

Üblicherweise wird das Kanalattribut NETPRTY verwendet, um zwischen Netzen zu unterscheiden, die die gleichen Zielorte verbinden, aber unterschiedliche Aufwände verursachen oder unterschiedliche Geschwindigkeiten aufweisen.

**Anmerkung:** Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).

## Geschwindigkeit für nicht persistente Nachrichten (NPMSPEED)

Dieses Attribut gibt die Geschwindigkeit an, mit der nicht persistente Nachrichten gesendet werden sollen.

Mögliche Werte:

### **NORMAL**

Nicht persistente Nachrichten in einem Kanal werden innerhalb von Transaktionen übertragen.

### **FAST**

Nicht persistente Nachrichten in einem Kanal werden nicht innerhalb von Transaktionen übertragen.

Der Standardwert ist FAST. Der Vorteil besteht darin, dass nicht persistente Nachrichten sehr viel schneller zum Abruf verfügbar sind. Der Nachteil ist, dass Nachrichten bei einem Übertragungsfehler oder einem Stopp des Kanals während der Nachrichtenübertragung verloren gehen können, weil sie nicht Teil einer Transaktion sind. Siehe [Nachrichtensicherheit](#).

### **Hinweise:**

1. Wenn die aktiven Wiederherstellungsprotokolle für IBM MQ for z/OS häufiger als erwartet umschalten und archivieren, da die über einen Kanal gesendeten Nachrichten nicht persistent sind, kann die Einstellung NPMSPEED(FAST) sowohl auf der sendenden als auch auf der empfangenden Seite des Kanals die Aktualisierungen von SYSTEM.CHANNEL.SYNCQ minimieren.
2. Wenn es wegen der Aktualisierungen von SYSTEM.CHANNEL.SYNCQ zu einer hohen CPU-Auslastung kommt, kann das Festlegen von NPMSPEED(FAST) die CPU-Auslastung erheblich verringern.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Kennwort (PASSWORD)

Dieses Attribut gibt ein Kennwort an, das der MCA beim Herstellungsversuch einer sicheren LU 6.2-Sitzung zu einem fernen MCA verwenden kann.

Sie können ein Kennwort mit maximal 12 Zeichen angeben, von dem jedoch nur die ersten 10 Zeichen verwendet werden.

Es ist für die Kanaltypen Sender, Server, Requester und Clientverbindung gültig.

Unter IBM MQ for z/OS ist dieses Attribut nur für Clientverbindungskanäle gültig. Auf anderen Plattformen ist es für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender

## Portnummer (PORT)

Geben Sie die Portnummer an, die für die Verbindung des AMQP-Clients verwendet wird.

Der Standardport für AMQP 1.0-Verbindungen ist 5672. Falls Port 5672 bereits verwendet wird, können Sie einen anderen Port angeben.

## PUT-Berechtigung (PUTAUT)


Dieses Attribut gibt den Typ der vom MCA auszuführenden Sicherheitsverarbeitung an.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Serververbindung (nur z/OS)
- Clusterempfänger

Dieses Attribut wird verwendet, um den Typ der Sicherheitsverarbeitung festzulegen, den der MCA bei Ausführung verwenden soll:

- Ein MQPUT-Befehl an die Zielwarteschlange (für Nachrichtenkanäle) oder
- ein MQI-Aufruf (für MQI-Kanäle).

 Unter z/OS die markierten Benutzer-IDs, deren Anzahl von der Einstellung für das hlq.RES-LEVEL-Profil der MQADMIN-RACF-Klasse abhängt. Je nachdem, welche Zugangsstufe die Benutzer-ID des Kanalinitiators für das Profil hlq.RESLEVEL hat, sind null, ein oder zwei Benutzer-IDs markiert. Unter [RESLEVEL- und Kanalinitiator-Verbindungen](#) finden Sie Anweisungen zur Einsicht in die markierten Benutzer-IDs. Weitere Informationen zu markierten Benutzer-IDs finden Sie unter [Vom Kanalinitiator verwendete Benutzer-IDs](#).

Sie können eine der folgenden Optionen wählen:

### Prozesssicherheit, auch als Standardberechtigung (DEF) bezeichnet

Die Standard-Benutzer-ID wird verwendet.

Auf allen Plattformen außer z/OS wird zur Überprüfung der Öffnungsberechtigung für die Warteschlange die Benutzer-ID des Prozesses bzw. des Benutzers verwendet, der den MCA am empfangenden Ende des Nachrichtenkanals ausführt.

Unter z/OS wird je nach Anzahl der zu überprüfenden Benutzer-IDs entweder die aus dem Netz erhaltene oder die von [MCAUSER](#) abgeleitete Benutzer-ID verwendet.

Die Warteschlangen werden mit dieser Benutzer-ID und der Öffnungsoption MQOO\_SET\_ALL\_CONTEXT geöffnet.

### Kontextsicherheit (CTX)

Die Benutzer-ID aus den Kontextinformationen der Nachricht wird als alternative Benutzer-ID verwendet.

Die *UserIdentifier* im Nachrichtendeskriptor wird in das Feld *AlternateUserId* im Objektdeskriptor verschoben. Die Warteschlange wird mit den Öffnungsoptionen MQOO\_SET\_ALL\_CONTEXT und MQOO\_ALTERNATE\_USER\_AUTHORITY geöffnet.

Auf allen Plattformen außer z/OS wird zur Überprüfung der Öffnungsberechtigung für die Warteschlange für MQOO\_SET\_ALL\_CONTEXT und MQOO\_ALTERNATE\_USER\_AUTHORITY die Benutzer-ID des Prozesses bzw. des Benutzers verwendet, der den MCA am empfangenden Ende des Nachrichtenkanals ausführt. Die Benutzer-ID, die zum Überprüfen der Öffnungsberechtigung für die Warteschlange für MQOO\_OUTPUT verwendet wird, ist die *UserIdentifier* im Nachrichtendeskriptor.

Unter z/OS kann je nach Anzahl der zu überprüfenden Benutzer-IDs die aus dem Netz erhaltene oder die von [MCAUSER](#) abgeleitete Benutzer-ID verwendet werden, ebenso wie die Benutzer-ID aus den Kontextinformationen des Nachrichtendeskriptors.

Kontextsicherheit (CTX) wird für Serververbindungskanäle nicht unterstützt.

### Sicherheit ONLYMCA (Nur Nachrichtenkanalagent)

Die von [MCAUSER](#) abgeleitete Benutzer-ID wird verwendet.

Die Warteschlangen werden mit der Öffnungsoption MQOO\_SET\_ALL\_CONTEXT geöffnet.

Dieser Wert wird nur unter z/OS unterstützt.






## Sicherheit ALTMCA (Alternativer Nachrichtenkanalagent)

Je nach Anzahl der zu überprüfenden Benutzer-IDs wird die Benutzer-ID aus den Kontextinformationen des Nachrichtendeskriptors (Feld *UserIdentifier*) oder die von MCAUSER abgeleitete Benutzer-ID verwendet.

Dieser Wert wird nur unter z/OS unterstützt.

Weitere Details zu Kontextfeldern und Öffnungsoptionen finden Sie im Abschnitt [Kontextinformationen steuern](#).

Weitere Informationen zu Sicherheit finden Sie hier:

- [Sicherheit](#)
-  [Sicherheit unter UNIX, Linux, and Windows einrichten](#)
-  [Sicherheit unter IBM i einrichten](#)
-  [Sicherheit unter z/OS einrichten](#)

## Name des Warteschlangenmanagers (QMNAME)

Dieses Attribut gibt den Namen des Warteschlangenmanagers oder der Warteschlangenmanagergruppe an, zu dem bzw. der von einer IBM MQ MQI client-Anwendung eine Verbindung angefordert werden kann.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clientverbindung

## Name des Empfangsexits (RCVEXIT)

Dieses Attribut gibt den Namen des Benutzerexitprogramms an, das vom Empfangsbenutzerexit des Kanals ausgeführt werden soll.

Dieses Attribut kann aus einer Namensliste von Programmen bestehen, die nacheinander ausgeführt werden sollen. Ist kein Kanalempfangsbenutzerexit aktiv, wird dieses Attribut leer gelassen.

Das Format und die maximale Länge dieses Attributs hängen von der Plattform ab:

- Unter z/OS handelt es sich um einen Lademodulnamen mit einer maximalen Länge von 8 Zeichen bzw. bei Clientverbindungskanälen mit maximal 128 Zeichen.
- Auf IBM i wird folgendes Format verwendet:

```
libname/progname
```

wenn es in CL-Befehlen verwendet wird.

Bei Angabe in IBM MQ-Befehlen (MQSC) hat es das Format:

```
progname libname
```

, wobei *progname* die ersten zehn Zeichen belegt und *libname* die zweite Gruppe von 10 Zeichen (beide sofern erforderlich nach rechts mit Leerzeichen gefüllt). Die maximale Länge der Zeichenfolge beträgt 20 Zeichen.

- Auf Windowsn wird folgendes Format verwendet:

```
dllname(functionname)
```

Dabei wird *dllname* ohne das Suffix `.DLL` angegeben. Die maximale Länge der Zeichenfolge beträgt 40 Zeichen.

- Auf UNIXn wird folgendes Format verwendet:

```
libraryname(functionname)
```

Die maximale Länge der Zeichenfolge beträgt 40 Zeichen.

Bei der automatischen Definition des Clustersenderkanals unter z/OS werden Kanalexitnamen in das z/OS-Format konvertiert. Wenn Sie die Konvertierung von Exitnamen steuern möchten, können Sie einen Exit für die automatische Kanaldefinition eingeben. Weitere Informationen finden Sie unter [Exitprogramm für die automatische Kanaldefinition](#).

Sie können eine Namensliste von Empfangs-, Sende- oder Nachrichtenexitprogrammen angeben. Die Namen müssen durch ein Komma, ein Leerzeichen oder beides getrennt sein. Beispiel:

```
RCVEXIT(exit1 exit2)  
MSGEXIT(exit1,exit2)  
SENDEXIT(exit1, exit2)
```

Die Gesamtlänge der Zeichenfolge aus Exitnamen und Zeichenfolgen aus Benutzerdaten für einen bestimmten Exittyp ist auf 500 Zeichen begrenzt. In IBM MQ for IBM i können bis zu 10 Exitnamen aufgelistet werden. In IBM MQ for z/OS können bis zu acht Exitnamen aufgelistet werden.

Dieses Attribut ist für alle Kanaltypen gültig.

## Benutzerdaten des Empfangsexits (RCVDATA)

Dieses Attribut gibt die Benutzerdaten an, die an das Empfangsexit übermittelt werden.

Sie können eine Reihe von Empfangsexits ausführen. Die Zeichenfolge aus Benutzerdaten für eine Reihe von Exits muss jeweils durch Komma, Leerzeichen oder beides getrennt werden. Beispiel:

```
RCVDATA(exit1_data exit2_data)  
MSGDATA(exit1_data,exit2_data)  
SENDDATA(exit1_data, exit2_data)
```

Auf IBM MQ for UNIX- und Windows-Systemen ist die Länge der Zeichenfolge aus Exitnamen und Zeichenfolgen aus Benutzerdaten auf 500 Zeichen begrenzt. In IBM MQ for IBM i können Sie bis zu 10 Exitnamen angeben und die Länge der Benutzerdaten ist jeweils auf 32 Zeichen beschränkt. Unter IBM MQ for z/OS können Sie bis zu acht Zeichenfolgen aus Benutzerdaten angeben, von denen jede 32 Zeichen lang sein kann.

Dieses Attribut ist für alle Kanaltypen gültig.

## Name des Sicherheitsexits (SCYEXIT)

Dieses Attribut gibt den Namen des Exitprogramms an, das vom Kanalsicherheitsexit ausgeführt werden muss.

Ist kein Kanalsicherheitsexit aktiv, wird das Attribut leer gelassen.

Das Format und die maximale Länge des Namens sind plattformabhängig, wie bei „[Name des Empfangsexits \(RCVEXIT\)](#)“ auf Seite 145. Sie können jedoch nur ein Sicherheitsexit angeben.

Dieses Attribut ist für alle Kanaltypen gültig.

## Benutzerdaten des Sicherheitsexits (SCYDATA)

Dieses Attribut gibt die Benutzerdaten an, die an das Sicherheitsexit übergeben werden.

Die maximal zulässige Länge beträgt 32 Zeichen.

Dieses Attribut ist für alle Kanaltypen gültig.

## Name des Sendeexits (SENDEXIT)

Dieses Attribut gibt den Namen des Exitprogramms an, das vom Kanalsendeexit ausgeführt wird.

Dieses Attribut kann eine Liste von Programmnamen sein, die nacheinander ausgeführt werden müssen. Lassen Sie das Attribut leer, wenn kein Sendeexit aktiv ist.

Das Format und die maximale Länge dieses Attributs sind plattformabhängig, wie bei „Name des Empfangsexits (RCVEXIT)“ auf Seite 145.

Dieses Attribut ist für alle Kanaltypen gültig.

## Benutzerdaten des Sendeexits (SENDDATA)

Dieses Attribut gibt die Benutzerdaten an, die an das Sendeexit übergeben werden.

Sie können eine Sequenz von Sendeexits ausführen. Die Beschränkungen für die Benutzerdatenlänge und die Angabe von SENDDATA für mehrere Exits entsprechen denen für RCVDATA. (siehe „Benutzerdaten des Empfangsexits (RCVDATA)“ auf Seite 146).

Dieses Attribut ist für alle Kanaltypen gültig.

## Wert für Folgenummernserie (SEQWRAP)

Dieses Attribut gibt die höchste Nummer an, die die Nachrichtenfolgennummer erreicht, bevor sie wieder bei 1 beginnt.

Der Wert der Nummer muss hoch genug sein, um zu vermeiden, dass eine Nummer erneut verwendet wird, während diese noch durch eine frühere Nachricht in Verwendung ist. Den beiden Kanaltypen muss derselbe Wert für die Folgenummernserie zugeordnet sein, wenn der Kanal gestartet wird; andernfalls tritt ein Fehler auf.

Hier kann ein Wert zwischen 100 und 999 999 999 angegeben werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Kurzer Wiederholungszähler (SHORTRTY)

Dieses Attribut gibt die maximale Anzahl an Wiederholungsversuchen des Kanals an, seinem Partner eine Sitzung zuzuordnen.

Das Attribut *Zähler für kurze Wiederholungsversuche* kann auf einen Wert zwischen null und 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Wenn der erste Zuordnungsversuch fehlschlägt, wird der Wert für *short retry count* verringert und der Kanal wiederholt die verbleibenden Wiederholungen mit einem Intervall, das im Attribut **short retry interval** definiert ist, zwischen den einzelnen Versuchen. Wenn er immer noch fehlschlägt, wiederholt er *long retry count* die Anzahl der Wiederholungen mit einem Intervall von *long retry*

*interval* zwischen den einzelnen Versuchen. Kann die Operation weiterhin nicht ausgeführt werden, wird der Kanal geschlossen.

Unter z/OS kann ein Kanal keine Wiederholungen durchführen, wenn die maximale Anzahl an Kanälen (**MAXCHL**) überschritten wurde.

Auf IBM i-, UNIX- und Windows-Systemen wird nur eine Wiederholung versucht, wenn ein Kanalinitiator ausgeführt wird. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

Wenn der Kanalinitiator (unter z/OS) bzw. der Kanal (auf Multiplatforms) während eines Wiederholungsversuchs gestoppt wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* zurückgesetzt, wenn der Kanalinitiator bzw. der Kanal neu gestartet wird oder wenn eine Nachricht erfolgreich über den Senderkanal versendet wird. Wenn jedoch der Kanalinitiator (unter z/OS) bzw. der Warteschlangenmanager (auf Multiplatforms) beendet und erneut gestartet wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht zurückgesetzt. Der Kanal behält die Werte des Zählers für Wiederholungsversuche bei, die vor dem Neustart des Warteschlangenmanagers bzw. vor dem Versenden der Nachricht galten.

Auf IBM i-Systemen, UNIXn und Windows-Systemen:

1. Wenn der Status des Kanals von RE TRYING zu RUNNING wechselt, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht unverzüglich zurückgesetzt. Die Zähler werden erst zurückgesetzt, wenn die erste Nachricht erfolgreich über den Kanal versendet wird, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., wenn der lokale Kanal die Anzahl der versendeten Nachrichten bestätigt hat.
2. Der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* werden beim Neustart des Kanals zurückgesetzt.

## **Intervall für kurze Wiederholungsversuche (SHORTTMR)**

Dieses Attribut gibt die ungefähre Länge des Zeitintervalls in Sekunden an, das der Kanal bis zum erneuten Versuch eines Verbindungsaufbaus abwartet, während der Modus für kurze Wiederholungsversuche aktiviert ist.

Das Intervall zwischen zwei Verbindungsversuchen kann größer sein, wenn ein Kanal abwarten muss, bis er aktiv ist.

Dieses Attribut kann auf einen Wert von null bis 999 999 festgelegt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## **SSL-Verschlüsselungsspezifikation (SSLCIPH)**

Dieses Attribut gibt eine einzelne CipherSpec für eine TLS-Verbindung an.

Jede IBM MQ-Kanaldefinition enthält das Attribut SSLCIPH. Der Wert besteht aus einer Zeichenfolge mit einer maximalen Länge von 32 Zeichen.

Dabei ist Folgendes zu beachten:

- Das Attribut SSLCIPH kann einen Leerwert enthalten; dies bedeutet, dass Sie TLS nicht verwenden. Wenn ein Ende des Kanals ein Attribut SSLCIPH mit einem Leerwert aufweist, muss auch das andere Ende des Kanals ein leeres Attribut des Typs SSLCIPH aufweisen.
- Wenn SSLCIPH einen nicht leeren Wert aufweist, versucht der Kanal alternativ, die angegebene Verschlüsselung für die Nutzung von TLS zu verwenden. Auch in diesem Fall müssen beide Enden des Kanals denselben SSLCIPH-Wert aufweisen.

- Für die Regel, dass der Wert des Attributs SSLCIPH an beiden Enden eines Kanals identisch sein muss, gibt es nur eine Ausnahme: Auf einem vollständig verwalteten .NET-Client kann der Sonderwert **\*NEGOTIATE** angegeben werden. Diese Option ermöglicht dem Kanal die Auswahl der neuesten Protokollversion, die vom .NET-Framework unterstützt wird sowie das Aushandeln eines CipherSpec-Werts, der vom Server unterstützt wird.

Dieses Attribut gilt nur für Kanäle mit dem Transporttyp (TRPTYPE) TCP. Wenn TRPTYPE nicht TCP ist, werden die Daten ignoriert, ohne dass eine Fehlernachricht ausgegeben wird.

Weitere Informationen zu SSLCIPH finden Sie in [DEFINE CHANNEL](#) und [CipherSpecs](#) angeben.

## SSL-Clientauthentifizierung (SSLCAUTH)

Dieses Attribut gibt an, ob der Kanal ein TLS-Zertifikat von einem TLS-Client empfangen und authentifizieren muss.

Mögliche Werte:

### OPTIONAL

Wenn der TLS-Peer-Client ein Zertifikat sendet, wird das Zertifikat normal verarbeitet; die Authentifizierung schlägt aber nicht fehl, wenn kein Zertifikat gesendet wird.

### ERFORDERLICH

Wenn der TLS-Client kein Zertifikat sendet, schlägt die Authentifizierung fehl.

Der Standardwert ist REQUIRED.

Sie können in einer Nicht-TLS-Kanaldefinition einen Wert für SSLCAUTH angeben. Dabei handelt es sich um eine Kanaldefinition, in der das Attribut SSLCIPH fehlt oder leer ist.

SSLCAUTH ist ein optionales Attribut.

Dieses Attribut ist mit Ausnahme von Senderkanälen für alle Kanaltypen gültig, die einen Kanalinitialisierungsfluss empfangen.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Server
- Empfänger
- Requester
- Serververbindung
- Clusterempfänger

Weitere Informationen zu SSLCAUTH finden Sie in den Abschnitten [DEFINE CHANNEL \(MQTT\)](#) und [Sicherheit](#).

## SSL-Peer (SSLPEER)

Dieses Attribut wird zum Überprüfen des definierten Namens (Distinguished Name, DN) des Zertifikats vom Peerwarteschlangenmanager oder Client am anderen Ende eines IBM MQ-Kanals aus verwendet.

**Anmerkung:** Alternativ können zur Beschränkung von Verbindungen auf bestimmte Kanäle durch Überprüfung des definierten TLS-Namens auch Kanalauthentifizierungsdatensätze verwendet werden. Über die Authentifizierungsdatensätze für Kanäle können verschiedene Muster für definierte Namen des Zertifikatinhabers in TLS auf denselben Kanal angewendet werden. Wenn SSLPEER auf dem Kanal und ein Kanalauthentifizierungsdatensatz für denselben Kanal verwendet werden, muss das eingehende Zertifikat mit beiden Mustern übereinstimmen, damit eine Verbindung hergestellt werden kann.

Wenn der vom Peer empfangene definierte Name nicht mit dem SSLPEER-Wert übereinstimmt, wird der Kanal nicht gestartet.

SSLPEER ist ein optionales Attribut. Ist hier kein Wert angegeben, wird der definierte Name des Peers beim Starten des Kanals nicht überprüft.

Unter z/OS beträgt die maximale Länge des Attributs 256 Byte. Auf allen anderen Plattformen beträgt sie 1024 Byte. Kanalauthentifizierungsdatensätze ermöglichen mehr Flexibilität bei der Verwendung von SSLPEER und Unterstützung von 1024 Bytes auf allen Plattformen.

Unter z/OS werden die verwendeten Attributwerte nicht überprüft. Wenn Sie falsche Werte eingeben, schlägt der Kanal beim Starten fehl und an beiden Enden des Kanals werden in das Fehlerprotokoll Fehlermeldungen ausgegeben. Außerdem wird an beiden Enden des Kanals ein SSL-Fehler-Ereignis für den Kanal erstellt. Auf Plattformen, die SSLPEER unterstützen (keine z/OS-Plattformen), wird die Gültigkeit der Zeichenfolge bei der ersten Eingabe überprüft.

Sie können in einer Nicht-TLS-Kanaldefinition, in der SSLCIPH fehlt oder leer ist, für SSLPEER einen Wert eingeben. In diesem Fall können Sie TLS temporär für die Fehlerbehebung inaktivieren, ohne die TLS-Parameter entfernen und später erneut eingeben zu müssen.

Weitere Informationen zur Verwendung von SSLPEER finden Sie in den Abschnitten [SET CHLAUTH](#) und [Sicherheit](#).

Dieses Attribut ist für alle Kanaltypen gültig.

### Zugehörige Informationen

[Kanalauthentifizierungsdatensätze](#)

## V 9.0.0 Themen-Root (TPROOT)

Dieses Attribut gibt den Themen-Root für einen AMQP-Kanal an.

