

7.5

IBM WebSphere MQ Konfigurationsreferenz

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 153 gelesen werden.

Diese Ausgabe bezieht sich auf Version 7 Release 5 von IBM® WebSphere MQ und auf alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

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

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

Inhaltsverzeichnis

Konfigurationsreferenz.....	5
Beispielkonfigurationsdaten für alle Plattformen.....	5
Verwendung der Kommunikationsbeispiele.....	7
Beispielkonfiguration - IBM WebSphere MQ for Windows.....	8
Beispielkonfiguration - IBM WebSphere MQ for AIX.....	18
Beispielkonfiguration - IBM WebSphere MQ for HP-UX.....	25
Beispielkonfiguration - IBM WebSphere MQ for Solaris.....	31
Beispielkonfiguration - IBM WebSphere MQ for Linux.....	36
Warteschlangennamen.....	43
Weitere Objektnamen.....	45
Auflösung des Warteschlangennamens.....	47
Was ist die Auflösung von Warteschlangennamen?.....	48
System-und Standardobjekte.....	49
Objekte der Windows-Standardkonfiguration.....	52
SYSTEM.BASE.TOPIC.....	54
Informationen zu Zeilengruppen.....	55
Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen.....	57
Kanalattribute.....	58
Kanalattribute und Kanaltypen.....	59
Kanalattribute in alphabetischer Reihenfolge.....	62
Clusterbefehle für IBM WebSphere MQ.....	91
Befehle für die Warteschlangenmanagerdefinition.....	92
Befehle für die Kanaldefinition.....	93
Befehle für die Warteschlangendefinition.....	96
DISPLAY CLUSQMGR.....	98
SUSPEND QMGR und RESUME QMGR.....	100
REFRESH CLUSTER.....	101
RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen.....	102
Lastausgleich.....	103
Aufruf des Exits für Clusterauslastung und Datenstrukturen.....	116
Kanalprogramme.....	141
Umgebungsvariablen.....	142
Beispiel einer Nachrichtenkanalplanung für verteilte Plattformen.....	147
Inhalt dieses Beispiels.....	147
Beispiel ausführen.....	151
Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek.....	151
Bemerkungen.....	153
Informationen zu Programmierschnittstellen.....	154
Marken.....	155

Konfigurationsreferenz

Konfigurieren Sie WebSphere MQ mithilfe der Referenzinformationen in diesem Abschnitt.

Die Konfigurationsreferenzinformationen werden in folgenden Unterabschnitten bereitgestellt:

Zugehörige Tasks

[Konfiguration](#)

Beispielkonfigurationsdaten

In den Konfigurationsbeispielen werden die Aufgaben beschrieben, die zum Aufbau eines funktionierenden WebSphere MQ-Netztes auszuführen sind. Zu den Aufgaben gehört die Einrichtung von Sender- und Empfängerkanälen