Mit dem Attribut TPROOT können Sie einen Themen-Root für einen AMQP-Kanal angeben. Durch die Verwendung dieses Attributs ist sichergestellt, dass eine MQ Light-Anwendung bei ihrer Implementierung auf einem Warteschlangenmanager Nachrichten nicht in den Bereichen der Themenstruktur veröffentlicht, die von anderen Anwendungen verwendet werden, oder Nachrichten aus derartigen Bereichen subskribiert.

Der Standardwert für TPROOT ist SYSTEM.BASE.TOPIC. Bei diesem Wert weist die Themenzeichenfolge, die von einem AMQP-Client für die Veröffentlichung oder Subskription verwendet wird, kein Präfix auf und der Client kann Nachrichten mit anderen MQ-Publish/Subscribe-Anwendungen austauschen. Damit AMQP-Clients unter einem Themenpräfix Veröffentlichungen und Subskriptionen vornehmen können, müssen Sie zunächst ein MQ-Themenobjekt mit einer Themenzeichenfolge erstellen, die mit dem gewünschten Präfix festgelegt wird. Anschließend müssen Sie den Wert des Attributs TPROOT für den AMQP-Kanal in den Namen des von Ihnen erstellten MQ-Themenobjekts ändern. Im folgenden Beispiel ist ein Themen-Root dargestellt, der für den AMQP-Kanal MYAMQP auf den Wert APPGROUP1.BASE.TOPIC gesetzt wurde:

```
DEFINE CHANNEL(MYAMQP) CHLTYPE(AMQP) TPROOT(APPGROUP1.BASE.TOPIC) PORT(5673)
```

**Anmerkung:** Wenn der TPROOT-Attributwert oder die von ihm unterstützte Themenzeichenfolge geändert wird, kann dies dazu führen, dass bereits vorhandene AMQP-Themen und ihre Nachrichten verwaist sind.

## Name der Übertragungswarteschlange (XMITQ)

Dieses Attribut gibt den Namen der Übertragungswarteschlange an, aus der Nachrichten abgerufen werden.

Dieses Attribut ist für Kanäle vom Typ Sender und Server erforderlich und für andere Kanaltypen nicht gültig.

Geben Sie den Namen der Übertragungswarteschlange an, die diesem Sender- bzw. Serverkanal, der dem Warteschlangenmanager am fernen Ende des Kanals entspricht, zugeordnet werden muss. Sie können der Übertragungswarteschlange den gleichen Namen wie dem Warteschlangenmanager am fernen Ende geben.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server

## Transporttyp (TRPTYPE)

Dieses Attribut gibt den zu verwendenden Transporttyp an.

Folgende Werte sind möglich:

LU62	LU 6.2
TCP	TCP/IP
NETBIOS	NetBIOS („1“ auf Seite 151)
SPX	SPX („1“ auf Seite 151)
<p><b>Hinweise:</b></p> <p>1. Zur Verwendung unter Windows. Kann auch unter z/OS zur Definition von Clientverbindungskanälen zur Verwendung unter Windows eingesetzt werden.</p>	

Dieses Attribut gilt für alle Kanaltypen, wird jedoch von antwortenden Nachrichtenkanalagenten ignoriert.

## Client-ID verwenden (USECLTID)

Verwenden Sie die Client-ID für die Verbindung mit dem AMQP-Kanal.

Geben Sie an, ob die Client-ID für die Verbindung bei einem AMQP-Kanal verwendet werden soll. Setzen Sie dieses Attribut auf Yes (Ja) oder No (Nein).

## Warteschlange für nicht zustellbare Mail verwenden (USEDLQ)

Dieses Attribut bestimmt, ob die Warteschlange für nicht zustellbare Nachrichten (oder die Warteschlange für nicht zugestellte Nachrichten) verwendet wird, wenn Kanäle Nachrichten nicht übermitteln können.

Mögliche Werte:

### NO

Nachrichten, die von einem Kanal nicht zugestellt werden konnten, werden als Fehler behandelt. Entweder verwirft der Kanal die Nachricht oder der Kanal wird abgebrochen, entsprechend der NPMSPEED-Einstellung.

### YES (Standardwert)

Wenn das Attribut DEADQ des Warteschlangenmanagers den Namen einer Warteschlange für nicht zustellbare Nachrichten angibt, wird diese Warteschlange verwendet. Andernfalls ist das Verhalten wie bei NO.

## Benutzer-ID (USERID)

Dieses Attribut gibt die Benutzer-ID an, die der MCA beim Aufbauversuch einer sicheren SNA-Sitzung zu einem fernen MCA verwenden soll.

Sie können eine Task-Benutzer-ID mit 20 Zeichen angeben.

Es ist für die Kanaltypen Sender, Server, Requester und Clientverbindung gültig.

Dieses Attribut gilt mit Ausnahme von Clientverbindungskanälen nicht für IBM MQ for z/OS.

Wenn auf der Empfangsseite verschlüsselte Kennwörter verwendet werden und die LU 6.2-Software ein anderes Verschlüsselungsverfahren verwendet, schlägt der Kanalstart mit Einzelangaben zu ungültigen Sicherheitsbedingungen fehl. Dies kann vermieden werden, indem die empfangende SNA-Konfiguration dahingehend geändert wird, dass entweder

- die Kennwortersetzung inaktiviert wird oder

- eine Sicherheitsbenutzer-ID und das entsprechende Kennwort definiert werden.


Unter IBM MQ for z/OS ist dieses Attribut nur für Clientverbindungskanäle gültig. Auf anderen Plattformen ist es für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender

## IBM MQ-Clusterbefehle

Die zu den IBM MQ-Scriptbefehlen zählenden **runmqsc**-Befehle weisen spezielle Attribute und Parameter für Cluster auf. Es gibt aber auch andere Verwaltungsschnittstellen, die Sie zur Verwaltung von Clustern verwenden können.

Die MQSC-Befehle werden angezeigt, als ob Sie vom Systemadministrator in der Befehlskonsole eingegeben worden wären. Sie müssen diese Befehle aber nicht auf diese Weise ausgeben. Je nach Plattform gibt es verschiedene andere Methoden zur Ausgabe dieser Befehle, zum Beispiel:

- In IBM MQ for IBM i führen Sie MQSC-Befehle interaktiv über Option 26 von **WRKMQM** aus. Ebenso können Sie CL-Befehle verwenden oder MQSC-Befehle in einer Datei speichern und den CL-Befehl **STRMQMMQSC** verwenden.
-  Unter z/OS können Sie die Funktion COMMAND des Dienstprogramms **CSQUTIL**, die Bedienungs- und Steuerkonsolen oder die z/OS-Konsole verwenden.
- Auf allen anderen Plattformen können Sie die Befehle in einer Datei speichern und den Befehl **runmqsc** verwenden.

In einem MQSC-Befehl kann ein mit dem Attribut CLUSTER angegebener Clustername bis zu 48 Zeichen lang sein.

Eine mit dem Attribut CLUSNL angegebene Liste mit Clusternamen kann bis zu 256 Namen enthalten. Zur Erstellung einer Cluster-Namensliste verwenden Sie den Befehl **DEFINE NAMELIST**.

### IBM MQ Explorer

Die grafische Benutzerschnittstelle von IBM MQ Explorer kann einen Cluster mit Repository-Warteschlangenmanagern unter IBM WebSphere MQ for z/OS 6 oder höher verwalten. Die Benennung eines weiteren Repositorys auf einem separaten System ist nicht erforderlich. Bei früheren Versionen von IBM MQ for z/OS kann IBM MQ Explorer keinen Cluster mit Repository-Warteschlangenmanagern verwalten. Daher müssen Sie ein zusätzliches Repository auf einem System benennen, das von IBM MQ Explorer verwaltet werden kann.

Unter IBM MQ for Windows und IBM MQ for Linux können Sie auch IBM MQ Explorer für die Arbeit mit Clustern verwenden. Ebenso können Sie den eigenständigen IBM MQ Explorer-Client verwenden.

Bei Verwendung von IBM MQ Explorer können Sie Clusterwarteschlangen anzeigen und den Status von Clustersender- und Clusterempfängerkanälen abfragen. In IBM MQ Explorer gibt es zwei Assistenten, die Sie durch folgende Tasks führen:

- Cluster erstellen
- Unabhängigen Warteschlangenmanager mit einem Cluster verbinden



## Programmable Command Formats (PCF)

Tabelle 31. PCF-Entsprechungen zu MQSC-Befehlen für die Verwaltung von Clustern	
runmqsc-Befehl	PCF-Entsprechung
DISPLAY CLUSQMGR	MQCMD_INQUIRE_CLUSTER_Q_MGR
SUSPEND QMGR	MQCMD_SUSPEND_Q_MGR_CLUSTER
RESUME QMGR	MQCMD_RESUME_Q_MGR_CLUSTER
REFRESH CLUSTER	MQCMD_REFRESH_CLUSTER
RESET CLUSTER	MQCMD_RESET_CLUSTER

### Zugehörige Informationen

[Clustering: Best Practices für REFRESH CLUSTER verwenden](#)

## Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Um anzugeben, dass ein Warteschlangenmanager über ein vollständiges Repository für einen Cluster verfügt, verwenden Sie den Befehl ALTER QMGR unter Angabe des Attributs REPOS (*Clustername*). Um eine Liste mehrerer Clusternamen anzugeben, definieren Sie eine Clusternamensliste und verwenden Sie dann das Attribut REPOSNL (*Namensliste*) im Befehl ALTER QMGR:

```
DEFINE NAMLIST(CLUSTERLIST)
  DESCR('List of clusters whose repositories I host')
  NAMES(CLUS1, CLUS2, CLUS3)
ALTER QMGR REPOSNL(CLUSTERLIST)
```

Sie können im Befehl ALTER QMGR auch zusätzliche Clusterattribute angeben.

### CLWLEXIT (*Name*)

Gibt den Namen eines Benutzerexits an, der aufgerufen werden soll, wenn eine Nachricht in eine Clusterwarteschlange eingereicht wird.

### CLWLDATA (*Daten*)

Gibt die Daten an, die an den Benutzerexit für Clusterauslastung übergeben werden sollen.

### CLWLLEN (*Länge*)

Gibt die Höchstmenge an Nachrichtendaten an, die an den Benutzerexit für Clusterauslastung übergeben werden sollen.

### CLWLMRUC (*Kanäle*)

Gibt die maximale Anzahl an abgehenden Clusterkanälen an.

CLWLMRUC ist ein Attribut für einen lokalen Warteschlangenmanager, das nicht innerhalb des Clusters weitergegeben wird. Es wird den Exits für Clusterauslastung und dem Algorithmus für Clusterauslastung zur Verfügung gestellt, der die Zieladresse für Nachrichten auswählt.

### CLWLUSEQ( LOCAL|ANY )

Gibt das Verhalten der MQPUT-Operation an, wenn die Zielwarteschlange eine lokale Instanz und mindestens eine ferne Clusterinstanz besitzt. Geht der PUT-Vorgang von einem Clusterkanal aus, wird dieses Attribut nicht verwendet. Es ist möglich, CLWLUSEQ als Warteschlangenattribut und als Warteschlangenmanagerattribut anzugeben.

Wenn Sie die Option ANY angeben, sind sowohl die lokale Warteschlange als auch die fernen Warteschlangen mögliche Ziele der MQPUT-Operation.

Wenn Sie die Option LOCAL angeben, ist die lokale Warteschlange das einzige Ziel der MQPUT-Operation.

Die entsprechenden PCFs lauten MQCMD\_CHANGE\_Q\_MGR und MQCMD\_INQUIRE\_Q\_MGR.

## Zugehörige Konzepte

### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

## Zugehörige Verweise

### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Mit den Befehlen `SUSPEND QMGR` und `RESUME QMGR` können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Die Befehle `DEFINE CHANNEL`, `ALTER CHANNEL` und `DISPLAY CHANNEL` beinhalten zwei bestimmte `CHLTYPE`-Parameter für Cluster: `CLUSRCVR` und `CLUSSDR`. Um einen Clusterempfängerkanal zu definieren, verwenden Sie den Befehl `DEFINE CHANNEL`, wobei Sie `CHLTYPE (CLUSRCVR)` angeben. Viele der Attribute in einer Definition für einen Clusterempfängerkanal sind die gleichen Attribute wie die in einer Definition für einen Empfänger- oder Senderkanal. Um einen Clustersenderkanal zu definieren, verwenden Sie den Befehl `DEFINE CHANNEL`, wobei Sie `CHLTYPE (CLUSSDR)` angeben, sowie viele der Attribute, die Sie auch bei der Definition eines Senderkanals verwenden.

Für die Definition eines Clustersenderkanals ist es nicht mehr erforderlich, den Namen des vollständigen Warteschlangenmanager-Repositorys anzugeben. Wenn Ihnen die Namenskonvention bekannt ist, die für Kanäle in Ihrem Cluster verwendet wird, können Sie eine `CLUSSDR`-Definition mithilfe der Konstruktion `+QMNAME+` vornehmen. Die `+QMNAME+`-Konstruktion wird unter z/OS nicht unterstützt. Nachdem die Verbindung hergestellt wurde, ändert IBM MQ den Namen des Kanals und ersetzt `+QMNAME+` durch den korrekten Namen des vollständigen Warteschlangenmanager-Repositorys. Der so entstandene Kanalname wird auf 20 Zeichen abgeschnitten.

Weitere Informationen zu Namenskonventionen finden Sie im Abschnitt Namenskonventionen clustern.

Das Verfahren funktioniert nur dann, wenn Ihre Konvention für die Benennung von Kanälen den Namen des Warteschlangenmanagers miteinbezieht. Ein Beispiel: Sie definieren einen Warteschlangenmanager für ein vollständiges Repository mit der Bezeichnung `QM1` in einem Cluster mit der Bezeichnung `CLUSTER1` mit einem Clusterempfängerkanal `CLUSTER1.QM1.ALPHA`. Jeder andere Warteschlangenma-

nager kann einen Clustersenderkanal für diesen Warteschlangenmanager mit dem Kanalnamen CLUSTER1. +QMNAME+ . ALPHA definieren.

Wenn Sie für alle Ihre Kanäle die gleiche Namenskonvention verwenden, bedenken Sie, dass es immer nur eine +QMNAME+-Definition geben kann.

Die folgenden Attribute in den Befehlen DEFINE CHANNEL und ALTER CHANNEL gelten speziell für Clusterkanäle:

#### **CLUSTER**

Das Attribut CLUSTER gibt den Namen des Clusters an, dem dieser Kanal zugeordnet ist. Alternativ dazu können Sie das Attribut CLUSNL verwenden.

#### **CLUSNL**

Das Attribut CLUSNL gibt eine Namensliste mit Clusternamen an.

#### **NETPRTY**

Nur Clusterempfänger.

Das Attribut NETPRTY gibt eine Netzpriorität für den Kanal an. NETPRTY unterstützt die Auslastungsmanagementroutinen. Wenn es mehrere mögliche Routen zu einer Zieladresse gibt, wählt die Auslastungsmanagementroutine diejenige mit der höchsten Priorität aus.

#### **CLWLPRTY**

Der Parameter CLWLPRTY weist zu Auslastungsmanagementzwecken Kanälen mit der gleichen Zieladresse einen Prioritätsfaktor zu. Dieser Parameter gibt in Zusammenhang mit einer gleichmäßigen Clusterauslastung die jeweilige Priorität eines Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste Priorität und 9 die höchste Priorität bezeichnet.

#### **CLWLRANK**

Der Parameter CLWLRANK weist zu Auslastungsmanagementzwecken den Kanälen einen Rangfolgefaktor zu. Dieser Parameter gibt in Zusammenhang mit einer gleichmäßigen Clusterauslastung den jeweiligen Rang eines Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 den niedrigsten Rang und 9 den höchsten Rang bezeichnet.

#### **CLWLWGHT**

Der Parameter CLWLWGHT weist zum Zweck einer gleichmäßigen Clusterauslastung den Kanälen einen Gewichtungsfaktor zu. CLWLWGHT gewichtet den jeweiligen Kanal, sodass der Anteil der Nachrichten, die über diesen Kanal versendet werden, gesteuert werden kann. Der Algorithmus für Clusterauslastung verwendet den Parameter CLWLWGHT, um die Auswahl der Zieladresse zu beeinflussen, sodass mehr Nachrichten über einen bestimmten Kanal versendet werden können. Alle Kanalgewichtungsattribute haben standardmäßig den gleichen Standardwert. Das Gewichtungsattribut ermöglicht es Ihnen, einem Kanal auf einer leistungsfähigen UNIX-Maschine eine stärkere Gewichtung zu geben als einem Kanal auf einem kleinen Desktop-PC. Eine stärkere Gewichtung bedeutet, dass der Algorithmus für Clusterauslastung die UNIX-Maschine häufiger als Zieladresse für Nachrichten auswählt als den PC.

#### **CONNAME**

Der Parameter CONNAME, der in der Definition für einen Clusterempfängerkanal angegeben ist, wird im gesamten Cluster verwendet, um die Netzadresse des Warteschlangenmanagers zu bestimmen. Achten Sie darauf, für den Parameter CONNAME einen Wert auszuwählen, der in Ihrem gesamten IBM MQ-Cluster aufgelöst wird. Verwenden Sie keinen generischen Namen. Beachten Sie, dass der Wert, der im Clusterempfängerkanal angegeben wird, Vorrang vor jedem Wert hat, der in einem entsprechenden Clustersenderkanal angegeben wird.

Diese Attribute in den Befehlen DEFINE CHANNEL und ALTER CHANNEL gelten ebenso für den Befehl DISPLAY CHANNEL.

**Anmerkung:** Automatisch definierte Clustersenderkanäle erhalten ihre Attribute von der jeweiligen Definition für den entsprechenden Clusterempfängerkanal im Empfangswarteschlangenmanager. Auch wenn es sich um einen manuell definierten Clustersenderkanal handelt, werden seine Attribute automatisch geändert, um sicherzustellen, dass sie mit den Attributen der entsprechenden Clusterempfängerdefinition übereinstimmen. Sie können beispielsweise einen CLUSRCVR-Kanal definieren, ohne eine Portnummer im Parameter CONNAME anzugeben, und dabei manuell einen CLUSSDR-Kanal definieren, bei dem eine Portnummer angegeben wird. Wenn der automatisch definierte CLUSSDR den manuell definierten Clus-

tersenderkanal ersetzt, wird die Portnummer (vom CLUSRCVR übernommen) leer. In diesem Fall würde die Standardportnummer verwendet werden und der Kanal würde fehlschlagen.

**Anmerkung:** Der Befehl `DISPLAY CHANNEL` zeigt keine automatisch definierten Kanäle an. Sie können jedoch den Befehl `DISPLAY CLUSQMGR` verwenden, um die Attribute von automatisch definierten Clustersenderkanälen zu prüfen.

Verwenden Sie den Befehl `DISPLAY CHSTATUS`, um den Status von Clustersender- oder Clusterempfängerkanälen anzuzeigen. Dieser Befehl zeigt den Status sowohl von manuell definierten Kanälen als auch von automatisch definierten Kanälen an.

Die entsprechenden PCFs lauten `MQCMD_CHANGE_CHANNEL`, `MQCMD_COPY_CHANNEL`, `MQCMD_CREATE_CHANNEL` und `MQCMD_INQUIRE_CHANNEL`.

## Weglassen des Werts **CONNNAME** in einer **CLUSRCVR**-Definition

Unter Umständen können Sie den Wert **CONNNAME** in einer **CLUSRCVR**-Definition weglassen. Unter **z/OS** darf der Wert **CONNNAME** jedoch nicht weggelassen werden.

**Multi** Unter Multiplatforms ist die Angabe des TCP/IP-Verbindungsnamensparameters eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM MQ automatisch einen Verbindungsnamen, wobei der Standardport vorausgesetzt und die aktuelle IP-Adresse des Systems verwendet wird. Sie können die Standardportnummer überschreiben, aber weiterhin die aktuelle IP-Adresse des Systems verwenden. Lassen Sie für jeden Verbindungsnamen den IP-Namen leer und übergeben Sie die Portnummer in runden Klammern; Beispiel:

(1415)

Der generierte **CONNNAME** wird nicht als alphanumerischer DNS-Hostname angegeben, sondern immer im dezimalen Format mit Punkten als Trennzeichen (IPv4) oder im hexadezimalen Format (IPv6).

Diese Funktion ist hilfreich, wenn Ihre Systeme Dynamic Host Configuration Protocol (DHCP) verwenden. Wenn Sie für den Parameter **CONNNAME** in einem **CLUSRCVR**-Kanal keinen Wert angeben, müssen Sie die **CLUSRCVR**-Definition nicht ändern. DHCP ordnet Ihnen eine neue IP-Adresse zu.

Wenn Sie für den Parameter **CONNNAME** in der **CLUSRCVR**-Definition ein Leerzeichen angeben, generiert IBM MQ einen Wert für **CONNNAME** aus der IP-Adresse des Systems. Nur der generierte **CONNNAME** wird in den Repositorys gespeichert. Andere Warteschlangenmanager im Cluster wissen nicht, dass für **CONNNAME** ursprünglich ein Leerzeichen angegeben war.

Wenn Sie den Befehl `DISPLAY CLUSQMGR` ausgeben, wird der generierte **CONNNAME** angezeigt. Wenn Sie jedoch den Befehl `DISPLAY CHANNEL` aus dem lokalen Warteschlangenmanager ausgeben, können Sie sehen, dass für **CONNNAME** ein Leerzeichen angegeben ist.

Wenn der Warteschlangenmanager aufgrund von DHCP gestoppt und mit einer anderen IP-Adresse neu gestartet wird, generiert IBM MQ **CONNNAME** erneut und aktualisiert die Repositorys entsprechend.

### Zugehörige Konzepte

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

**z/OS** Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter **z/OS** erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### Zugehörige Verweise

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Mit den Befehlen `SUSPEND QMGR` und `RESUME QMGR` können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## **Befehle für die Warteschlangendefinition**

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

### **Die Befehle DEFINE QLOCAL, DEFINE QREMOTE und DEFINE QALIAS**

Die Clusterattribute in den Befehlen `DEFINE QLOCAL`, `DEFINE QREMOTE` und `DEFINE QALIAS` und die drei äquivalenten `ALTER`-Befehle sind:

#### **CLUSTER**

Gibt den Namen des Clusters an, dem die Warteschlange zugeordnet ist.

#### **CLUSNL**

Gibt eine Namensliste mit Clusternamen an.

#### **DEFBIND**

Gibt die Bindung an, die verwendet werden soll, wenn `MQOO_BIND_AS_Q_DEF` im `MQOPEN`-Aufruf einer Anwendung angegeben wird. Folgende Optionen stehen für dieses Attribut zur Verfügung:

- Wird `DEFBIND(OPEN)` angegeben, wird die Warteschlangenkennung beim Öffnen der Warteschlange an eine bestimmte Instanz der Clusterwarteschlange gebunden. `DEFBIND(OPEN)` ist die Standardeinstellung für dieses Attribut.
- Wird `DEFBIND(NOTFIXED)` angegeben, so wird die Warteschlangenkennung nicht an eine Instanz der Clusterwarteschlange gebunden.
- Wird `DEFBIND(GROUP)` angegeben, kann eine Anwendung fordern, dass alle Nachrichten einer Nachrichtengruppe an dieselbe Zielinstanz übergeben werden.

Wenn mehrere Warteschlangen mit demselben Namen in einem Warteschlangenmanagercluster zugänglich gemacht werden, können Anwendungen auswählen, ob alle Nachrichten aus dieser Anwendung an eine einzelne Instanz (`MQOO_BIND_ON_OPEN`) gesendet werden sollen, damit der Lastausgleichsalgorithmus für jede Nachricht das am besten geeignete Ziel auswählen kann (`MQOO_BIND_NOT_FIXED`), oder ob eine Anwendung anfordern kann, dass eine "Gruppe" von Nachrichten vollständig derselben Zielinstanz zugeordnet wird (`MQOO_BIND_ON_GROUP`). Der Lastausgleich wird zwischen Nachrichtengruppen erneut ausgeführt (ohne dass ein `MQCLOSE` und ein `MQOPEN` für die Warteschlange erforderlich sind).

Wenn Sie in einer Warteschlangendefinition `DEFBIND` angeben, wird die Warteschlange mit einem der Attribute `MQBND_BIND_ON_OPEN`, `MQBND_BIND_NOT_FIXED` oder `MQBND_BIND_ON_GROUP` definiert. Bei der Verwendung von Gruppen mit Clustern muss entweder `MQBND_BIND_ON_OPEN` oder `MQBND_BIND_ON_GROUP` angegeben werden.

Es wird empfohlen, für das Attribut `DEFBIND` in allen Instanzen einer Clusterwarteschlange denselben Wert anzugeben. Da `MQOO_BIND_ON_GROUP` in IBM WebSphere MQ 7.1 ein neues Attribut ist,

darf dieses Attribut nicht verwendet werden, wenn eine der Anwendungen, die diese Warteschlange öffnen, eine Verbindung mit Warteschlangenmanagern aus IBM WebSphere MQ 7.0.1 oder früheren Versionen herstellt.

#### **CLWLRANK**

Weist zu Auslastungsmanagementzwecken Warteschlangen einen Rangfolgefaktor zu. Der CLWLRANK-Parameter wird von Modellwarteschlangen nicht unterstützt. Der Algorithmus für Clusterauslastung wählt eine Zielwarteschlange mit dem höchsten Rang aus. Standardmäßig wird CLWLRANK für alle Warteschlangen auf null gesetzt.

Wenn es sich bei dem Zielort um einen Warteschlangenmanager in einem anderen Cluster handelt, können Sie den Rang jedes zwischengeschalteten Gateway-Warteschlangenmanagers am Schnittpunkt benachbarter Cluster festlegen. Nachdem unter den zwischengeschalteten Warteschlangenmanagern eine Rangfolge festgelegt wurde, wählt der Algorithmus für Clusterauslastung ordnungsgemäß jeweils den Zielwarteschlangenmanager aus, der sich näher am Zielort befindet.

Das Gleiche gilt für Aliaswarteschlangen. Die Auswahl nach Rangfolge erfolgt noch, bevor der Kanalstatus überprüft wird. Daher stehen auch nicht verfügbare Warteschlangenmanager für die Auswahl zur Verfügung. Dadurch wird es einer Nachricht ermöglicht, durch das Netz weitergeleitet zu werden, anstatt dass sie zwischen zwei möglichen Zieladressen wählen muss (wie es bei der Prioritätseinstellung der Fall wäre). Wenn nun ein Kanal nicht in dem Bereich gestartet wird, den sein Rang angibt, wird die Nachricht nicht an den nächsthöheren Rang weitergeleitet, sondern wartet, bis ein Kanal für diese Zieladresse verfügbar ist (d. h. die Nachricht wird in der Übertragungswarteschlange gespeichert).

#### **CLWLPRTY**

Weist zu Auslastungsmanagementzwecken Warteschlangen einen Prioritätsfaktor zu. Der Algorithmus für Clusterauslastung wählt eine Zielwarteschlange mit der höchsten Priorität aus. Standardmäßig wird für alle Warteschlangen die Priorität null angegeben.

Bei zwei möglichen Zielwarteschlangen können Sie dieses Attribut dazu verwenden, eine Zieladresse als Ausfallsicherung für die andere Zieladresse einzurichten. Die Prioritätsauswahl erfolgt nach der Überprüfung des Kanalstatus. Alle Nachrichten werden an die Warteschlange mit der höchsten Priorität gesendet, es sei denn, der Status des Kanals zu dieser Zieladresse ist nicht so günstig wie der Status anderer Kanäle zu anderen Zieladressen. Dies bedeutet, dass nur die am besten zugänglichen Zieladressen für die Auswahl zur Verfügung stehen. Dadurch erfolgt eine Priorisierung unter mehreren Zieladressen, die alle verfügbar sind.

#### **CLWLUSEQ**

Gibt das Verhalten einer MQPUT-Operation für eine Warteschlange an. Dieser Parameter gibt das Verhalten einer MQPUT-Operation an, wenn die Zielwarteschlange über eine lokale Instanz und mindestens eine ferne Clusterinstanz verfügt (außer wenn der MQPUT-Aufruf von einem Clusterkanal stammt). Dieser Parameter ist nur für lokale Warteschlangen gültig.

Folgende Werte sind möglich: QMGR (das Verhalten entspricht dem, was im CLWLUSEQ-Parameter der Warteschlangenmanagerdefinition angegeben ist), ANY (der Warteschlangenmanager behandelt die lokale Warteschlange zum Zweck der Lastverteilung wie eine weitere Instanz der Clusterwarteschlange) und LOCAL (die lokale Warteschlange ist das einzige Ziel der MQPUT-Operation, wobei angegeben wird, dass für die lokale Warteschlange das Einreihen aktiviert ist). Das Verhalten der MQPUT-Operation ist vom Algorithmus für das Clusterauslastungsmanagement abhängig.

### **Befehle DISPLAY QUEUE und DISPLAY QCLUSTER**

Die Attribute in den Befehlen DEFINE QLOCAL, DEFINE QREMOTE und DEFINE QALIAS gelten ebenso für den DISPLAY QUEUE-Befehl.

Um Informationen zu Clusterwarteschlangen anzuzeigen, geben Sie im Befehl DISPLAY QUEUE einen Warteschlangentyp wie QCLUSTER oder das Schlüsselwort CLUSINFO an, oder verwenden Sie den Befehl DISPLAY QCLUSTER.

Die Befehle DISPLAY QUEUE und DISPLAY QCLUSTER geben den Namen des Warteschlangenmanagers zurück, der die Warteschlange enthält (oder die Namen aller Warteschlangenmanager, wenn es sich um mehrere Instanzen einer Warteschlange handelt). Außerdem werden der Systemname für jeden Warte-

schlangenmanager, der die Warteschlange enthält, der dargestellte Warteschlangentyp sowie Datum und Uhrzeit des Zeitpunkts zurückgegeben, zu dem die Definition für den lokalen Warteschlangenmanager verfügbar gemacht wurde. Diese Daten werden mithilfe der Attribute CLUSQMGR, QMID, CLUSQT, CLUSDATE und CLUSTIME zurückgegeben.

Der Systemname für den Warteschlangenmanager (QMID) ist ein eindeutiger, systemgenerierter Name für den Warteschlangenmanager.

Sie haben die Möglichkeit, eine Clusterwarteschlange zu definieren, die gleichzeitig eine gemeinsam genutzte Warteschlange ist. Beispiel: Unter z/OS können Sie Folgendes definieren:

```
DEFINE QLOCAL(MYQUEUE) CLUSTER(MYCLUSTER) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

Die entsprechenden PCFs sind MQCMD\_CHANGE\_Q, MQCMD\_COPY\_Q, MQCMD\_CREATE\_Q und MQCMD\_INQUIRE\_Q.

## Zugehörige Konzepte

### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

## Zugehörige Verweise

### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

### DISPLAY CLUSQMGR

Verwenden Sie den Befehl DISPLAY CLUSQMGR, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Mit den Befehlen SUSPEND QMGR und RESUME QMGR können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

### REFRESH CLUSTER

Geben Sie den Befehl REFRESH CLUSTER in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## DISPLAY CLUSQMGR

Verwenden Sie den Befehl DISPLAY CLUSQMGR, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

Wenn Sie diesen Befehl von einem Warteschlangenmanager mit einem vollständigen Repository aufrufen, gelten die zurückgegebenen Informationen für alle Warteschlangenmanager im Cluster. Andernfalls gelten die zurückgegebenen Informationen nur für die jeweils betroffenen Warteschlangenmanager, d. h. für jeden Warteschlangenmanager, an den versucht wurde, eine Nachricht zu senden, und für jeden Warteschlangenmanager mit einem vollständigen Repository.

Die Daten beinhalten die meisten Kanalattribute sowohl für Clustersender- als auch für Clusterempfängerkanäle. Zusätzlich können die folgenden Attribute angezeigt werden:

**CHANNEL**

Der Name des Clusterempfängerkanals für den Warteschlangenmanager.

**CLUSTRATE**

Das Datum, an dem die Definition dem lokalen Warteschlangenmanager zur Verfügung gestellt wurde.

**CLUSTER**

Gibt an, zu welchen Clustern der Warteschlangenmanager gehört.

**CLUSTIME**

Zeit, zu der die Definition für den lokalen Warteschlangenmanager verfügbar war.

**DEFTYPE**

Gibt die Definition des Warteschlangenmanagers an. Für DEFTYPE sind folgende Werte möglich:

**CLUSSDR**

Ein Clustersenderkanal wurde auf dem lokalen Warteschlangenmanager administrativ definiert, aber noch nicht vom Ziel-Warteschlangenmanager erkannt. Dieser Status tritt auf, wenn ein lokaler Warteschlangenmanager einen manuellen Clustersenderkanal definiert hat, der empfangende Warteschlangenmanager die Clusterinformationen jedoch nicht akzeptiert hat. Ursache hierfür kann sein, dass der Kanal aufgrund fehlender Verfügbarkeit oder eines Fehlers in der Clustersenderkonfiguration nicht aufgebaut wurde, zum Beispiel aufgrund einer fehlenden Übereinstimmung für die Eigenschaft CLUSTER in der Sender- und Empfängerdefinition. Hierbei handelt es sich um einen vorübergehenden Zustand oder Fehlerstatus, der untersucht werden sollte.

**CLUSSDRA**

Dieser Wert gibt einen automatisch erkannten Cluster-Warteschlangenmanager an, lokal ist kein Clustersenderkanal definiert. Dies ist der DEFTYPE-Wert für Cluster-Warteschlangenmanager, für die der lokale Warteschlangenmanager nicht über eine lokale Konfiguration verfügt, aber über die er informiert ist. Beispiel:

- Wenn der lokale Warteschlangenmanager ein Warteschlangenmanager für ein vollständiges Repository ist, muss der DEFTYPE-Wert für alle Teilrepository-Warteschlangenmanager im Cluster verwendet werden.
- Wenn der lokale Warteschlangenmanager ein Teilrepository ist, kann dies der Host einer Clusterwarteschlange sein, der von diesem lokalen Warteschlangenmanager oder von einem zweiten Warteschlangenmanager für ein vollständiges Repository verwendet wird, mit dem dieser Warteschlangenmanager zusammenarbeiten soll.

Wenn der DEFTYPE-Wert CLUSSDRA lautet und sowohl der lokale als auch der ferne Warteschlangenmanager vollständige Repositories für den benannten Cluster sind, ist die Konfiguration nicht korrekt, da ein lokal definierter Clustersenderkanal zum Umwandeln in den DEFTYPE-Wert CLUSSDRB definiert sein muss.

**CLUSSDRB**

Ein Clustersenderkanal wurde auf dem lokalen Warteschlangenmanager administrativ definiert und als gültiger Clusterkanal vom Ziel-Warteschlangenmanager erkannt. Dies ist der erwartete DEFTYPE-Wert für einen manuell konfigurierten Warteschlangenmanager für ein vollständiges Repository eines Teilrepository-Warteschlangenmanagers. Außerdem muss er der DEFTYPE-Wert für jeden CLUSQMGR von einem vollständigen Repository zu einem anderen vollständigen Repository im Cluster sein. Manuelle Clustersenderkanäle dürfen nicht zu Teilrepositories oder von einem Teilrepository-Warteschlangenmanager zu mehr als einem vollständigen Repository konfiguriert werden. Wenn der DEFTYPE-Wert CLUSSDRB in einer dieser Situationen festgestellt wird, muss er überprüft und korrigiert werden.

**CLUSRCVR**

Administrativ definiert als Clusterempfängerkanal für den lokalen Warteschlangenmanager. Gibt den lokalen Warteschlangenmanager im Cluster an.

**Anmerkung:** Informationen zum Ermitteln, welche CLUSQMGRs Warteschlangenmanager für vollständige Repositories für den Cluster sind, finden Sie unter der Eigenschaft QMTYPE.



Weitere Informationen zum Definieren von Clusterkanälen finden Sie unter [Clusterkanäle](#).

### **QMTYPE**

Gibt an, ob der Warteschlangenmanager über ein vollständiges Repository oder nur über ein Teilrepository verfügt.

### **STATUS**

Gibt den Status des Clustersenderkanals für diesen Warteschlangenmanager an.

### **SUSPEND**

Gibt an, ob der Warteschlangenmanager ausgesetzt wurde.

### **VERSION**

Die Version der IBM MQ -Installation, der der Clusterwarteschlangenmanager zugeordnet ist.

Die Version hat das Format VVRRMMFF:

- VV: Version
- RR: Release
- MM: Wartungsstufe
- FF: Fixversion

### **XMITQ**

Die vom Warteschlangenmanager verwendete Clusterübertragungswarteschlange.

Siehe dazu den Befehl `DISPLAY QCLUSTER`. Er wird kurz in `DISPLAY QUEUE` und im Abschnitt [Die Befehle DISPLAY QUEUE und DISPLAY QCLUSTER von „Befehle für die Warteschlangendefinition“ auf Seite 157](#) beschrieben. Beispiele zur Verwendung des Befehls `DISPLAY QCLUSTER` finden Sie in den Informationen zu `"DISPLAY QCLUSTER"` und `"DIS QCLUSTER"`.

### **Zugehörige Konzepte**

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

#### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### **Zugehörige Verweise**

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Mit den Befehlen `SUSPEND QMGR` und `RESUME QMGR` können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## Zugehörige Informationen

MQSC-Befehl **DISPLAY CLUSQMGR**

## Befehle **SUSPEND QMGR** und **RESUME QMGR** im Cluster

Mit den Befehlen **SUSPEND QMGR** und **RESUME QMGR** können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

Wenn ein Warteschlangenmanager auf einem Cluster ausgesetzt wird, empfängt er keine Nachrichten auf den gehosteten Clusterwarteschlangen, falls es eine verfügbare Warteschlange mit dem gleichen Namen auf einem alternativen Warteschlangenmanager im Cluster gibt. Werden Nachrichten jedoch explizit an diesen Warteschlangenmanager gerichtet oder ist die Zielwarteschlange nur auf diesem Warteschlangenmanager verfügbar, dann werden die Nachrichten weiterhin an diesen Warteschlangenmanager übertragen.

Der fortgesetzte Empfang eingehender Nachrichten im ausgesetzten Warteschlangenmanager kann verhindert werden, indem die Clusterempfängerkanäle für diesen Cluster gestoppt werden. Zum Stoppen der Clusterempfängerkanäle für einen Cluster verwenden Sie den Befehl **SUSPEND QMGR** im **FORCE**-Modus.

### Zugehörige Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

 Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### Zugehörige Verweise

Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

DISPLAY CLUSQMGR

Verwenden Sie den Befehl **DISPLAY CLUSQMGR**, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

REFRESH CLUSTER

Geben Sie den Befehl **REFRESH CLUSTER** in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

### Zugehörige Informationen

SUSPEND QMGR

RESUME QMGR

Verwalten eines Warteschlangenmanagers

## REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

Es gibt drei Formen dieses Befehls:

### **REFRESH CLUSTER(Clustername) REPOS(NO)**

Die Standardeinstellung. Der Warteschlangenmanager hat Kenntnis von allen lokal definierten Clusterwarteschlangenmanagern und Clusterwarteschlangen sowie allen Clusterwarteschlangenmanagern, bei denen es sich um vollständige Repositorys handelt. Wenn der Warteschlangenmanager darüber hinaus ein vollständiges Repository für den Cluster ist, hat er ebenso Kenntnis von den anderen Clusterwarteschlangenmanagern innerhalb des Clusters. Alles andere wird von der lokalen Kopie des Repositorys gelöscht und aus den anderen vollständigen Repositorys innerhalb des Clusters wiederhergestellt. Clusterkanäle werden nicht gestoppt, wenn `REPOS(NO)` verwendet wird. Ein vollständiges Repository verwendet seine `CLUSDR`-Kanäle, um den Rest des Clusters darüber zu informieren, dass seine Aktualisierung abgeschlossen ist.

### **REFRESH CLUSTER(Clustername) REPOS(YES)**

Zusätzlich zum Standardverhalten werden Objekte, die Clusterwarteschlangenmanager für vollständige Repositorys darstellen, ebenfalls aktualisiert. Diese Option kann nicht verwendet werden, wenn der Warteschlangenmanager ein vollständiges Repository ist; wird sie verwendet, schlägt der Befehl fehl und es wird der Fehler `AMQ9406/CSQX406E` protokolliert. Wenn es sich um ein vollständiges Repository handelt, müssen Sie es zuerst so ändern, dass es kein vollständiges Repository für den betreffenden Cluster ist. Die Adresse des vollständigen Repositorys wird anhand der manuell definierten `CLUSDR`-Definitionen wiederhergestellt. Nach Abschluss der Aktualisierung mit Angabe von `REPOS(YES)` kann der Warteschlangenmanager bei Bedarf geändert werden, sodass er wieder ein vollständiges Repository ist.

### **REFRESH CLUSTER(\*)**

Aktualisiert den Warteschlangenmanager in allen Clustern, denen er angehört. In Verbindung mit `REPOS(YES)` zwingt `REFRESH CLUSTER(*)` den Warteschlangenmanager dazu, die Suche nach vollständigen Repositorys in den lokalen `CLUSDR`-Definitionen erneut durchzuführen. Die Suche wird auch dann durchgeführt, wenn der `CLUSDR`-Kanal den Warteschlangenmanager mit mehreren Clustern verbindet.

**Anmerkung:** Bei großen Clustern kann die Verwendung des Befehls **REFRESH CLUSTER** während der Ausführung des Clusters und anschließend in 27-Tage-Intervallen, wenn die Clusterobjekte Statusaktualisierungen automatisch an alle interessierten WS-Manager senden, unterbrechend für den Cluster sein. Nähere Informationen hierzu erhalten Sie im Abschnitt [Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken](#).

### **Zugehörige Konzepte**

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

#### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### **Zugehörige Verweise**

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl **DISPLAY CLUSQMGR**, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Mit den Befehlen **SUSPEND QMGR** und **RESUME QMGR** können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

#### **Zugehörige Informationen**

Clustering: Best Practices für REFRESH CLUSTER verwenden

## **RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen**

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

Sie können den Befehl **RESET CLUSTER** nur in Warteschlangenmanagern für ein vollständiges Repository ausgeben. Je nachdem, ob Sie mit Namen oder ID auf den Warteschlangenmanager verweisen, kann der Befehl zwei Formen annehmen.

1. 

```
RESET CLUSTER( clustername
) QMNAME( qmname ) ACTION(FORCEREMOVE) QUEUES(NO)
```

2. 

```
RESET CLUSTER( clustername
) QMID( qmid ) ACTION(FORCEREMOVE) QUEUES(NO)
```

Sie können nicht beide Optionen QMNAME und QMID angeben. Wenn Sie QMNAME verwenden, wird der Befehl nicht ausgeführt, wenn sich im Cluster mehrere Warteschlangenmanager mit diesem Namen befinden. Verwenden Sie QMID anstelle von QMNAME, um sicherzustellen, dass der Befehl **RESET CLUSTER** ausgeführt wird.

Bei der Angabe von QUEUES(NO) in einem **RESET CLUSTER**-Befehl handelt es sich um die Standardeinstellung. Durch die Angabe von QUEUES(YES) werden Verweise auf Clusterwarteschlangen entfernt, die zu den Warteschlangenmanagern des Clusters gehören. Diese Verweise werden zusätzlich zur Entfernung des Warteschlangenmanagers aus dem Cluster selbst entfernt.

Die Verweise werden entfernt, auch wenn der Clusterwarteschlangenmanager im Cluster nicht sichtbar ist, weil dessen Entfernung zuvor ohne die Option QUEUES erzwingen wurde.

Sie können den Befehl **RESET CLUSTER** verwenden, wenn beispielsweise ein Warteschlangenmanager gelöscht wurde, aber immer noch über für den Cluster definierte Clusterempfängerkanäle verfügt. Anstatt darauf zu warten, dass IBM MQ diese Definitionen entfernt (was automatisch geschieht), können Sie den Befehl **RESET CLUSTER** ausgeben, um die Definitionen bereits früher zu löschen. Alle anderen Warteschlangenmanager im Cluster werden daraufhin informiert, dass dieser Warteschlangenmanager nicht mehr verfügbar ist.

Wenn ein Warteschlangenmanager vorübergehend beschädigt ist, können Sie die anderen Warteschlangenmanager im Cluster benachrichtigen, bevor sie versuchen, Nachrichten an diesen Warteschlangenmanager zu senden. Der Befehl **RESET CLUSTER** entfernt den beschädigten Warteschlangenmanager. Später, wenn der beschädigte Warteschlangenmanager wieder funktioniert, verwenden Sie den Befehl **REFRESH CLUSTER**, um die Wirkung von **RESET CLUSTER** umzukehren und den Warteschlangenmana-

ger an den Cluster zurückzugeben. Wenn sich der Warteschlangenmanager in einem Publish/Subscribe-Cluster befindet, müssen Sie alle erforderlichen Proxy-Subskriptionen wiederherstellen. Siehe [Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster](#).

**Anmerkung:** Bei großen Clustern kann die Verwendung des Befehls **REFRESH CLUSTER** während der Ausführung des Clusters und anschließend in 27-Tage-Intervallen, wenn die Clusterobjekte Statusaktualisierungen automatisch an alle interessierten WS-Manager senden, unterbrechend für den Cluster sein. Nähere Informationen hierzu erhalten Sie im Abschnitt [Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken](#).

Die Verwendung des Befehls **RESET CLUSTER** ist die einzige Möglichkeit, automatisch definierte Cluster-senderkanäle zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter normalen Umständen benötigen. Das IBM Support Center rät Ihnen möglicherweise dazu, den Befehl abzusetzen, um die Clusterinformationen zu bereinigen, die in den Clusterwarteschlangenmanagern gespeichert sind. Verwenden Sie diesen Befehl nicht als verkürztes Verfahren, um Warteschlangenmanager aus einem Cluster zu entfernen. Die korrekte Vorgehensweise beim Entfernen eines Warteschlangenmanagers aus einem Cluster wird im Abschnitt [Warteschlangenmanager aus einem Cluster entfernen](#) beschrieben.

Repositorys speichern Daten nur für 90 Tage. Daher kann ein Warteschlangenmanager, dessen Entfernung erzwungen wurde, nach diesem Zeitraum die Verbindung zum Cluster wiederherstellen. Der Warteschlangenmanager stellt die Verbindung automatisch wieder her, es sei denn, er wurde gelöscht. Wenn Sie einen Warteschlangenmanager davon abhalten wollen, die Verbindung zu einem Cluster wiederherzustellen, müssen Sie entsprechende Sicherheitsmaßnahmen ergreifen.

Mit Ausnahme von **DISPLAY CLUSQMGR** agieren alle Clusterbefehle asynchron. Befehle zum Ändern von Objektattributen unter Einbeziehung von Clustering aktualisieren das Objekt und senden eine Anforderung an den Repositoryprozessor. Befehle für die Arbeit mit Clustern werden im Hinblick auf die Syntax überprüft und es wird eine Anforderung an den Repositoryprozessor gesendet.

Die an den Repositoryprozessor gesendeten Anforderungen werden gemeinsam mit den von anderen Clustermitgliedern empfangenen Clusteranforderungen asynchron verarbeitet. Die Verarbeitung nimmt möglicherweise eine längere Zeit in Anspruch, wenn die Anforderungen an den gesamten Cluster gesendet werden müssen, um festzustellen, ob sie erfolgreich ausgeführt wurden oder nicht.

## Zugehörige Konzepte

### [Lastausgleich in Clustern](#)

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

### [Asynchrones Verhalten von Clusterbefehlen unter z/OS](#)

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

## Zugehörige Verweise

### [Befehle für Warteschlangenmanagerdefinitionen](#)

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

### [Befehle für die Kanaldefinition](#)

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

### [Befehle für die Warteschlangendefinition](#)

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

### [DISPLAY CLUSQMGR](#)

Verwenden Sie den Befehl **DISPLAY CLUSQMGR**, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

### [Befehle SUSPEND QMGR und RESUME QMGR im Cluster](#)

Mit den Befehlen `SUSPEND QMGR` und `RESUME QMGR` können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER (Cluster zurücksetzen)

## Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Passende Zieladressen werden vom Verwaltungsalgorithmus für die Clusterauslastung auf Basis der Verfügbarkeit von Warteschlangenmanager und Warteschlange ausgewählt, sowie auf Basis einer Reihe spezifischer Attribute zur Clusterauslastung, die Warteschlangenmanagern, Warteschlangen und Kanälen zugeordnet sind. Diese Attribute werden in den Unterthemen beschrieben.

**Anmerkung:** Die Kanalattribute für die Clusterauslastung müssen Sie auf den Ziel-Warteschlangenmanagern auf den Clusterempfängerkanälen angeben. Lastausgleichsattribute, die Sie auf den zugehörigen Clustersenderkanälen angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie unter Clusterkanäle.

Wenn die Konfiguration der spezifischen Attribute zur Clusterauslastung sich nicht wie erwartet verhält, untersuchen Sie die Einzelheiten zur Auswahl eines Warteschlangenmanagers durch den Algorithmus. Siehe „Algorithmus für das Clusterauslastungsmanagement“ auf Seite 176. Wenn die Ergebnisse dieses Algorithmus nicht Ihren Bedürfnissen entsprechen, können Sie ein Benutzerexitprogramm für Clusterauslastung schreiben und diesen Exit verwenden, um Nachrichten zu der Warteschlange Ihrer Wahl im Cluster weiterzuleiten. Weitere Informationen finden Sie unter Exits für Clusterauslastung schreiben und kompilieren.

### Zugehörige Konzepte

 Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### Zugehörige Verweise

Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

Befehle `SUSPEND QMGR` und `RESUME QMGR` im Cluster

Mit den Befehlen `SUSPEND QMGR` und `RESUME QMGR` können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## Warteschlangenattribut **CLWLPRTY**

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Warteschlangenattribut **CLWLPRTY** legen Sie die Priorität einer Zielwarteschlange fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Anforderungen werden dem Warteschlangenmanager mit der höchsten Priorität zugestellt, während Warteschlangenmanager mit niedrigerer Priorität als Reserve fungieren. Fällt der Warteschlangenmanager mit der höchsten Priorität aus, übernimmt der Warteschlangenmanager mit der nächsthöheren Priorität, sofern er zur Verfügung steht.

IBM MQ ruft die Priorität von Warteschlangenmanagern erst nach der Überprüfung des Kanalstatus ab. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

### **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**

Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

Warteschlangenattribut **CLWLRANK**

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Warteschlangenattribut **CLWLUSEQ**

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

Warteschlangenmanagerattribut **CLWLUSEQ**

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

Warteschlangenmanagerattribut **CLWLMRUC**

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLPRTY

Das Kanalattribut CLWLPRTY legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

#### Kanalattribut NETPRTY

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## **Warteschlangenattribut CLWLRANK**

Das Warteschlangenattribut CLWLRANK gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Das Warteschlangenattribut CLWLRANK sollten Sie verwenden, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Wenn CLWLRANK gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen.

Stellen Sie sich zum Beispiel vor, Sie hätten zwei identisch konfigurierte Gateway-Warteschlangenmanager, durch die die Verfügbarkeit eines Gateways verbessert werden soll. Dazu hätten Sie an den Gateways Cluster-Aliaswarteschlangen für eine im Cluster definierte lokale Warteschlange konfiguriert. Sollte die lokale Warteschlange nicht zur Verfügung stehen, soll eine Nachricht an einem der Gateways zurückgehalten werden, bis die Warteschlange wieder zur Verfügung steht. Um die Cluster-Aliaswarteschlange an einem der beiden Gateways zurückzustellen, muss die lokale Warteschlange mit einem höheren Rang definiert sein als die Aliaswarteschlangen am Gateway.

Hätte die lokale Warteschlange den gleichen Rang wie die Aliaswarteschlangen und stünde die lokale Warteschlange nicht zur Verfügung, würde die Nachricht zwischen den Gateways hin und her übertragen werden. Sobald der erste Gateway-Warteschlangenmanager feststellt, dass die lokale Warteschlange nicht verfügbar ist, würde er die Nachricht an das andere Gateway senden. Das andere Gateway würde erneut versuchen, die Nachricht der lokalen Warteschlange zuzustellen. Stünde die lokale Warteschlange nach wie vor nicht zur Verfügung, würde die Nachricht zurück zum ersten Gateway geleitet werden. Die Nachricht wird auf diese Weise zwischen den beiden Gateways hin und her gesendet, bis die lokale Warteschlange wieder zur Verfügung steht. Erhält die lokale Warteschlange aber einen höheren Rang, würde die Nachricht selbst bei Nichtverfügbarkeit der Warteschlange keinem Ziel mit niedrigerem Rang zugestellt werden.

IBM MQ ruft den Warteschlangenrang noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Warteschlangen zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

Wenn Sie hingegen das Prioritätsattribut 'CLWLPRTY' verwenden, wählt IBM MQ nur zwischen verfügbaren Zielen aus. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

### **Zugehörige Konzepte**

#### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.



## Zugehörige Verweise

### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut CLWLPRTY gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut CLWLUSEQ auf QMGR gesetzt ist.

### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut CLWLMRUC legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet CLWLMRUC zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

### Kanalattribut CLWLPRTY

Das Kanalattribut CLWLPRTY legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

### Kanalattribut CLWLWGHT

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

### Kanalattribut NETPRTY

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

Das Warteschlangenattribut CLWLUSEQ ist nur für lokale Warteschlangen gültig. Es wird nur angewendet, wenn die Nachricht von einer Anwendung oder von einem Kanal eingereicht wird, der kein Clusterkanal ist.

### **LOCAL**

Das einzige Ziel von MQPUT ist die lokale Warteschlange, vorausgesetzt, diese ist PUT-aktiviert. Das Verhalten der MQPUT-Operation ist vom Clusterauslastungsmanagement abhängig.

### **QMGR**

Das Verhalten wird durch das Warteschlangenmanagerattribut CLWLUSEQ bestimmt.

### **ANY**

Hinsichtlich der Lastverteilung behandelt MQPUT die lokale Warteschlange genauso wie alle anderen Instanzen der Warteschlange im Cluster.

## Zugehörige Konzepte

### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

## Zugehörige Verweise

### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

#### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLPRTY

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von **CLUSDDR**- und **CLUSRCVR**-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

#### Kanalattribut NETPRTY

Das Kanalattribut **NETPRTY** gibt die Priorität eines **CLUSRCVR**-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## **Warteschlangenmanagerattribut CLWLUSEQ**

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

Das Warteschlangenattribut **CLWLUSEQ** ist nur für lokale Warteschlangen gültig. Es wird nur angewendet, wenn die Nachricht von einer Anwendung oder von einem Kanal eingereicht wird, der kein Clusterkanal ist.

### **LOCAL**

Das einzige Ziel von **MQPUT** ist die lokale Warteschlange. **LOCAL** ist der Standardwert.

### **ANY**

Hinsichtlich der Lastverteilung behandelt **MQPUT** die lokale Warteschlange genauso wie alle anderen Instanzen der Warteschlange im Cluster.

### **Zugehörige Konzepte**

#### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

#### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

#### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLPRTY

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von **CLUSDR**- und **CLUSRCVR**-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

#### Kanalattribut NETPRTY

Das Kanalattribut **NETPRTY** gibt die Priorität eines **CLUSRCVR**-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## **Warteschlangenmanagerattribut CLWLMRUC**

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

Der anfängliche Standardwert ist 999 999 999.

### **Zugehörige Konzepte**

#### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

#### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

#### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

#### Kanalattribut CLWLPRTY

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

#### Kanalattribut NETPRTY

Das Kanalattribut **NETPRTY** gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## **Kanalattribut CLWLPRTY**

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Kanalattribut **CLWLPRTY** legen Sie die Reihenfolge der Priorität der verfügbaren Clusterziele fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Nachrichten gehen an den Warteschlangenmanager mit dem Kanal mit der höchsten Priorität. Ist dieser Kanal nicht verfügbar, gehen die Nachrichten an den Warteschlangenmanager mit der nächsthöchsten Priorität. Warteschlangenmanager mit niedrigerer Priorität fungieren als Reserve.

Vor der Priorisierung der Kanäle prüft IBM MQ den Kanalstatus. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt Clusterkanäle.
- 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 (Sicherheit) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.
- Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherheitsziel 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 Sicherheit.

### **Zugehörige Konzepte**

#### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

#### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

#### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut CLWLUSEQ auf QMGR gesetzt ist.

#### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut CLWLMRUC legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet CLWLMRUC zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

#### Kanalattribut NETPRTY

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

### **Kanalattribut CLWLRANK**

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Verwenden Sie das Kanalattribut **CLWLRANK**, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Sie steuern die Auswahl des Zielorts, indem Sie den Rang der Kanäle festlegen, die einen Warteschlangenmanager am Schnittpunkt der Cluster mit den Gateway-Warteschlangenmanagern verbinden.

Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen. Ein Beispiel: Nachrichten kommen an einem Gateway-Warteschlangenmanager an, der sie an einen der beiden Warteschlangenmanager weiterleiten kann, die Kanäle mit dem Rang 1 und 2 verwenden. Die Nachrichten werden automatisch an den Warteschlangenmanager gesendet, der durch einen Kanal mit dem höchsten Rang verbunden ist; in diesem Fall ist dies der Warteschlangenmanagerkanal mit dem Rang 2.

IBM MQ ruft den Rang von Kanälen noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Kanäle zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

#### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Wenn Sie auch das Prioritätsattribut **CLWLPRTY** verwenden, wählt IBM MQ zwischen verfügbaren Zielen aus. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

#### **Zugehörige Konzepte**

##### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

## Zugehörige Verweise

### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut CLWLPRTY gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut CLWLRANK gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut CLWLUSEQ gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut CLWLUSEQ auf QMGR gesetzt ist.

### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut CLWLMRUC legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet CLWLMRUC zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

### Kanalattribut CLWLPRTY

Das Kanalattribut CLWLPRTY legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

### Kanalattribut CLWLWGHT

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

### Kanalattribut NETPRTY

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## Kanalattribut CLWLWGHT

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

Verwenden Sie CLWLWGHT, um mehr Nachrichten an Server mit einer größeren Verarbeitungskapazität zu senden. Je stärker ein Kanal gewichtet ist, desto mehr Nachrichten werden über diesen Kanal versendet.

### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Standardeinstellung des Parameters CLWLWGHT ist 50. Wird diese Einstellung für einen Kanal geändert, so wird der Lastausgleich abhängig von der Häufigkeit, die die einzelnen Kanäle für eine an eine Clusterwarteschlange gesendete Nachricht ausgewählt wurden. Weitere Informationen finden Sie unter [„Algorithmus für das Clusterauslastungsmanagement“](#) auf Seite 176.

## Zugehörige Konzepte

### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

## Zugehörige Verweise

### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

#### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

#### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLPRTY

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut NETPRTY

Das Kanalattribut **NETPRTY** gibt die Priorität eines **CLUSRCVR**-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

### **Kanalattribut NETPRTY**

Das Kanalattribut **NETPRTY** gibt die Priorität eines **CLUSRCVR**-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Verwenden Sie das Attribut **NETPRTY**, um ein Netz als primäres Netz und ein weiteres Netz als Sicherungsnetz festzulegen. Bei einer Gruppe von Kanälen mit gleichem Rang wird durch Clustering der Pfad mit der höchsten Priorität ausgewählt, wenn mehrere Pfade zur Verfügung stehen.

Üblicherweise wird das Kanalattribut **NETPRTY** verwendet, um zwischen Netzen zu unterscheiden, die die gleichen Zielorte verbinden, aber unterschiedliche Aufwände verursachen oder unterschiedliche Geschwindigkeiten aufweisen.

**Anmerkung:** Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).

### **Zugehörige Konzepte**

#### Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

#### Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

#### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

#### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLPRTY

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von **CLUSDDR**- und **CLUSRCVR**-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

## **Algorithmus für das Clusterauslastungsmanagement**

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

Immer, wenn ein Ziel ausgewählt werden muss, wird der Workload-Management-Algorithmus ausgeführt:

- Sie wird an dem Punkt verwendet, an dem eine Clusterwarteschlange unter Verwendung der Option **MQOO\_BIND\_ON\_OPEN** geöffnet wird.
- Sie wird verwendet, wenn eine Nachricht in eine Clusterwarteschlange eingereicht wird, wenn sie mit **MQOO\_BIND\_NOT\_FIXED** geöffnet wird.
- Sie wird bei jedem Start einer neuen Nachrichtengruppe verwendet, wenn **MQOO\_BIND\_ON\_GROUP** verwendet wird, um eine Clusterwarteschlange zu öffnen.
- Er wird für Topic-Host-Routing bei jedem Veröffentlichen einer Nachricht für ein Clusterthema (Cluster-Topic) verwendet. Wenn es sich beim lokalen Warteschlangenmanager nicht um den Host für dieses Thema handelt, wird mit dem Algorithmus ein Host-Warteschlangenmanager ausgewählt, über den die Nachricht gesendet wird.

Im folgenden Abschnitt wird der Algorithmus für das Workload-Management beschrieben, der zur Bestimmung der Zielorte der in Clusterwarteschlangen befindlichen Nachrichten verwendet wird. Dessen Regeln werden durch die Einstellungen der folgenden Attribute für Warteschlangen, Warteschlangenmanager und Kanäle beeinflusst:



Tabelle 32. Attribute für das Clusterauslastungsmanagement

Warteschlangen	Warteschlangenmanager	Kanäle
<ul style="list-style-type: none"> <li>• <u>CLWLPRTY</u><sup>1</sup></li> <li>• <u>CLWLRANK</u><sup>1</sup></li> <li>• <u>CLWLUSEQ</u><sup>1</sup></li> <li>• PUT / PUB</li> </ul>	<ul style="list-style-type: none"> <li>• <u>CLWLUSEQ</u><sup>1</sup></li> <li>• <u>CLWLMRUC</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>CLWLPRTY</u></li> <li>• <u>CLWLRANK</u></li> <li>• <u>CLWLWGHT</u></li> <li>• <u>NETPRTY</u></li> </ul>

Zunächst erstellt der Warteschlangenmanager mithilfe der folgenden beiden Verfahren eine Liste der möglichen Ziele:

- Abgleich des `ObjectName` und `ObjectQmgrName` des Ziels mit Warteschlangenmanager-Aliasdefinitionen, die in denselben Clustern wie der Warteschlangenmanager verwendet werden.
- Ermitteln eindeutiger Routen (also Kanäle) zu einem Warteschlangenmanager, der eine Warteschlange mit dem Namen `ObjectName` bereitstellt, die sich in einem der Cluster befindet, zu denen der Warteschlangenmanager gehört.

Der Algorithmus durchläuft die folgenden Regeln, um die Liste der möglichen Ziele zu reduzieren.

1. Ferne Instanzen von Warteschlangen oder Themen oder ferne CLUSRCVR-Kanäle, die keinen Cluster mit dem lokalen Warteschlangenmanager gemeinsam nutzen, werden ausgeschlossen.
2. Wenn eine Warteschlange oder ein Themenname angegeben ist, werden ferne CLUSRCVR-Kanäle ausgeschlossen, die sich nicht in demselben Cluster wie die Warteschlange oder das Thema befinden.

**Anmerkung:** Alle verbleibenden Warteschlangen, Themen und Kanäle werden in dieser Phase dem Exit für Clusterauslastung zur Verfügung gestellt, wenn dieser konfiguriert ist.

3. Alle Kanäle zu Warteschlangenmanagern oder Warteschlangenmanager-Aliasnamen, die einen CLWLRANK-Wert aufweisen, der kleiner ist als der maximale Rang aller verbleibenden Kanäle oder Warteschlangenmanager-Aliasnamen, werden ausgeschlossen.
4. Alle Warteschlangen (keine Warteschlangenmanager-Aliasnamen) mit einem CLWLRANK-Wert, der kleiner als der maximale Rang aller verbleibenden Warteschlangen ist, werden ausgeschlossen.
5. Wenn mehr als eine Instanz einer Warteschlange, eines Themas oder eines Warteschlangenmanager-Aliasnamens übrig bleibt und für mindestens eine davon 'pub put' aktiviert ist, werden alle Instanzen ausgeschlossen, auf denen 'put' inaktiviert ist.

**Anmerkung:** Wenn ausschließlich Instanzen übrig bleiben, auf denen 'put' inaktiviert ist, können nur noch Abfrageoperationen erfolgreich ausgeführt werden, alle anderen Operationen schlagen mit `MQRC_CLUSTER_PUT_INHIBITED` fehl.

6. Wenn bei der Auswahl einer Warteschlange die verbleibende Gruppe der Warteschlangen die lokale Instanz der Warteschlange enthält, wird in der Regel die lokale Instanz verwendet. Die lokale Instanz der Warteschlange wird verwendet, wenn eine der folgenden Bedingungen zutrifft:
  - Das Attribut `CLWLUSEQ` der Warteschlange ist auf `LOCAL` gesetzt.
  - Die beiden folgenden Aussagen treffen zu:
    - Das Attribut `CLWLUSEQ` der Warteschlange ist auf `QMGR` gesetzt.
    - Das Attribut `CLWLUSEQ` des Warteschlangenmanagers ist auf `LOCAL` gesetzt.
  - Die Nachricht wurde nicht direkt von einer lokalen Anwendung eingereicht, sondern über einen Clusterkanal empfangen.
  - Auf lokal definierte Warteschlangen, die mit `CLWLUSEQ(ANY)` definiert sind oder diese Einstellung aus dem Warteschlangenmanager übernehmen, treffen folgende Punkte innerhalb der breiter gesteckten Gruppe von geltenden Bedingungen zu:

<sup>1</sup> Dieses Attribut gilt bei Auswahl einer Clusterwarteschlange, nicht bei Auswahl eines Themas.

- Die lokale Warteschlange wird auf der Basis der statusder lokal definierten CLUSRCVR-Kanäle in demselben Cluster wie die Warteschlange ausgewählt. Dieser Status wird mit dem Status der CLUSSDR-Kanäle verglichen, die die Nachricht an remote definierte gleichnamige Warteschlangen leiten würden.  
 Beispiel: Es gibt einen CLUSRCVR im gleichen Cluster wie die Warteschlange. Dieser CLUSRCVR hat den Status STOPPING, während die anderen Warteschlangen desselben Namens im Cluster den Status RUNNING oder INACTIVE haben. In diesem Fall werden die fernen Kanäle gewählt und die lokale Warteschlange wird nicht verwendet.
  - Die lokale Warteschlange wird auf der Basis der numbeɀvon CLUSRCVR-Kanälen ausgewählt (im Vergleich zu CLUSSDR-Kanälen mit demselben Status), die die Nachricht an fern definierte Warteschlangen mit demselben Namen senden würden.  
 Beispiel: Es gibt vier CLUSRCVR-Kanäle im selben Cluster wie die Warteschlange und einen CLUSSDR-Kanal. Alle Kanäle haben denselben Status, d. h. entweder INACTIVE oder RUNNING. Daher stehen fünf Kanäle zur Auswahl und zwei Instanzen der Warteschlange. Vier Fünftel (80 Prozent) der Nachrichten gehen an die lokale Warteschlange.
7. Wenn mehr als ein Warteschlangenmanager übrig bleibt und mindestens einer davon nicht vorübergehend gesperrt ist, werden alle anderen ausgeschlossen, die vorübergehend gesperrt sind.
  8. Wenn mehrere ferne Instanzen einer Warteschlange oder eines Themas verbleiben, werden alle Kanäle mit den Status INACTIVE oder RUNNING aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
    - MQCHS\_INACTIVE
    - MQCHS\_RUNNING
  9. Wenn keine ferne Instanz einer Warteschlange oder eines Themas verbleibt, werden alle rev="d1">Kanäle mit den Status BINDING, INITIALIZING, STARTING oder STOPPING aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
    - MQCHS\_BINDING
    - MQCHS\_INITIALIZING
    - MQCHS\_STARTING
    - MQCHS\_STOPPING
  10. Wenn keine Instanz einer Warteschlange oder eines Themas verbleibt, werden alle Kanäle eingeschlossen, für die ein erneuter Verbindungsversuch durchgeführt wird. Die Statuskonstante wird aufgeführt:
    - MQCHS\_RETRYING
  11. Wenn keine ferne Instanz einer Warteschlange oder eines Themas verbleibt, werden alle Kanäle mit den Status REQUESTING, PAUSED oder STOPPED aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
    - MQCHS\_REQUESTING
    - MQCHS\_PAUSED
    - MQCHS\_STOPPED
    - MQCHS\_SWITCHING
  12. Wenn mehr als eine ferne Instanz einer Warteschlange oder eines Themas auf einem beliebigen Warteschlangenmanager verbleibt, werden Kanäle mit dem höchsten NETPRTY-Wert für die einzelnen Warteschlangenmanager ausgewählt.
  13. Alle verbleibenden Kanäle und Warteschlangenmanager-Aliasnamen mit Ausnahme der Kanäle und Aliasnamen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen. Verbleiben dabei Warteschlangenmanager-Aliasnamen, so bleiben die Kanäle zu diesem Warteschlangenmanager eingeschlossen.
  14. Bei Auswahl einer Warteschlange:

- Alle Warteschlangen mit Ausnahme der Warteschlangen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen, Kanäle bleiben aber erhalten.

15. Die verbleibenden Kanäle werden dann durch Entfernen der Kanäle mit den niedrigsten MQWDR.DestSeqNumber-Werten auf nicht mehr als die maximal zulässige Anzahl der zuletzt verwendeten Kanäle (CLWLMRUC) reduziert.

**Anmerkung:** Interne Steuernachrichten für das Cluster werden ggf. mit dem gleichen Algorithmus für die Clusterauslastung gesendet.

Nachdem die Liste der gültigen Ziele berechnet wurden, wird ein Lastausgleich für die Nachrichten mithilfe der folgenden Logik durchgeführt:

- Wenn mehrere ferne Instanzen eines Ziels verbleiben und für alle Kanäle zu diesem Ziel die Option CLWLWGHT auf die Standardeinstellung 50 gesetzt ist, wird der am längsten nicht mehr verwendete Kanal ausgewählt. Dies entspricht etwa einem Umlaufstil beim Durchführen des Lastausgleichs, wenn mehrere ferne Instanzen vorhanden sind.
- Wenn mehrere Instanzen eines Ziels verbleiben und für mindestens einen Kanal zu diesen Warteschlangen die Option CLWLWGHT auf eine andere Einstellung als die Standardeinstellung gesetzt ist (selbst wenn alle Kanäle einen übereinstimmenden, nicht standardmäßigen Wert haben), ist die Weiterleitung abhängig von den relativen Gewichtungen jedes Kanals und der Gesamtzahl, mit der jeder Kanal zuvor beim Senden von Nachrichten ausgewählt wurde.
- Beim Beobachten der Verteilung von Nachrichten für eine einzelne Clusterwarteschlange mit mehreren Instanzen führt dies scheinbar zu einer nicht ausgeglichenen Verteilung in einer Untergruppe der Warteschlangeninstanzen. Dies liegt daran, dass nicht nur die Nachrichtenübertragung für diese Warteschlange ausgeglichen wird, sondern die historische Verwendung jedes Clustersenderkanals aus diesem Warteschlangenmanager berücksichtigt wird. Wenn dieses Verhalten nicht erwünscht ist, führen Sie einen der folgenden Schritte aus:
  - Setzen Sie CLWLWGHT auf allen Clusterempfängerkanälen auf 50, wenn eine gleichmäßige Verteilung erforderlich ist.
  - oder, falls bestimmte Warteschlangeninstanzen anders als andere gewichtet werden müssen, definieren Sie diese Warteschlangen in einem dedizierten Cluster mit definierten dedizierten Clusterempfängerkanälen. Durch diese Aktion wird der Lastausgleich dieser Warteschlangen vom Lastausgleich anderer Warteschlangen im Cluster eingegrenzt.
- Die historischen Daten, mit denen der Lastausgleich für die Kanäle durchgeführt wird, werden zurückgesetzt, wenn ein beliebiges Attribut zur Clusterauslastung der verfügbaren Clusterempfängerkanäle geändert oder der Status eines Clusterempfängerkanals verfügbar wird. Durch Änderungen an den Auslastungsattributen manuell definierter Clustersenderkanäle werden die historischen Daten nicht zurückgesetzt.
- Wenn Sie erwägen, die Logik des Exits für die Clusterauslastung zu verwenden, wird der Kanal mit dem niedrigsten Wert für MQWDR.DestSeqFactor ausgewählt. Bei jeder Auswahl des Kanals wird dieser Wert um etwa  $1000/CLWLWGHT$  erhöht. Falls mehrere Kanäle mit diesem Wert vorliegen, wird einer der Kanäle mit dem niedrigsten Wert für MQWDR.DestSeqNumber ausgewählt.

Die Verteilung der Benutzernachrichten erfolgt nicht immer exakt nach diesen Regeln, da auch durch die Clusterverwaltung Nachrichten über Kanäle übertragen werden. Dies führt zu einer zunächst ungleichmäßigen Verteilung der Benutzernachrichten, die sich erst nach einer gewissen Zeit stabilisiert. Aufgrund der Mischung aus Verwaltungs- und Benutzernachrichten sollten Sie sich nicht auf die genaue Verteilung von Nachrichten während des Lastausgleichs verlassen.

### Zugehörige Verweise

Warteschlangenattribut CLWLPRTY

Das Warteschlangenattribut CLWLPRTY gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Warteschlangenattribut CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Warteschlangenattribut CLWLUSEQ

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

#### Warteschlangenmanagerattribut CLWLUSEQ

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf **QMGR** gesetzt ist.

#### Warteschlangenmanagerattribut CLWLMRUC

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

#### Kanalattribut CLWLPRTY

Das Kanalattribut **CLWLPRTY** legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

#### Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt den Rang der Kanäle für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

#### Kanalattribut CLWLWGHT

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von **CLUSDR**- und **CLUSRCVR**-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

#### Kanalattribut NETPRTY

Das Kanalattribut **NETPRTY** gibt die Priorität eines **CLUSRCVR**-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

## Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

Für beide Befehle **REFRESH CLUSTER** und **RESET CLUSTER** gilt, dass die Nachricht **CSQM130I** an den Ausgeber des Befehls gesendet wird, um anzuzeigen, dass eine Anforderung gesendet wurde. Darauf folgt die Nachricht **CSQ9022I**, um anzuzeigen, dass der Befehl erfolgreich ausgeführt wurde, d. h. dass eine Anforderung gesendet wurde. Sie gibt nicht an, dass die Clusteranforderung selbst erfolgreich ausgeführt wurde.

Alle eventuell auftretenden Fehler werden an die z/OS-Konsole des Systems ausgegeben, auf dem der Kanalinitiator aktiv ist; sie werden nicht an den Ausgeber des Befehls gemeldet.

Das asynchrone Verhalten steht im Kontrast zu Kanalbefehlen (**CHANNEL**). Sofort wird eine Nachricht ausgegeben, die anzeigt, dass ein Kanalbefehl akzeptiert wurde. Zu einem späteren Zeitpunkt wird, sobald der Befehl ausgeführt worden ist, eine Nachricht an den Ausgeber des Befehls gesendet, um eine normale oder abnormale Ausführung des Befehls anzuzeigen.

### Zugehörige Konzepte

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

### Zugehörige Verweise

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Mit den Befehlen `SUSPEND QMGR` und `RESUME QMGR` können Sie die eingehende Clusteraktivität auf diesem Warteschlangenmanager vorübergehend verringern, bevor Sie die Wartung auf diesem Warteschlangenmanager ausführen, und diese anschließend wiederherstellen.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

#### **Zugehörige Informationen**

Prüfen, ob asynchrone Befehle für verteilte Netze abgeschlossen sind

## Kanalprogramme

In diesem Abschnitt werden die verschiedenen Arten von Kanalprogrammen (MCAs) behandelt, die bei den Kanälen verwendet werden können.

Die Namen der MCAs werden in der folgenden Tabelle aufgeführt.

<b>Programmname</b>	<b>Verbindungsrichtung</b>	<b>Kommunikation</b>
amqrmppa		Alle
runmqlsr	Eingehend	Alle
amqcrs6a	Eingehend	LU 6.2
amqcrsta	Eingehend	TCP
runmqchl	Ausgehend	Alle
runmqchi	Ausgehend	Alle

Die Steuerbefehle 'runmqlsr' (Ausführung des IBM MQ-Empfangsprogramms), 'runmqchl' (Ausführung des IBM MQ-Kanals) und 'runmqchi' (Ausführung des IBM MQ-Kanalinitialtors) können in der Befehlszeile eingegeben werden.

'amqcrsta' wird für TCP-Kanäle auf Systemen unter UNIX and Linux mit 'inetd' aufgerufen, auf denen kein Empfangsprogramm gestartet wurde.

'amqcrs6a' wird bei Verwendung von LU6.2 als Transaktionsprogramm aufgerufen.

## Umgebungsvariablen

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

## Verwendungsbeispiele

- **Linux** **UNIX** Auf UNIX and Linux -Systemen verwenden Sie `export [environment variable]=value`.
- **Windows** Verwenden Sie auf Windows -Systemen `Set [environment variable]=value`.
- **IBM i** Verwenden Sie auf IBM i-Systemen Folgendes: `ADDENVVAR ENVVAR(environment variable) VALUE(xx)`

### AMQ\_BAD\_COMMS\_DATA\_FDCS

Diese Umgebungsvariable ist wirksam, wenn sie auf einen beliebigen Wert gesetzt wird.

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

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

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

### V 9.0.0.9 AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS

Diese Umgebungsvariable ist wirksam, wenn sie auf einen beliebigen Wert gesetzt wird.

Wenn IBM MQ die ursprüngliche Datenübertragung nicht erkennt, wenn versucht wird, einen Nicht-IBM MQ -Client mit einem IBM MQ -TCP/IP-Listener zu verbinden, führt dies dazu, dass der Warteschlangenmanager eine `AMQ9207E` -Fehlermeldung in die Fehlerprotokolle des Warteschlangenmanagers schreibt. Ein FDC-Datensatz (FDC = Failure Data Capture) wird ebenfalls geschrieben. Sie können die Generierung dieser Diagnosedateien mit der Umgebungsvariablen **AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS** unterdrücken. Wenn **AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS** auf einen beliebigen Wert gesetzt ist (z. B. TRUE), weist dies IBM MQ an, keine FFSTs zu generieren, wenn Fehlermeldungen `AMQ9207E` im ursprünglichen Kommunikationsfluss gemeldet werden. Um wirksam zu sein, sollte die Umgebungsvariable vor dem Start des Warteschlangenmanagers und der Listenerprozesse gesetzt werden.

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

### Multi V 9.0.0.6 AMQ\_CONVEBCDICNEWLINE

Ab IBM MQ 9.0.0 Fix Pack 6 können Sie mit dieser Umgebungsvariablen angeben, wie IBM MQ ein landessprachliches EBCDIC-Zeichen in ASCII-Format konvertieren soll. Die Umgebungsvariable hat dieselben Werte wie das Attribut **ConvEBCDICNewline** von `mqs.ini`, d. h. `NL_TO_LF`, `TABLE` oder `ISO` (siehe Alle Warteschlangenmanager). Sie können beispielsweise statt des Zeilengruppenattributs **ConvEBCDICNewline** die Umgebungsvariable **AMQ\_CONVEBCDICNEWLINE** verwenden, um **ConvEBCDICNewline**-Funktionalität auf der Clientseite in Situationen bereitzustellen, in denen die Datei `mqs.ini` nicht verwendet werden kann.

Wenn sowohl das Zeilengruppenattribut als auch die Umgebungsvariable festgelegt sind, hat das Zeilengruppenattribut Vorrang.

### V 9.0.3 AMQ\_DIAGNOSTIC\_MSG\_SEVERITY

Wenn ab IBM MQ 9.0.3 die Umgebungsvariable **AMQ\_DIAGNOSTIC\_MSG\_SEVERITY** für einen IBM MQ-Prozess auf 1 gesetzt ist, wird die Nachrichtenbewertung als einzelnes alphabetisches Zeichen in Großbuchstaben an die Nachrichtennummer angehängt, wenn der IBM MQ-Prozess eine Nachricht in ein Fehlerprotokoll oder in die Konsole schreibt.

V 9.0.4 Ab IBM MQ 9.0.4 ist das von **AMQ\_DIAGNOSTIC\_MSG\_SEVERITY** aktivierte Verhalten standardmäßig voreingestellt. Sie können dieses Verhalten inaktivieren, indem Sie die Umgebungsvariable auf 0 setzen.

### Z/OS LTS AMQ\_DMPMQCFG\_QSGDISP\_DEFAULT

Ab IBM MQ 9.0.0 Fix Pack 9 können Sie mit dem Befehl **dmpmqcfcfg** die Disposition eines Warteschlangenmanagers in einer Gruppe mit gemeinsamer Warteschlange abrufen.

Details zu den Werten, die Sie mit dieser Umgebungsvariablen verwenden können, finden Sie in **dmpmqcfcfg**.

### V 9.0.0.9 AMQ\_LDAP\_TRACE

Ab IBM MQ 9.0.0 Fix Pack 9 ist es möglich, den LDAP-Client-Trace ein- bzw. auszuschalten, ohne auch den Warteschlangenmanager zu stoppen oder zu starten, wenn die Umgebungsvariable **AMQ\_LDAP\_TRACE** auf einen Wert ungleich null gesetzt ist. Weitere Informationen finden Sie im Abschnitt [Dynamisches Tracing für LDAP-Clientbibliothekscode aktivieren](#).

### AMQ\_MQS\_INI\_LOCATION

Auf Systemen mit UNIX and Linux können Sie die für die Datei `mqs.ini` verwendete Position ändern, indem Sie in dieser Variablen die Position der Datei `mqs.ini` festlegen. Diese Variable muss auf Systemebene festgelegt werden.

### AMQ\_NO\_IPV6

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

### AMQ\_REVERSE\_COMMIT\_ORDER

Über diese Variable wird ein Warteschlangenmanager so konfiguriert, dass in einer XA-Transaktion die Änderung des IBM MQ-Warteschlangenmanagers nach Abschluss der entsprechenden Datenbankaktualisierung festgeschrieben wird.

Definieren Sie **AMQ\_REVERSE\_COMMIT\_ORDER** nicht, ohne zuvor das im Abschnitt [Isolationsstufe](#) beschriebene Szenario zu lesen und es sich verständlich zu machen.

### V 9.0.0.1 V 9.0.2

### AMQ\_SSL\_ALLOW\_DEFAULT\_CERT

Ab IBM MQ 9.0.0 Fix Pack 1 und IBM MQ 9.0.2 kann eine Anwendung, wenn die Umgebungsvariable **AMQ\_SSL\_ALLOW\_DEFAULT\_CERT** nicht festgelegt ist, nur dann eine Verbindung zu einem Warteschlangenmanager mit einem persönlichen Zertifikat im Client-Keystore herstellen, wenn das Zertifikat die Bezeichnung `ibmwebspheremquserid` enthält. Wenn die Umgebungsvariable **AMQ\_SSL\_ALLOW\_DEFAULT\_CERT** festgelegt ist, erfordert das Zertifikat nicht den Kennsatznamen `ibmwebspheremquserid`. Das heißt, das Zertifikat, das zum Herstellen einer Verbindung zu einem Warteschlangenmanager verwendet wird, kann ein Standardzertifikat sein, sofern ein Standardzertifikat im Schlüsselrepository vorhanden ist und das Schlüsselrepository kein persönliches Zertifikat mit dem Präfix `ibmwebspheremquserid` enthält. Weitere Informationen finden Sie im technischen Hinweis [Specifying the userid in the SSL certificate label for an MQ client](#).

Der Wert 1 lässt die Verwendung eines Standardzertifikats zu.

Ab IBM MQ 8.0 kann eine Anwendung anstelle der Umgebungsvariablen **AMQ\_SSL\_ALLOW\_DEFAULT\_CERT** die Einstellung **CertificateLabel** der SSL-Zeilengruppe in der Datei `mqclicent.ini` verwenden. Weitere Informationen finden Sie in den Abschnitten [Digitale Zertifikatsbezeichnungen - Anforderungen](#) und [SSL-Zeilengruppe in der Clientkonfigurationsdatei](#).

### V 9.0.4 V 9.0.0.2

## **AMQ\_SSL\_LDAP\_SERVER\_VERSION**

Ab IBM MQ 9.0.0 Fix Pack 2 und IBM MQ 9.0.4 kann diese Variable verwendet werden, um sicherzustellen, dass LDAP v2 oder LDAP v3 von IBM MQ-Verschlüsselungskomponenten in Fällen verwendet wird, in denen CRL-Server erfordern, dass eine bestimmte Version des LDAP-Protokolls verwendet wird.

Setzen Sie die Variable auf einen geeigneten Wert für die Umgebung, die zum Starten des Warteschlangenmanagers oder Kanals verwendet wird. Setzen Sie `AMQ_SSL_LDAP_SERVER_VERSION=2`, um die Verwendung von LDAP v2 anzufordern. Legen Sie `AMQ_SSL_LDAP_SERVER_VERSION=3` fest, um die Verwendung von LDAP v3 anzufordern.

Diese Variable hat keine Auswirkungen auf LDAP-Verbindungen, die vom IBM MQ-Warteschlangenmanager für die Benutzerauthentifizierung oder die Benutzerberechtigung erstellt werden.

## **GMQ\_MQ\_LIB**

Wenn sowohl der IBM MQ MQI client als auch der IBM MQ-Server auf Ihrem System installiert sind, werden MQAX-Anwendungen standardmäßig auf dem Server ausgeführt. Um MQAX auf dem Client auszuführen, müssen die Clientbinding-Bibliotheken in der Umgebungsvariable 'GMQ\_MQ\_LIB' angegeben sein, wie beispielsweise `GMQ_MQ_LIB=mqic.dll`. Für eine reine Client-Installation ist es nicht erforderlich, die Umgebungsvariable `GMQ_MQ_LIB` zu setzen. Wenn diese Variable nicht gesetzt ist, versucht IBM MQ `amqzst.dll` zu laden. Wenn diese DLL nicht vorhanden ist, wie bei reinen Client-Installationen, versucht IBM MQ, `mqic.dll` zu laden.

## **HOME**

Diese Variable enthält den Namen des Verzeichnisses, in dem nach der Datei `mqclient.ini` gesucht wird. In dieser Datei sind Konfigurationsinformationen enthalten, die von IBM MQ MQI clients auf folgenden Plattformen verwendet werden:

-  IBM i
-  UNIX
-  Linux

## **HOMEDRIVE und HOMEPATH**

Beide Variablen müssen gesetzt werden, damit sie verwendet werden können. Sie enthalten den Namen des Verzeichnisses, in dem nach der Datei `mqclient.ini` gesucht wird. In dieser Datei sind Konfigurationsinformationen enthalten, die von IBM MQ MQI clients auf Windows-Systemen verwendet werden.

## **LDAP\_BASEDN**

Die Umgebungsvariable, die für das Ausführen eines LDAP-Beispielprogramms erforderlich ist. Sie gibt den Basis-DN für die Verzechnissuche an.

## **LDAP\_HOST**

Eine optionale Umgebungsvariable für das Ausführen eines LDAP-Beispielprogramms. Sie gibt den Namen des Hosts an, auf dem der LDAP-Server ausgeführt wird. Wenn keine Angabe erfolgt, wird als Wert standardmäßig der lokale Host verwendet.

## **LDAP\_VERSION**

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

## **MQAPI\_TRACE\_LOGFILE**

Das Beispiel-API-Exitprogramm erstellt einen MQI-Trace zu einer benutzerdefinierten Datei mit einem Präfix, das in der Umgebungsvariablen `MQAPI_TRACE_LOGFILE` definiert ist.

## **MQCCSID**

Gibt die zu verwendende ID des codierten Zeichensatzes an und überschreibt die native CCSID der Anwendung.



## **MQCERTLABL**

Definiert die Zertifikatsbezeichnung.

## **MQCERTVPOL**

Bestimmt den verwendeten Typ der Prüfung der Gültigkeit des Zertifikats:

### **ANY**

Es wird eine Zertifikatsprüfrichtlinie verwendet, die von der zugrunde liegenden Secure-Sockets-Bibliothek unterstützt wird. Dies ist die Standardeinstellung.

### **RFC5280**

Es wird nur eine Zertifikatsprüfung verwendet, die dem Standard RFC 5280 entspricht.

## **MQCHLLIB**

Gibt den Verzeichnispfad zu der Datei an, die die Definitionstabelle für Clientkanal enthält. Die Datei wird auf dem Server erstellt, kann aber auf die IBM MQ MQI client-Workstation kopiert werden.

## **MQCHLTAB**

MQCHLTAB gibt den Namen der Datei an, in der die Definitionstabelle für Clientkanal (CCDT) enthalten ist. Der Standarddateiname ist AMQCLCHL.TAB.

## **MQS\_IPC\_HOST**

Wenn IBM MQ-Dateien gemeinsam genutzt werden und der für myHostName generierte Wert Probleme verursacht, setzen Sie myHostName mithilfe der Umgebungsvariablen MQS\_IPC\_HOST.

## **MQCLNTCF**

Verwenden Sie diese Umgebungsvariable, um den Pfad der Datei `mqClient.ini` zu ändern.

## **MQ\_CHANNEL\_SUPPRESS\_INTERVAL**

Gibt das Zeitintervall in Sekunden an, in dem die mit `MQ_CHANNEL_SUPPRESS_MSGS` definierten Nachrichten unterdrückt werden sollen, bevor sie in das Fehlerprotokoll geschrieben werden, zusammen mit der Häufigkeit, mit der eine Nachricht während des angegebenen Zeitintervalls auftreten darf, bevor sie unterdrückt werden. Der Standardwert lautet 60,5. Dies bedeutet, dass alle weiteren Vorkommen einer bestimmten Nachricht nach den ersten fünf Vorkommen der betreffenden Nachricht in einem 60-Sekunden-Intervall unterdrückt werden. Weitere Informationen finden Sie im Abschnitt [Kanalfehlernachrichten in Fehlerprotokollen auf Multiplatforms unterdrücken](#).

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


## **MQ\_CHANNEL\_SUPPRESS\_MSGS**

Gibt IBM MQ-Kanalfehlernachrichten an, die nur in der angegebenen Häufigkeit in das Fehlerprotokoll geschrieben werden sollen, mit der diese Nachrichten während des in `MQ_CHANNEL_SUPPRESS_INTERVAL` definierten Zeitintervalls auftreten dürfen, bevor sie unterdrückt werden, bis dieses Zeitintervall abläuft. Weitere Informationen finden Sie im Abschnitt [Kanalfehlernachrichten in Fehlerprotokollen auf Multiplatforms unterdrücken](#).

Die Umgebungsvariable `MQ_CHANNEL_SUPPRESS_MSGS` ist mit `SuppressMessage` in der Datei `qm.ini` vergleichbar, obwohl sie anders angegeben ist.

## **MQ\_CONNECT\_TYPE**

Verwenden Sie auf folgenden Plattformen diese Umgebungsvariable zusammen mit dem Binding-Typ, der im Feld 'Optionen' für die MQNO-Struktur angegeben ist, die bei MQCONNX-Aufrufen verwendet wird. Siehe [MQCONNX-Umgebungsvariable](#).

-  Windows
-   UNIX and Linux
-  IBM i

## **V9.0.0.1**

## **MQ\_CROSS\_QUEUE\_ORDER\_ALL**

Wenn Sie die Umgebungsvariable `MQ_CROSS_QUEUE_ORDER_ALL` auf einen Wert ungleich null setzen, wird die Reihenfolge der Nachrichteneinreihung in einer Arbeitseinheit beibehalten. Dies bedeu-

tet Folgendes: Wenn Nachrichten in einer Arbeitseinheit (UOW) in mehrere Warteschlangen eingereicht werden (beispielsweise in Q1 und danach in Q2) werden sie bei einer MQCMIT-Ausgabe in der Reihenfolge übermittelt und zur Verfügung gestellt, in der sie EINGEREIHT wurden.

In einer Umgebung mit mehreren Warteschlangenmanagern muss MQ\_CROSS\_QUEUE\_ORDER\_ALL sowohl auf der sendenden als auch auf der empfangenden Seite vor dem Start der einzelnen Warteschlangenmanager vorhanden sein und einen nicht leeren Wert aufweisen.

### **MQ\_FILE\_PATH**

Bei der Installation des Laufzeitpakets auf der Windows-Plattform wird eine neue Umgebungsvariable mit der Bezeichnung MQ\_FILE\_PATH konfiguriert. Diese Umgebungsvariable enthält die gleichen Daten wie der folgende Schlüssel in der Windows-Registry:

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

### **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.

### **MQ\_JAVA\_DATA\_PATH**

Gibt das Verzeichnis für Protokoll und Traceausgabe an.

### **MQ\_JAVA\_INSTALL\_PATH**

Gibt das Verzeichnis an, in dem IBM MQ classes for Java installiert sind, wie in den IBM MQ classes for Java-Installationsverzeichnissen angezeigt.

### **MQ\_JAVA\_LIB\_PATH**

Gibt das Verzeichnis an, in dem die IBM MQ classes for Java-Bibliotheken gespeichert werden. Diese Umgebungsvariable wird von einigen Scripts verwendet, die mit IBM MQ classes for Java bereitgestellt werden, wie z. B. IVTRun.

**Multi**

### **MQMAXERRORLOGSIZE**

MQMAXERRORLOGSIZE gibt die Größe des Fehlerprotokolls für den Warteschlangenmanager an, das an die Sicherungsposition kopiert wird.

### **MQNAME**

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

### **MQNOREMPOOL**

Wenn Sie diese Variable setzen, wird das Zusammenschließen von Kanälen inaktiviert, sodass Kanäle als Listener-Threads ausgeführt werden.

### **MQPSE\_TRACE\_LOGFILE**

Verwenden Sie diese Umgebungsvariable, wenn Sie das Publish-Exit-Beispielprogramm nutzen. Im Anwendungsprozess, für den ein Trace erstellt werden soll, beschreibt diese Umgebungsvariable, wohin die Tracedateien geschrieben werden sollen. Siehe [Publish-Exit-Beispielprogramm](#).

### **MQSERVER**

Mit der Umgebungsvariablen "MQSERVER" wird der Minimalkanal definiert. Sie können MQSERVER nicht verwenden, um einen TLS-Kanal oder einen Kanal mit Kanalexits zu definieren. MQSERVER gibt die Position des IBM MQ-Servers sowie das zu verwendende Kommunikationsverfahren an.

### **MQ\_SET\_NODELAYACK**

Wenn Sie diese Variable setzen, wird die verzögerte TCP-Bestätigung inaktiviert.

**AIX**

Wenn Sie diese Variable unter AIX setzen, wird die verzögerte TCP-Bestätigung durch einen setsockopt-Aufruf des Betriebssystems mit der Option TCP\_NODELAYAC inaktiviert. Diese Funktion wird nur unter AIX unterstützt; die Umgebungsvariable MQ\_SET\_NODELAYACK hat also nur unter AIX eine Auswirkung.

## **MQSNOAUT**

Wenn Sie diese Variable auf einen beliebigen Wert setzen, inaktiviert dies den Objektberechtigungsmanager (OAM) und verhindert Sicherheitsprüfungen. Die Variable MQSNOAUT wird nur bei der Erstellung eines Warteschlangenmanagers aktiv.

Um den OAM zu aktivieren, müssen Sie den Warteschlangenmanager und die Umgebungsvariable löschen und dann den Warteschlangenmanager erneut erstellen, ohne MQSNOAUT anzugeben.

## **MQSPREFIX**

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

## **MQSSLCRYP**

MQSSLCRYP enthält eine Parameterzeichenfolge, die Sie verwenden können, um die im System vorhandene Verschlüsselungshardware zu konfigurieren. Die gültigen Werte entsprechen den Werten für den Parameter SSLCRYP des Befehls ALTER QMGR.

## **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.

## **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 finden Sie im Abschnitt zum Parameter SSLKEYR des Befehls ALTER QMGR.

## **MQSSLPROXY**

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

## **MQSSLRESET**

MQSSLRESET gibt die Nummer der unverschlüsselten Bytes an, die auf einem TLS-Kanal vor der Neuvereinbarung des geheimen TLS-Schlüssels versendet und empfangen wurden.

## **MQS\_TRACE\_OPTIONS**

Verwenden Sie die Umgebungsvariable 'MQS\_TRACE\_OPTIONS', um die Funktionen für hohe Detail- und Parameterdichte für Traces einzeln festzulegen.

## **MQTCPTIMEOUT**

Diese Variable gibt an, wie lange IBM MQ auf eine TCP-Verbindungsanfrage wartet.

## **MQSUITEB**

Diese Variable gibt an, ob Suite B-kompatible Verschlüsselung genutzt werden soll. Wenn Suite B-kompatible Verschlüsselung verwendet wird, können Sie die Verschlüsselungsstärke angeben, indem Sie die Umgebungsvariable MQSUITEB auf einen der folgenden Werte setzen:

- KEINE
- 128\_BIT, 192\_BIT
- 128\_BIT
- 192\_BIT

## **ODQ\_MSG**

Wenn Sie eine Steuerroutine der Warteschlange für nicht zustellbare Nachrichten verwenden, die von RUNMQDLQ abweicht, können Sie die Quelle des Beispiels als Basis nutzen. Das Beispiel entspricht der Steuerroutine für nicht zustellbare Nachrichten, die innerhalb des Produkts bereitgestellt wird, verfügt aber über andere Trace- und Fehlermeldungsfunktionen. Verwenden Sie die Umgebungsvariable ODQ\_MSG, um den Namen der Datei zu setzen, die Fehler- und Informationsnachrichten enthält. Die bereitgestellte Datei hat die Bezeichnung `amqsdlq.msg`.

## **ODQ\_TRACE**

Wenn Sie eine Steuerroutine der Warteschlange für nicht zustellbare Nachrichten verwenden, die von RUNMQDLQ abweicht, können Sie die Quelle des Beispiels als Basis nutzen. Das Beispiel entspricht

der Steuerroutine für nicht zustellbare Nachrichten, die innerhalb des Produkts bereitgestellt wird, verfügt aber über andere Trace- und Fehlermeldungenfunktionen. Setzen Sie die Umgebungsvariable ODQ\_TRACE auf "YES" oder "yes", um die Traceerstellung zu aktivieren.

### **OMQ\_PATH**

Diese Umgebungsvariable gibt an, wo Sie den Symptombericht zum ersten Auftreten eines Fehlers finden, wenn Ihre IBM MQ-Automatisierungsklassen für ActiveX-Scripts fehlschlagen.

### **OMQ\_TRACE**

MQAX enthält eine Tracefunktion, mit der eine Serviceorganisation ermitteln kann, was geschieht, wenn ein Problem auftritt. Sie zeigt den Pfad an, der bei der Ausführung Ihres MQAX-Scripts verwendet wird. Wenn keine Probleme auftreten, inaktivieren Sie die Traceerstellung, um unnötiges Auslasten von Systemressourcen zu vermeiden. OMQ\_TRACE ist eine von drei Umgebungsvariablen, mit denen die Traceerstellung gesteuert wird. Wenn Sie einen beliebigen Wert für OMQ\_TRACE angeben, wird die Traceerstellung aktiviert. Auch wenn Sie den Wert "OFF" für OMQ\_TRACE angeben, ist die Traceerstellung immer noch aktiviert. Siehe [Trace verwenden](#)

### **OMQ\_TRACE\_PATH**

Eine der drei Umgebungsvariablen, mit denen die Traceerstellung gesteuert wird. Siehe [Trace verwenden](#)

### **OMQ\_TRACE\_LEVEL**

Eine der drei Umgebungsvariablen, mit denen die Traceerstellung gesteuert wird. Siehe [Trace verwenden](#)

### **ONCONFIG**

Der Name der Informix-Serverkonfigurationsdatei. Verwenden Sie beispielsweise auf Systemen mit UNIX and Linux folgende Angabe:

```
export ONCONFIG=onconfig.hostname_1
```

Verwenden Sie auf Systemen mit Windows folgende Angabe:

```
set ONCONFIG=onconfig.hostname_1
```

### **WCF\_TRACE\_ON**

Für den angepassten WCF-Kanal stehen zwei verschiedene Traceverfahren zur Verfügung. Die beiden Traceverfahren können unabhängig voneinander oder zusammen aktiviert werden. Jedes Verfahren liefert eine eigene Tracedatei, sodass zwei Traceausgabedateien erstellt werden, wenn beide Traceverfahren aktiviert wurden. Es gibt vier mögliche Kombinationen für das Aktivieren und Inaktivieren der beiden Traceverfahren. Neben diesen Kombinationen zum Aktivieren der WCF-Traceerstellung kann auch der XMS .NET-Trace unter Verwendung der Umgebungsvariablen WCF\_TRACE\_ON aktiviert werden. Weitere Informationen finden Sie unter [WCF-Tracekonfiguration und Tracedateinamen](#).

### **WMQSOAP\_HOME**

Verwenden Sie diese Umgebungsvariable, wenn Sie zusätzliche Konfigurationsschritte durchführen, nachdem die .NET SOAP over JMS-Service-Hostingumgebung ordnungsgemäß in IBM MQ installiert und konfiguriert wurde. Sie ist für lokale Warteschlangenmanager zugänglich. Weitere Informationen finden Sie unter [Beispiel eines WCF-Clients für .NET-Service, der von IBM MQ gehostet wird](#) und [Beispiel für WCF-Client für Axis Java-Service, der von IBM MQ gehostet wird](#).

Die Umgebungsvariable kann auch verwendet werden, wenn Sie IBM MQ-Webtransport für SOAP installieren. Weitere Informationen hierzu finden Sie unter [IBM MQ-Webtransport für SOAP installieren](#).

### **Zugehörige Informationen**

[IBM MQ-Umgebungsvariablen verwenden](#)

---

## **IBM i** **Übergreifende Kommunikationsjobs**

Die folgenden Jobs sind der übergreifenden Kommunikation unter IBM i zugeordnet. Die Namen werden in der folgenden Tabelle aufgeführt.

<i>Tabelle 34. Jobnamen</i>	
<b>Jobname</b>	<b>Beschreibung</b>
AMQCLMAA	Empfangsprogramm ohne Threads
AMQCRSTA	Responder-Job ohne Threads
AMQRMPPA	Channel-Pool-Job
RUNMQCHI	Kanalinitiator
RUNMQCHL	Channel-Job
RUNMQLSR	Empfangsprogramm mit Threads

## IBM i Kanalzustände unter IBM i

Der Kanalstatus wird in der Anzeige "Work with Channels" (Mit Kanälen arbeiten) eingeblendet.

<i>Tabelle 35. Kanalzustände unter IBM i</i>	
<b>Statusname</b>	<b>Bedeutet</b>
STARTING	Der Kanal ist zur Verhandlung mit dem Ziel-MCA bereit.
BINDING	Einrichten einer Sitzung und erster Datenaustausch
REQUESTING	Requesterkanal initialisiert eine Verbindung
RUNNING	Übertragung wird ausgeführt oder bereit für Übertragung
PAUSED	Warten auf Nachrichtenwiederholungs- intervall
STOPPING	Aufbau von Wiederholung oder Stopp
RETRYING	Warten auf nächsten Versuch
STOPPED	Kanal aufgrund eines Fehlers oder der Ausgabe eines "end-channel"-Befehls gestoppt
INACTIVE	Kanal hat die Verarbeitung normal beendet oder Kanal wurde nie gestartet
*Keine	Kein Status (nur für Serververbindungskanäle)

## ULW Beispiel für Nachrichtenkanalplanung für UNIX, Linux, and Windows

Dieser Abschnitt enthält ein ausführliches Beispiel der Herstellung einer Verbindung zwischen zwei Warteschlangenmanagern, damit sie untereinander Nachrichten austauschen können.

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht. Sie müssen den Kanalinitiator starten, damit der Auslösevorgang funktioniert.

In diesem Beispiel wird SYSTEM.CHANNEL.INITQ als Initialisierungswarteschlange verwendet. Diese Warteschlange ist in IBM MQ bereits definiert. Sie können zwar eine andere Initialisierungswarteschlange verwenden, müssen diese jedoch selbst definieren und beim Start des Kanalinitiators den Namen der Warteschlange angeben.

Das Beispiel zeigt die IBM MQ-Befehle (MQSC), die Sie verwenden können.

In allen Beispielen werden die MQSC-Beispiele so angezeigt, wie sie in einer Befehlsdatei dargestellt und auch in einer Befehlszeile eingegeben würden. Die zwei Vorgehensweisen sehen identisch aus, aber um einen Befehl in die Befehlszeile einzugeben, müssen Sie zuerst `runmqsc` für den Standardwarteschlangenmanager bzw. `runmqsc qmname` eingeben, wobei `qmname` der Name der Name des betreffenden Warteschlangenmanagers ist. Geben Sie dann eine beliebige Anzahl Befehle ein, wie in den Beispielen gezeigt.

Eine alternative Möglichkeit besteht darin, eine Datei mit diesen Befehlen zu erstellen. Jegliche Fehler in den Befehlen können so leicht korrigiert werden. Wenn Sie Ihre Datei "mqsc.in" aufgerufen haben, um sie für den Warteschlangenmanager QMNAME auszuführen, geben Sie Folgendes ein:

```
runmqsc QMNAME < mqsc.in > mqsc.out
```

Durch folgende Eingabe können Sie die Befehle in Ihrer Datei verifizieren, bevor Sie sie ausführen:

```
runmqsc -v QMNAME < mqsc.in > mqsc.out
```

Die Zeilenlänge für Ihre Befehle sollte 72 Zeichen nicht übersteigen, um eine einwandfreie Portierbarkeit zu ermöglichen. Geben Sie ein Verkettungszeichen ein, wenn sich der Befehl über mehr als eine Zeile erstreckt. Verwenden Sie auf Windows-Systemen die Tastenkombination Strg + Z, um die Eingabe in die Befehlszeile zu beenden. Verwenden Sie auf Systemen mit UNIX and Linux die Tastenkombination Strg + D. Alternativ können Sie den Befehl **end** eingeben.

Abbildung 7 auf Seite 190 zeigt das Beispielszenario.

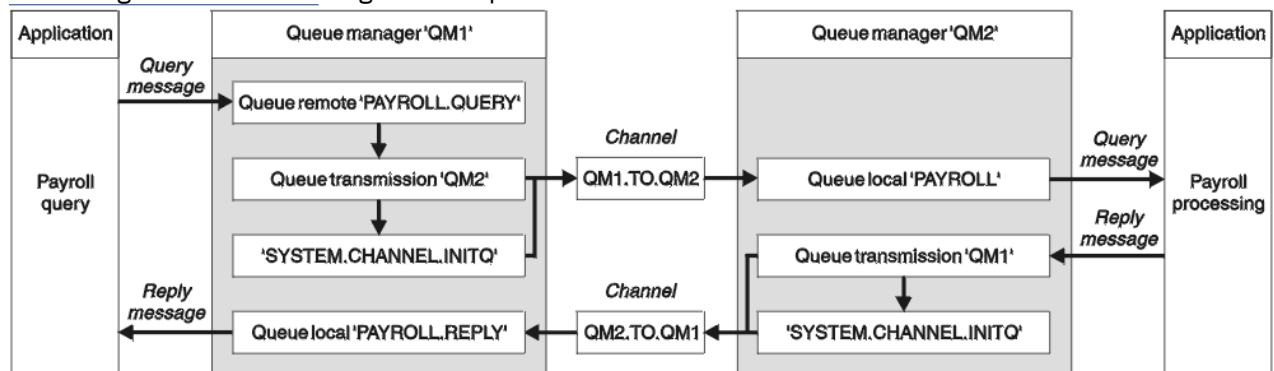


Abbildung 7. Nachrichtenkanalbeispiel für Systeme mit UNIX, Linux, and Windows

Im Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragen an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen. Die Lohnbuchhaltungsabfragen werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

In den Beispielf Definitionen für TCP/IP hat QM1 die Hostadresse 192.0.2.0 und ist am Anschluss 1411 empfangsbereit, und QM2 hat die Hostadresse 192.0.2.1 und ist empfangsbereit am Anschluss 1412. Voraussetzung bei diesem Beispiel ist, dass diese Einstellungen bereits in Ihrem System vorgenommen wurden und die Warteschlangenmanager zur Verfügung stehen.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 7 auf Seite 190 zeigt ein entsprechendes Anordnungsdiagramm.

## Warteschlangenmanager QM1 - Beispiel für UNIX, Linux, and Windows

Über diese Objektdefinitionen können Anwendungen, die mit Warteschlangenmanager QM1 verbunden sind, Anforderungsnachrichten an die Warteschlange PAYROLL auf Warteschlangenmanager QM2 senden und Antworten in der Warteschlange PAYROLL.REPLY auf Warteschlangenmanager QM1 empfangen.

Alle Objektdefinitionen wurden mit den Attributen DESCR und REPLACE bereitgestellt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM1.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM1 aus.

### Definition der fernen Warteschlange

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +  
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQNAME(QM2)
```

**Anmerkung:** Die Definition einer fernen Warteschlange ist keine physische Warteschlange, sondern ein Mittel zum Übertragen von Nachrichten an die Übertragungswarteschlange QM2, damit sie an Warteschlangenmanager QM2 gesendet werden können.

### Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM1.TO.QM2.PROCESS)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die

Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

### Senderkanaldefinition

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +  
CONNNAME('192.0.2.1(1412)')
```

### Empfängerkanaldefinition

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM2')
```

### Definition einer Empfangswarteschlange für Antworten

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to query messages sent to QM2')
```

Die Empfangswarteschlange für Antworten ist als PUT(ENABLED) definiert. Dies stellt sicher, dass Antwortnachrichten in die Warteschlange gestellt werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereiht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

## U1W

## Warteschlangenmanager QM2 - Beispiel für UNIX, Linux, and Windows

### OWS

Die folgenden Objektdefinitionen ermöglichen Anwendungen mit Verbindung zum Warteschlangenmanager QM2, Anforderungsnachrichten aus der lokalen Warteschlange PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in die Warteschlange mit dem PAYROLL.REPLY auf Warteschlangenmanager QM1 zu stellen.

Die Bereitstellung der Definition einer fernen Warteschlange zur Rückgabe der Antworten an QM1 entfällt hierbei. Der Nachrichtendeskriptor der aus der lokalen Warteschlange PAYROLL abgerufenen Nachricht enthält sowohl den Namen der Empfangswarteschlange für Antworten als auch den Namen des Managers der Empfangswarteschlange für Antworten. Solange also QM2 den Namen des Managers der Empfangswarteschlange für Antworten in den einer Übertragungswarteschlange auf Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Alle Objektdefinitionen enthalten die Attribute DESCR und REPLACE, dies stellt eine Mindestvoraussetzung für ein Funktionieren des Beispiels dar. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM2.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM2 aus.

### Definition der lokalen Warteschlange

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Local queue for QM1 payroll details')
```



Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

### Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM2.TO.QM1.PROCESS)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

### Senderkanaldefinition

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('192.0.2.0(1411)')
```

### Empfängerkanaldefinition

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM1')
```

**ULW**

## Beispiel für UNIX, Linux, and Windows ausführen

Informationen zum Starten des Kanalinitiators und des Empfangsprogramms sowie Vorschläge zum Erweitern dieses Szenarios.

Sobald diese Definitionen erstellt wurden, müssen Sie folgende Schritte ausführen:

- Starten Sie den Kanalinitiator für die einzelnen Warteschlangenmanager.
- Starten Sie das Empfangsprogramm für die einzelnen Warteschlangenmanager.

Informationen zum Starten des Kanalinitiators und des Empfangsprogramms finden Sie in den Abschnitten [Kommunikation für Windows konfigurieren](#) und [Kommunikation auf Systemen mit UNIX and Linux konfigurieren](#).

### Dieses Beispiel erweitern

Dieses einfache Beispiel lässt sich auf folgende Weise erweitern:

- Die Verwendung der LU 6.2-Kommunikation mit Verbindungen zwischen CICS-Systemen und Transaktionsverarbeitung.
- Hinzufügen weiterer Warteschlangen, Prozesse und Kanaldefinitionen, damit andere Anwendungen Nachrichten zwischen den zwei Warteschlangenmanagern übertragen können.
- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverschlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden Sie Warteschlangenmanager-Aliasnamen und Aliasnamen für die Warteschlange für Antwortnachrichten, um weitere Kenntnisse darüber zu erlangen, wie diese in der Organisation Ihres Warteschlangenmanagernetzes verwendet werden können.

Dieser Abschnitt enthält ein ausführliches Beispiel für die Herstellung einer Verbindung zwischen zwei IBM i-Warteschlangenmanagern, damit sie untereinander Nachrichten austauschen können.

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht.

In diesem Beispiel wird SYSTEM.CHANNEL.INITQ als Initialisierungswarteschlange verwendet. Diese Warteschlange ist in IBM MQ bereits definiert. Sie können eine andere Initialisierungswarteschlange verwenden, müssen diese jedoch selbst definieren, mit dem Befehl STRMQMCHLI eine neue Instanz des Kanalinitiators starten und ihr den Namen Ihrer Initialisierungswarteschlange bereitstellen. Weitere Informationen zum Auslösen von Kanälen finden Sie im Abschnitt [Kanäle auslösen](#).

### Erläuterung der Beispiele für IBM i

In diesem Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragenachrichten an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen.

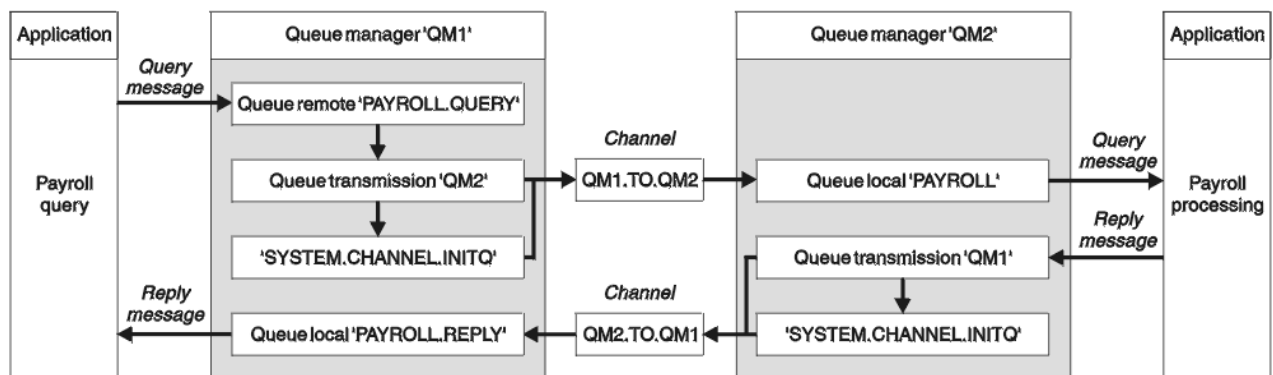


Abbildung 8. Nachrichtenkanalbeispiel für IBM MQ for IBM i

Die Lohnbuchhaltungsabfragenachrichten werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

Bei beiden Warteschlangenmanagern wird vorausgesetzt, dass sie unter IBM i ausgeführt werden. In den Beispielfinitionen hat QM1 die Hostadresse 192.0.2.0 und ist an Port 1411 empfangsbereit. QM2 hat die Hostadresse 192.0.2.1 und ist an Port 1412 empfangsbereit. In diesem Beispiel wird davon ausgegangen, dass diese Warteschlangenmanager bereits auf dem IBM i-System definiert wurden und die Warteschlangenmanager zur Verfügung stehen.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 8 auf Seite 194 zeigt ein entsprechendes Anordnungsdiagramm.

## Warteschlangenmanager QM1 - Beispiel für IBM i

Die folgenden Objektdefinitionen ermöglichen mit dem Warteschlangenmanager QM1 verbundenen Anwendungen, Anforderungsnachrichten an eine Warteschlange mit dem Namen PAYROLL auf QM2 zu senden, und Antworten in einer Warteschlange mit dem Namen PAYROLL.REPLY auf QM1 zu empfangen.

Sämtliche Objektdefinitionen werden in den TEXT-Attributen genannt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM1.

Führen Sie auf Warteschlangenmanager QM1 folgende Befehle aus:

### Definition der fernen Warteschlange

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	'PAYROLL.QUERY'
QTYPE	*RMT
TEXT	'Ferne Warteschlange für QM2'
PUTENBL	*YES
TMQNAME	'QM2' (Standard: Name des fernen Warteschlangenmanagers)
RMTQNAME	'PAYROLL'
RMTMQNAME	'QM2'

**Anmerkung:** Die Definition einer fernen Warteschlange ist keine physische Warteschlange, sondern ein Mittel zum Übertragen von Nachrichten an die Übertragungswarteschlange QM2, damit sie an Warteschlangenmanager QM2 gesendet werden können.

### Definition der Übertragungswarteschlange

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	QM2
QTYPE	*LCL
TEXT	'Übertragungswarteschlange zu QM2'
USAGE	*TMQ
PUTENBL	*YES

GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM1.TO.QM2

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

### Senderkanaldefinition

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM1.TO.QM2
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Senderkanal zu QM2'
TMQNAME	QM2
CONNNAME	'192.0.2.1(1412)'

### Empfängerkanaldefinition

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM2.TO.QM1
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Empfängerkanal von QM2'

### Definition einer Empfangswarteschlange für Antworten

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	PAYROLL.REPLY
QTYPE	*LCL
TEXT	'Antwortwarteschlange für Antworten zu Abfragenachrichten, die an QM2 gesendet wurden'
PUTENBL	*YES
GETENBL	*YES

Die Empfangswarteschlange für Antworten ist als PUT(ENABLED) definiert. Diese Definition stellt sicher, dass Antwortnachrichten in die Warteschlange eingereicht werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereicht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

Die folgenden Objektdefinitionen ermöglichen Anwendungen mit Verbindung zum Warteschlangenmanager QM2, Anforderungsnachrichten aus der lokalen Warteschlange PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in die Warteschlange mit dem PAYROLL.REPLY auf Warteschlangenmanager QM1 zu stellen.

Die Bereitstellung der Definition einer fernen Warteschlange zur Rückgabe der Antworten an QM1 entfällt hierbei. Der Nachrichtendeskriptor der aus der lokalen Warteschlange PAYROLL abgerufenen Nachricht enthält sowohl den Namen der Empfangswarteschlange für Antworten als auch den Namen des Managers der Empfangswarteschlange für Antworten. Solange also QM2 den Namen des Managers der Empfangswarteschlange für Antworten in den einer Übertragungswarteschlange auf Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Alle Objektdefinitionen werden mit dem Attribut TEXT genannt und sind die Mindestvoraussetzung für die erfolgreiche Ausführung des Beispiels. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM2.

Führen Sie auf Warteschlangenmanager QM2 folgende Befehle aus:

#### **Definition für lokale Warteschlange**

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	PAYROLL
QTYPE	*LCL
TEXT	'Lokale Warteschlange für Lohnbuchhaltungsdetails von QM1'
PUTENBL	*YES
GETENBL	*YES

Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

#### **Definition der Übertragungswarteschlange**

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	QM1
QTYPE	*LCL
TEXT	'Übertragungswarteschlange zu QM1'
USAGE	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM2.TO.QM1

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal.

#### **Senderkanaldefinition**

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM2.TO.QM1
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Senderkanal zu QM1'
TMQNAME	QM1
CONNNAME	'192.0.2.0(1411)'

### Empfängerkanaldefinition

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM1.TO.QM2
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Empfängerkanal von QM1'

## IBM i Beispiel für IBM i ausführen

Nachdem Sie die erforderlichen Objekte erstellt haben, müssen Sie die Kanalinitiatoren und Empfangsprogramme für die beiden Warteschlangenmanager starten.

Die Anwendungen können sich daraufhin gegenseitig Nachrichten zusenden. Das Starten der Kanäle wird durch die erste Nachricht ausgelöst, die in jeder Übertragungswarteschlange eintrifft, daher muss der Befehl STRMQMCHL nicht ausgegeben werden.

Ausführliche Informationen zum Starten eines Kanalinitiators und eines Empfangsprogramms finden Sie im Abschnitt [Kanäle unter IBM i überwachen und steuern](#).

## IBM i Beispiel für IBM i erweitern

Das Beispiel kann auf zahlreiche Art erweitert werden.

Dieses Beispiel können Sie wie folgt erweitern:

- Fügen Sie weitere Warteschlangen- und Kanaldefinitionen hinzu, damit andere Anwendungen Nachrichten zwischen den beiden Warteschlangenmanagern senden können.
- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverschlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden Sie Warteschlangenmanager-Aliasnamen und Aliasnamen für die Warteschlange für Antwortnachrichten, um weitere Kenntnisse darüber zu erlangen, wie diese Objekte in der Organisation Ihres Warteschlangenmanagernetzes verwendet werden können.

Eine Version dieses Beispiels, das MQSC-Befehle verwendet, finden Sie unter [„Beispiel für Nachrichtenkanalplanung für z/OS“](#) auf Seite 198.

## z/OS Beispiel für Nachrichtenkanalplanung für z/OS

Dieser Abschnitt enthält ein ausführliches Beispiel für die Herstellung einer Verbindung zwischen z/OS- oder MVS-Warteschlangenmanagern, damit sie untereinander Nachrichten austauschen können.

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen und LU 6.2-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht.

## z/OS Erläuterung der Beispiele für z/OS

In diesem Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragenachrichten an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen.

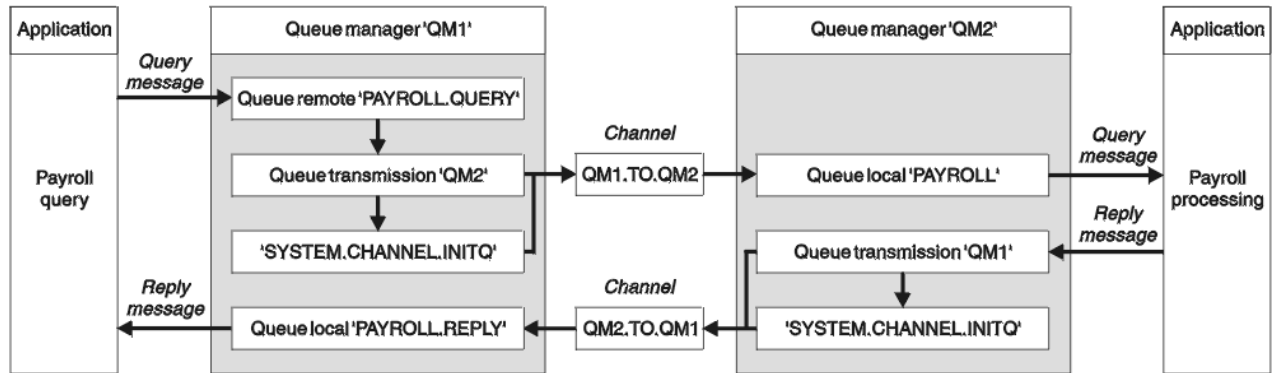


Abbildung 9. Erstes Beispiel für IBM MQ for z/OS

Die Lohnbuchhaltungsabfragenachrichten werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

Bei beiden Warteschlangenmanagern wird vorausgesetzt, dass sie unter z/OS ausgeführt werden. In den Beispieldefinitionen für TCP/IP hat QM1 die Hostadresse 192.0.2.0 und ist am Anschluss 1411 empfangsbereit, und QM2 hat die Hostadresse 192.0.2.1 und ist empfangsbereit am Anschluss 1412. In den Definitionen für LU 6.2 ist QM1 auf einer symbolischen logischen Einheit mit dem Namen LUNAME1 empfangsbereit und QM2 ist auf einer symbolischen logischen Einheit mit dem Namen LUNAME2 empfangsbereit. In diesem Beispiel wird davon ausgegangen, dass die Namen der logischen Einheiten (LUNAMES) bereits auf dem z/OS-System definiert wurden und die Warteschlangenmanager zur Verfügung stehen. Informationen zu ihrer Definition finden Sie in „MQ-Beispielkonfiguration für z/OS“ auf Seite 63.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL

- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

In diesem Beispiel wird davon ausgegangen, dass alle SYSTEM.COMMAND.\* und SYSTEM.CHANNEL.\* Warteschlangen, die für die Ausführung von DQM erforderlich sind, wurden wie in den bereitgestellten Beispieldefinitionen **CSQ4INSG** und **CSQ4INSX** definiert.

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 9 auf Seite 199 zeigt ein entsprechendes Anordnungsdiagramm.

## Warteschlangenmanager QM1 - Beispiel für z/OS

Die folgenden Objektdefinitionen ermöglichen mit dem Warteschlangenmanager QM1 verbundenen Anwendungen, Anforderungsnachrichten an eine Warteschlange mit dem Namen PAYROLL auf QM2 zu senden. Es ermöglicht Anwendungen ebenfalls, Antworten in einer Warteschlange auf QM1 mit dem Namen PAYROLL.REPLY zu empfangen.

Alle Objektdefinitionen wurden mit den Attributen DESCR und REPLACE bereitgestellt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM1.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM1 aus.

### Definition der fernen Warteschlange

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

**Anmerkung:** Die Definition einer fernen Warteschlange ist keine physische Warteschlange, sondern ein Mittel zum Übertragen von Nachrichten an die Übertragungswarteschlange QM2, damit sie an Warteschlangenmanager QM2 gesendet werden können.

### Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM1.TO.QM2) INITQ(SYSTEM.CHANNEL.INITQ)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal. Der Kanalinitiator kann nur Auslösenachrichten von der Warteschlange SYSTEM.CHANNEL.INITQ abrufen, verwenden Sie also keine andere Warteschlange als Initialisierungswarteschlange.

### Senderkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('192.0.2.1(1412)')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(LU62) +
```



```
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('LUNAME2')
```

## Empfängerkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM2')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM2')
```

## Definition einer Empfangswarteschlange für Antworten

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QM2')
```

Die Empfangswarteschlange für Antworten ist definiert als PUT(ENABLED), wodurch sichergestellt wird, dass Antwortnachrichten in die Warteschlange gestellt werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereiht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

## Warteschlangenmanager QM2 - Beispiel für z/OS

Die folgenden Objektdefinitionen ermöglichen Anwendungen mit Verbindung zum Warteschlangenmanager QM2, Anforderungsnachrichten aus der lokalen Warteschlange PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in die Warteschlange mit PAYROLL.REPLY auf Warteschlangenmanager QM1 zu stellen.

Die Bereitstellung der Definition einer fernen Warteschlange zur Rückgabe der Antworten an QM1 entfällt hierbei. Der Nachrichtendeskriptor der aus der lokalen Warteschlange PAYROLL abgerufenen Nachricht enthält sowohl den Namen der Empfangswarteschlange für Antworten als auch den Namen des Managers der Empfangswarteschlange für Antworten. Solange also QM2 den Namen des Managers der Empfangswarteschlange für Antworten in den einer Übertragungswarteschlange auf Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Alle Objektdefinitionen enthalten die Attribute DESCR und REPLACE, dies stellt eine Mindestvoraussetzung für ein Funktionieren des Beispiels dar. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM2.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM2 aus.

## Definition für lokale Warteschlange

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Local queue for QM1 payroll details')
```

Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

## Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
TRIGDATA(QM2.TO.QM1) INITQ(SYSTEM.CHANNEL.INITQ)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal. Der Kanalinitiator kann nur Auslösenachrichten aus SYSTEM.CHANNEL.INITQ abrufen, verwenden Sie also keine andere Warteschlange als Initialisierungswarteschlange.

### Senderkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNAME('192.0.2.0(1411)')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(LU62) +  
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNAME('LUNAME1')
```

### Empfängerkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM1')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(LU62) +  
REPLACE DESCR('Receiver channel from QM1')
```

## Beispiel für z/OS ausführen

Wenn Sie die erforderlichen Objekte erstellt haben, müssen Sie die Kanalinitiatoren und die Empfangsprogramme für beide Warteschlangenmanager starten.

Die Anwendungen können sich daraufhin gegenseitig Nachrichten zusenden. Da der Start der Kanäle durch die Ankunft der ersten Nachricht in den einzelnen Übertragungswarteschlangen ausgelöst wird, müssen Sie den Befehl START CHANNEL MQSC nicht ausgeben.

Ausführliche Informationen zum Starten eines Kanalinitiators und eines Empfangsprogramms finden Sie in den Abschnitten [Kanalinitiator starten](#) bzw. [Kanalempfangsprogramm starten](#).

## Beispiel für z/OS erweitern

Das Beispiel kann auf zahlreiche Art erweitert werden.

Das Beispiel kann durch folgende Schritte erweitert werden:

- Hinzufügen weiterer Warteschlangen und Kanaldefinitionen, damit andere Anwendungen Nachrichten zwischen den zwei Warteschlangenmanagern übertragen können.

- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverschlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden von Aliassen für Warteschlangenmanager und Aliassen für Empfangswarteschlangen für Antworten, um mehr darüber zu erfahren, wie diese Aliase in die Organisation Ihres Netzwerks aus Warteschlangenmanagern eingebunden werden können.

## **z/OS** Nachrichtenkanalplanungsbeispiel für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM3 verwendet, Nachrichten in eine Warteschlange einer Gruppe mit gemeinsamer Warteschlange, die die Warteschlangenmitglieder QM4 und QM5 enthält, einreihen kann.

Bevor Sie dieses Beispiel ausprobieren, sollten Sie unbedingt mit dem Beispiel im Abschnitt [„Beispiel für Nachrichtenkanalplanung für z/OS“](#) auf Seite 198 vertraut sein.

## **z/OS** Erläuterung des Beispiels der Gruppe mit gemeinsamer Warteschlange für z/OS

Dieses Beispiel zeigt die IBM MQ-Befehle (MQSC), die Sie unter IBM MQ for z/OS in Zusammenhang mit Gruppen mit gemeinsamer Warteschlange für die verteilte Steuerung von Warteschlangen verwenden können.

Dieses Beispiel erweitert das Szenario "Lohnbuchhaltungsabfrage" des Beispiels in [„Beispiel für Nachrichtenkanalplanung für z/OS“](#) auf Seite 198 und zeigt, wie eine höhere Verfügbarkeit der Abfrageverarbeitung durch Hinzufügen weiterer bereitstellender Anwendungen hinzugefügt wird, um eine gemeinsam genutzte Warteschlange bereitzustellen.

Die Anwendung zur Lohnbuchhaltungsabfrage ist nun mit dem Warteschlangenmanager QM3 verbunden und reiht eine Abfrage in die in QM3 definierte ferne Warteschlange 'PAYROLL QUERY' ein. Diese definierte ferne Warteschlange wird in die gemeinsam genutzte Warteschlange PAYROLL aufgelöst, die sich auf den Warteschlangenmanagern in der Gruppe QSG1 mit gemeinsamer Warteschlange befindet. Die Anwendung zur Lohnbuchhaltungsverarbeitung hat nun zwei aktive Instanzen - eine ist mit QM4 verbunden, die andere mit QM5.

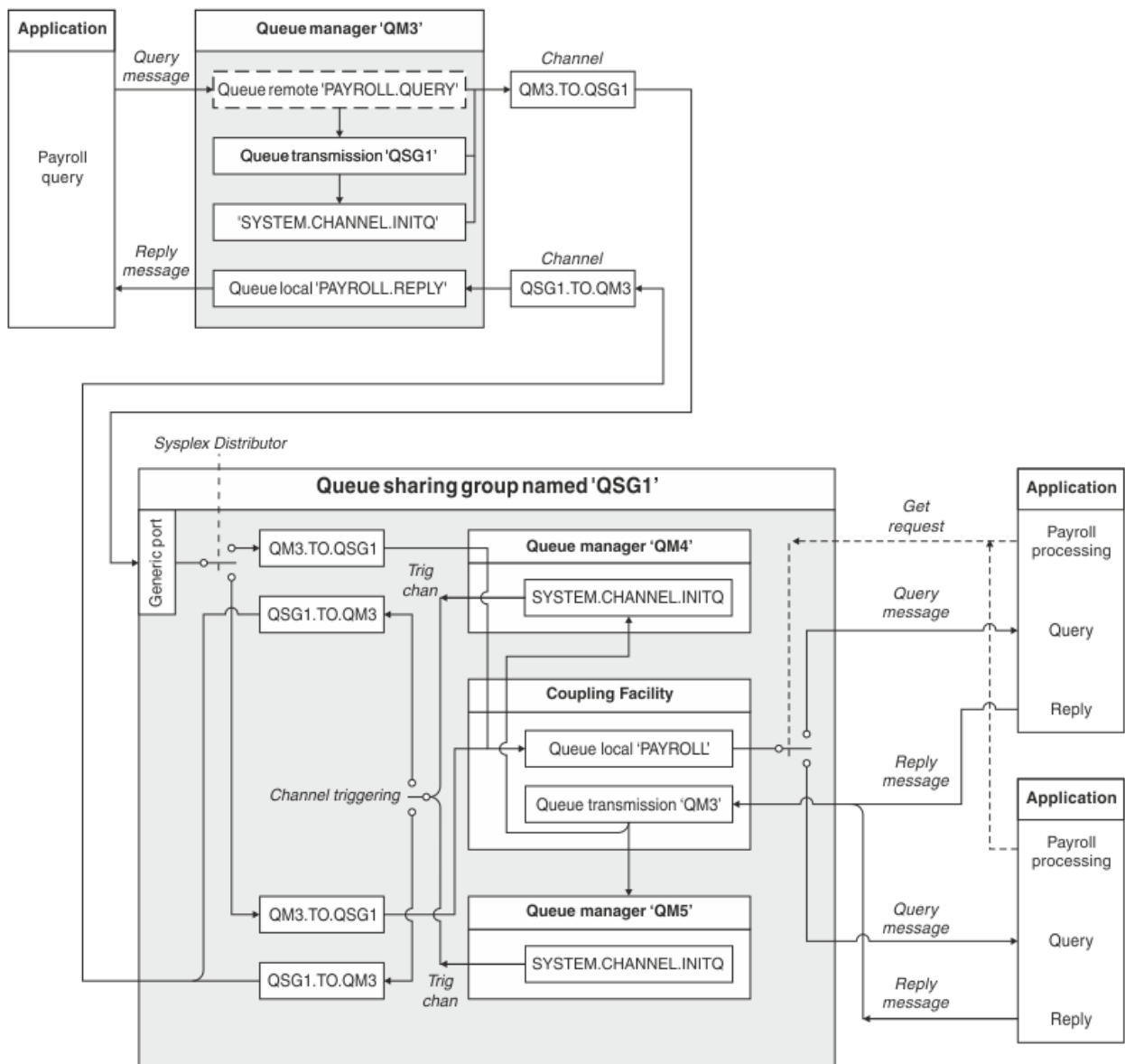


Abbildung 10. Nachrichtenkanalplanungsbeispiel für IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange

Bei allen drei Warteschlangenmanagern wird davon ausgegangen, dass sie unter z/OS aktiv sind. In den Beispielformen für TCP/IP hat QM4 die virtuelle IP-Adresse MVSIP01, QM5 die virtuelle IP-Adresse MVSIP02. Beide Warteschlangenmanager sind an Port 1414 empfangsbereit. Sysplex Distributor stellt für diese Gruppe die generische Adresse QSG1.MVSIP bereit. QM3 hat die Hostadresse 192.0.2.0 und ist an Port 1411 empfangsbereit.

In den Beispielformen für LU6.2 ist QM3 an dem symbolischen LU-Namen LUNAME1 empfangsbereit. Der Name der generischen Ressource, die für VTAM für die LU-Namen definiert ist, an denen QM4 und QM5 empfangsbereit sind, lautet LUQSG1. In diesem Beispiel wird davon ausgegangen, dass die Namen der logischen Einheiten (LUNAMES) bereits auf dem z/OS-System definiert wurden und verwendet werden können. Informationen zu ihrer Definition finden Sie unter „Selbstdefinition im Netz mithilfe generischer Ressourcen“ auf Seite 72.

In diesem Beispiel ist QSG1 der Name einer Gruppe mit gemeinsamer Warteschlange und die Namen der Warteschlangenmanager QM4 und QM5 sind Namen von Mitgliedern dieser Gruppe.

## Definitionen der Gruppe mit gemeinsamer Warteschlange

Durch Erstellen der folgenden Objektdefinitionen für ein Mitglied der Gruppe mit gemeinsamer Warteschlange stehen sie auch für alle anderen Mitglieder zur Verfügung.

Die Warteschlangenmanager QM4 und QM5 sind Mitglieder der Gruppe mit gemeinsamer Warteschlange. Die für QM4 erstellten Definitionen sind auch für QM5 verfügbar.

Es wird vorausgesetzt, dass die Listenstruktur der Coupling-Facility den Namen 'APPLICATION1' hat. Lautet der Name nicht 'APPLICATION1', verwenden Sie in dem Beispiel Ihren eigenen Namen der Listenstruktur der Coupling-Facility.

### Gemeinsam genutzte Objekte

Die Definitionen für gemeinsam genutzte Objekte sind in Db2 gespeichert, ihre zugehörigen Nachrichten sind in der Coupling-Facility gespeichert.

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) REPLACE PUT(ENABLED) GET(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Shared queue for payroll details')
```

```
DEFINE QLOCAL(QM3) QSGDISP(SHARED) REPLACE USAGE(XMITQ) PUT(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Transmission queue to QM3') TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QSG1.TO.QM3) GET(ENABLED) INITQ(SYSTEM.CHANNEL.INITQ)
```

### Gruppenobjekte

Die Gruppenobjektdefinitionen sind in Db2 gespeichert. Jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange erstellt eine lokale Kopie des definierten Objekts.

Senderkanaldefinition für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNNAME('192.0.2.0(1411)')
```

Senderkanaldefinition für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNNAME('LUNAME1')
```

Empfängerkanaldefinition für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

Empfängerkanaldefinition für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

### Zugehörige Verweise

„Disposition (QSGDISP)“ auf Seite 130

Dieses Attribut gibt die Disposition des Kanals in einer Gruppe mit gemeinsamer Warteschlange an. Dieser Parameter ist nur unter z/OS gültig.

## Warteschlangenmanager QM3 - Beispiel für z/OS

QM3 ist kein Mitglied der Gruppe mit gemeinsamer Warteschlange. Die folgenden Objektdefinitionen ermöglichen ihm, Nachrichten in eine Warteschlange in der Gruppe mit gemeinsamer Warteschlange einzureihen.

Der CONNAME für diesen Kanal ist die generische Adresse der Gruppe mit gemeinsamer Warteschlange, die je nach Transporttyp variiert.

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('QSG1.MVSIP(1414)')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('LUQSG1') TPNAME('MQSERIES') MODENAME('#INTER')
```

### Sonstige Definitionen

Diese Definitionen sind für denselben Zweck wie die Definitionen im ersten Beispiel erforderlich.

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QSG1') REPLACE +
PUT(ENABLED) XMITQ(QSG1) RNAME(APPL) RQMNAME(QSG1)
```

```
DEFINE QLOCAL(QSG1) DESCR('Transmission queue to QSG1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM3.TO.QSG1) INITQ(SYSTEM.CHANNEL.INITQ)
```

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QSG1')
```

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QSG1')
```

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QSG1')
```

## Beispiel mit Gruppe mit gemeinsamer Warteschlange für z/OS ausführen

Nachdem Sie die erforderlichen Objekte erstellt haben, müssen Sie die Kanalinitiatoren für alle drei Warteschlangenmanager starten. Gestartet werden müssen auch die Empfangsprogramme für beide Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange.

Bei einer TCP/IP-Verbindung muss für jedes Mitglied der Gruppe ein Gruppenempfangsprogramm gestartet werden, das an Port 1414 empfangsbereit ist.

```
STA LSTR PORT(1414) IPADDR(MVSIP01) INDISP(GROUP)
```

Der vorherige Eintrag startet beispielsweise das Empfangsprogramm in QM4.

Bei einer LU 6.2-Verbindung muss für jedes Mitglied der Gruppe ein Gruppenempfangsprogramm gestartet werden, das an einem symbolischen LU-Namen empfangsbereit ist. Dieser LU-Name muss der generischen Ressource LUQSG1 entsprechen.

- Starten Sie das Empfangsprogramm in QM3.

```
STA LSTR PORT(1411)
```

## Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek

Sie können ein Alias definieren, das auf eine MQ-Bibliothek in Ihrer Jobsteuersprache verweist, statt den Namen der MQ-Bibliothek direkt anzugeben. Wenn sich später der Name der MQ-Bibliothek ändert, brauchen Sie lediglich das Alias zu löschen und erneut zu definieren.

### Beispiel

Im folgenden Beispiel wird ein Alias MQM.SCSQANLE definiert, das auf die MQ-Bibliothek MQM.V600.SCSQANLE verweist:

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SCSQANLE)
DEFINE ALIAS (NAME(MQM.SCSQANLE) RELATE(MQM.V600.SCSQANLE))
/*
```

Anschließend können Sie in Ihrer Jobsteuersprache über das Alias MQM.SCSQANLE auf die Bibliothek MQM.V600.SCSQANLE verweisen.

**Anmerkung:** Der Bibliotheksname und der Aliasname müssen sich in demselben Katalog befinden; verwenden Sie deshalb für beide dasselbe übergeordnete Qualifikationsmerkmal (im vorliegenden Beispiel MQM).

z/OS

V 9.0.1

## Element mqzOSConnectService

Der MQ Service Provider wird als Liberty-Standardfeature bereitgestellt und daher mit server.xml konfiguriert. Jeder uni- oder bidirektionale Service wird in einem Element mqzOSConnectService definiert. Dieses Element und alle seine Attribute gelten für z/OS Connect V1 und z/OS Connect EE.

**Wichtig:** Ein Element mqzOSConnectService kann erst verwendet werden, wenn von einem Element zOSConnectService darauf verwiesen wird.

Im Folgenden ist ein Beispiel für ein Element mqzOSConnectService mit Angabe einiger Attribute dargestellt.

```
<mqzOSConnectService id="twoWay "
    connectionFactory="jms/cf1"
    destination="jms/requestQueue"
    replyDestination="jms/replyQueue"
    expiry="-1"
    waitInterval="10000"
    replySelection="msgIDToCorrelID"
    selector=""
    persistence="false"/>
```



**Achtung:** Depending on how the MQ Service Provider has been installed, the mqzOSConnectService element might be prefixed with a string followed by an underscore, for example `usr_mqzOSConnectService`.

Dies wird im Abschnitt [Installing the MQ Service Provider into WLP \(MQ Service Provider in WLP installieren\)](#) für z/OS Connect V1 und im Abschnitt [Installing the MQ Service Provider into IBM z/OS Connect EE \(MQ Service Provider in IBM z/OS Connect EE installieren\)](#) für z/OS Connect EE beschrieben.

Das folgende Beispiel zeigt das Format, in dem MQ Service Provider im WLP-Kernel installiert wurde (wie in Option 1 unter [MQ Service Provider in WLP installieren](#) beschrieben).

Tabelle 36. Attribute des Elements `mqzOSConnectService`

Attributname	Typ	Standardwert	Beschreibung
ID	Zeichenfolge		„ID“ auf Seite 208
connectionFactory	Ein JNDI-Name (Zeichenfolge).		„connectionFactory“ auf Seite 208
Ziel	Ein JNDI-Name (Zeichenfolge).		„Ziel“ auf Seite 208
replyDestination	Ein JNDI-Name (Zeichenfolge).		„replyDestination“ auf Seite 209
expiry	Integer	-1	„expiry“ auf Seite 209
waitInterval	Integer		„waitInterval“ auf Seite 209
replySelection	Zeichenfolge	msgIDToCorrelID	„replySelection“ auf Seite 210
Selektor	Zeichenfolge		„Selektor“ auf Seite 210
persistence	boolean	false	„persistence“ auf Seite 210
mqmdFormat	Zeichenfolge		„mqmdFormat“ auf Seite 211
userName	Zeichenfolge		„userName“ auf Seite 211
Kennwort	Zeichenfolge		„Kennwort“ auf Seite 211
useCallerPrincipal	boolean	false	„useCallerPrincipal“ auf Seite 211
receiveTextCCSID	Integer	273	„receiveTextCCSID“ auf Seite 211

## ID

**id** ist ein erforderliches Attribut, das für alle Elemente in `server.xml` eindeutig sein muss. **id** wird vom Element `zosConnectService` verwendet, um auf eine Ziel-Service-Provider-Instanz zu verweisen.

## connectionFactory

**connectionFactory** gibt den JNDI-Namen einer IBM MQ-Messaging-Provider-Verbindungsfactory an. Der MQ Service Provider verwendet die Verbindungsfactory, um eine Verbindung zu IBM MQ herzustellen.

**connectionFactory** ist ein erforderliches Attribut. Weitere Informationen zu Verbindungsfactorys finden Sie unter [JMS-Verbindungsfactory](#).

Sie sollten **transportType**="BINDINGS" für die Verbindungsfactory angeben.

## Ziel

**destination** gibt den JNDI-Namen eines IBM MQ-Messaging-Provider-Ziels an.

**destination** ist ein erforderliches Attribut.

Weitere Konfigurationsinformationen:

- Konfiguration einer Warteschlange in WLP: siehe Abschnitt [JMS Queue](#) (JMS-Warteschlange).



- Konfiguration eines Topics in WLP: siehe Abschnitt [JMS Topic](#).

Bei einem unidirektionalen Service wird **destination** als Ziel für HTTP POST-, HTTP GET- und HTTP DELETE-Anforderungen verwendet.

Beachten Sie, dass Warteschlangenziele für alle drei Anforderungstypen unterstützt werden, während Topic-Ziele nur bei HTTP POST-Anforderungen unterstützt werden.

Bei einem bidirektionalen Service muss **destination** ein Warteschlangenziel sein, das die vom Back-End-Service verwendete Anforderungswarteschlange darstellt.

Bidirektionale Services unterstützen nur HTTP POST-Anforderungen.

## replyDestination

**replyDestination** gibt den JNDI-Namen einer IBM MQ-Messaging-Provider-Warteschlange an.

**replyDestination** ist ein optionales Attribut.

Weitere Informationen zum Konfigurieren einer Warteschlange in WLP finden Sie im Abschnitt [JMS Queue](#) (JMS-Warteschlange).

Wenn **replyDestination** nicht angegeben ist, handelt es sich um einen unidirektionalen Service. Ist **replyDestination** dagegen angegeben, ist der Service ein bidirektionaler Service.

Diese Warteschlange ist das Antwortziel, an das der Back-End-Service Antwortnachrichten sendet.

## expiry

**expiry** gibt an, wie lange Nachrichten, die vom MQ Service Provider gesendet wurden, gültig sind (in Tausendstelsekunden ab dem Zeitpunkt, an dem sie gesendet wurden). Die Nachricht kann gelöscht werden, wenn sie nicht aus der Zielwarteschlange entfernt wird, bevor dieser Zeitraum verstrichen ist.

**expiry** ist ein optionales Attribut und entspricht der Einstellung des MQMD-Felds [Expiry](#).

Bei Angabe negativer Werte laufen die Nachrichten niemals ab. Der Standardwert für **expiry** lautet -1.

REST-Clients können **expiry** durch Angabe eines HTTP-Headers `ibm-mq-md-expiry` mit gültiger 64-Bit-Ganzzahl außer Kraft setzen.

## waitInterval

Bei HTTP DELETE-Anforderungen an unidirektionale Services gibt **waitInterval** die Anzahl der Millisekunden an, die der Service auf eine übereinstimmende Nachricht in der Warteschlange wartet, die über das Attribut [destination](#) angegeben wurde.

Bei HTTP POST-Anforderungen an bidirektionale Services gibt **waitInterval** die Anzahl der Millisekunden an, die der Service auf eine übereinstimmende Nachricht in der Warteschlange wartet, die über das Attribut [replydestination](#) angegeben wurde.

**waitInterval** ist ein optionales Attribut für unidirektionale Services, ein erforderliches Attribut für bidirektionale Services und entspricht der Einstellung des MQMD-Felds [WaitInterval](#).

**waitInterval** wird bei HTTP GET-Anforderungen nicht unterstützt.

Für **waitInterval** gilt Folgendes:

- Bei einem Wert von null wartet der Service nicht.
  - Bei bidirektionalen Services wird der Wert null für **waitInterval** nicht unterstützt.
- Bei einem negativen Wert wartet der Service unbegrenzt, bis eine Nachricht verfügbar ist.

REST-Clients können diesen Wert durch Angabe eines HTTP-Headers `ibm-mq-gmo-waitInterval` mit gültiger 64-Bit-Ganzzahl außer Kraft setzen.

**Anmerkung:** Wenn ein großer oder negativer Wert für **waitInterval** angegeben wird, wird es voraussichtlich zu Zeitlimitüberschreitungen bei den Transaktionen und asynchronen Serviceanforderungen

kommen. Wenn eines oder beide diese Ereignisse eintreten, erhöhen Sie das Zeitlimit und/oder verringern Sie das Warteintervall.

## replySelection

`replySelection` beschreibt den Mechanismus zum Abgleich von Antwortnachrichten mit Anforderungsnachrichten.

`replySelection` ist optional und wird nur bei bidirektionalen Services verwendet. Wenn `replySelection` bei einem unidirektionalen Service verwendet wird, wird die Angabe ignoriert.

Folgende Werte sind möglich:

### msgIDToCorrelID

Es wird angenommen, dass bei der Generierung von Antwortnachrichten für die Korrelations-ID der Wert der Nachrichten-ID der Anforderungsnachricht festgelegt wird. Der Service erstellt anhand dieser Informationen einen geeigneten Nachrichtenselektor. Dies ist der Standardwert.

### none

Es wird kein Mechanismus zum Korrelieren von Antwortnachrichten mit Anforderungsnachrichten verwendet. Der Service ruft die erste verfügbare Nachricht in der Antwortwarteschlange ab.

### correlIDToCorrelID

Es wird angenommen, dass bei der Generierung von Antwortnachrichten für die Korrelations-ID der Wert der Korrelations-ID der Anforderungsnachricht festgelegt wird. Der Service erstellt anhand dieser Informationen einen geeigneten Nachrichtenselektor. Wenn für die Anforderungsnachricht keine Korrelations-ID angegeben ist (siehe Abschnitt „[ibm-mq-md-correlID](#)“ auf Seite 212), generiert der Service eine zufällige Korrelations-ID für die Anforderungsnachricht.

## Selektor

`selector` muss ein gültiger JMS-Nachrichtenselektor gemäß der Beschreibung in der JMS-Spezifikation sein.

`selector` wird nur bei unidirektionalen Services verwendet und ist optional. Wenn `selector` bei bidirektionalen Services verwendet wird, wird die Angabe ignoriert. Weitere Informationen zu Selektoren finden Sie im Abschnitt [Nachrichtenselektoren in JMS](#).

`selector` wird bei HTTP GET- und HTTP DELETE-Anforderungen zur Auswahl der zurückgegebenen Nachricht verwendet. Wenn die Header „[ibm-mq-md-msgID](#)“ auf Seite 212 oder „[ibm-mq-md-correlID](#)“ auf Seite 212 angegeben sind, wird `selector` ignoriert.

Manche für 'selector' verwendete Zeichen müssen codiert werden, um in die Datei `server.xml` integriert werden zu können. Hierfür können Sie folgende Standardmechanismen verwenden:

```
" becomes &quot;  
' becomes &apos;  
< becomes &lt;  
> becomes &gt;
```

## persistence

`persistence` gibt die Persistenz von Nachrichten an, die von einem Service gesendet wurden.

`persistence` ist optional und entspricht der Einstellung des MQMD-Felds [Persistence](#) .

Folgende Werte sind möglich:

### false

Die Nachrichten sind nicht persistent. Dies ist der Standardwert.

### true

Die Nachrichten sind persistent.

`persistence` kann durch Verwendung eines HTTP-Headers `ibm-mq-md-persistence`, der dieselben Werte annimmt, außer Kraft gesetzt werden.

## **mqmdFormat**

Dieses Attribut wird verwendet, um den Wert des MQMD-Formatfeldes in Nachrichten festzulegen, die vom MQ Service Provider gesendet werden. Es wird allerdings nur verwendet, wenn der MQ Service Provider für die Verwendung von z/OS Connect-Datenkonvertierungen konfiguriert wurde. Andernfalls wird das Attribut ignoriert.

Wenn Sie dieses Attribut nicht angeben und Datenkonvertierungen verwendet werden, sind beim Senden von Nachrichten für das MQMD-Formatfeld Leerzeichen festgelegt. Der Wert für dieses Attribut darf maximal acht Zeichen umfassen.

## **userName**

Der Benutzername, den der MQ Service Provider IBM MQ zu Authentifizierungs- und Autorisierungszwecken angibt.

Wenn Sie dieses Attribut nicht angeben, wird das Attribut **userName** in der Verbindungsfactory verwendet, auf die das Attribut **connectionFactory** verweist.

Wenn ein Attribut **userName** in der referenzierten Verbindungsfactory und in MQ Service Provider angegeben ist, wird der Wert MQ Service Provider verwendet.

Wenn Sie dieses Attribut angeben, müssen Sie das Attribut **password** angeben.

## **Kennwort**

Das Kennwort, das der MQ Service Provider IBM MQ zu Authentifizierungs- und Autorisierungszwecken angibt.

Sie können das Kennwort als unverschlüsselter Text angeben, dies ist jedoch nicht zu empfehlen. Sie sollten das Kennwort mit dem Tool **securityUtility** codieren, das mit z/OS Connect bereitgestellt wird, und hierfür die Codierungsoption verwenden. Weitere Informationen finden Sie unter [Liberty: Befehl securityUtility](#).

Wenn Sie dieses Attribut nicht angeben, wird das Attribut 'password' in der Verbindungsfactory verwendet, auf die über das Attribut **connectionFactory** verwiesen wird.

Wenn das Attribut 'password' sowohl in der referenzierten Verbindungsfactory als auch auf dem MQ Service Provider angegeben wird, wird der Wert des MQ Service Providers verwendet.

Wenn Sie dieses Attribut angeben, müssen auch das Attribut **userName** angegeben werden.

## **useCallerPrincipal**

Wenn eine Anforderung an z/OS Connect gestellt wird, authentifiziert sich das aufrufende Programm bei z/OS Connect. Der Name des authentifizierten Principals kann zu Authentifizierungs- und Autorisierungszwecken an IBM MQ übergeben werden.

Legen Sie zu diesem Zweck für **useCallerPrincipal** den Wert 'true' fest.

Beim Herstellen einer Verbindung zu IBM MQ wird der Name des Principals, aber kein Kennwort verwendet. Alle Werte, die für die Attribute **password** und **userName** angegeben werden, werden ignoriert.

## **receiveTextCCSID**

Die CCSID, die verwendet wird, wenn eine Datenkonvertierung erhalten und `javax.jms.TextMessage` verarbeitet wird (d. h. HTTP GET oder HTTP DELETE bei einem unidirektionalen Service oder beim Abruf einer Antwortnachricht für einen bidirektionalen Service).

Der Text in der Nachricht wird in die über **receiveTextCCSID** angegebene CCSID umgewandelt.

## HTTP-Header, die mit dem MQ Service Provider verwendet werden können

Nur bei Ausgabe einer HTTP POST-Anforderung erwartet der MQ Service Provider bestimmte HTTP-Header.

In diesem Fall muss der Content-Type-Header auf "application/json" gesetzt sein. Wenn Sie als Teil dieses Headers einen Zeichensatz angeben, muss sein Wert utf-8 lauten.

Beispiel: Content-Type=application/json;charset=utf-8.

Andere HTTP-Header können in der HTTP-Anforderung angegeben werden, um das Verhalten des MQ Service Providers zu ändern. Diese werden in den folgenden Abschnitten erläutert. Alle anderen HTTP-Header werden ignoriert.

### **ibm-mq-md-msgID**

Dieser Header kann angegeben werden, wenn HTTP GET- oder HTTP DELETE-Anforderungen für unidirektionale Services ausgegeben werden.

Der Wert dieses Headers wird verwendet, um einen Nachrichtenselektor zu generieren und somit eine Nachricht mit der angegebenen Nachrichten-ID auszuwählen. Wenn auch ein „[ibm-mq-md-correlID](#)“ auf [Seite 212](#)-Header angegeben ist, wird ein Nachrichtenselektor generiert, der beide IDs abgleicht.

Informationen zum Wert des Formats für diesen Header finden Sie im Abschnitt [msgId: HTTP-Entitätsheader x-msg-msgId](#).

### **ibm-mq-md-correlID**

Dieser Header kann angegeben werden, wenn eine HTTP POST-Anforderung ausgegeben wird. In diesem Fall wird er zur Definition des MQMD-Felds [CorrelID](#) der Nachricht verwendet, die gesendet wird.

Dieser Header kann auch angegeben werden, wenn HTTP GET- oder HTTP DELETE-Anforderungen für unidirektionale Services ausgegeben werden. Der Wert dieses Headers wird verwendet, um einen Nachrichtenselektor zu generieren und somit eine Nachricht mit der angegebenen Korrelations-ID auszuwählen. Wenn auch ein „[ibm-mq-md-msgID](#)“ auf [Seite 212](#)-Header angegeben ist, wird ein Nachrichtenselektor generiert, der beide abgleicht.

Informationen zum Wert des Formats für diesen Header finden Sie im Abschnitt [correlId: HTTP-Entitätsheader x-msg-correlId](#).

### **ibm-mq-pmo-retain**

Sie können diesen Header mit dem Wert TRUE angeben, wenn Sie eine HTTP POST-Anforderung gestützt durch ein Topic an einen unidirektionalen Service ausgeben. Es wird dann eine ständige Veröffentlichung generiert. Weitere Informationen finden Sie im Abschnitt [Ständige Veröffentlichungen](#).

### **ibm-mq-usr**

Sie können diesen Header verwenden, um Nachrichteneigenschaften in den IBM MQ-Nachrichten anzugeben, die infolge von HTTP POST-Anforderungen an unidirektionale und bidirektionale Services gesendet werden.

Einzelheiten zum Wert des Formats für diesen Header finden Sie im Abschnitt [usr: HTTP-Entitätsheader "x-msg-usr"](#).

Zwar verwendet der MQ Service Provider einen anderen Namen, Informationen zum Wert des Formats dieses Headers finden Sie jedoch im Abschnitt [require-headers: HTTP-Anforderungsheader x-msg-require-headers](#).

## 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: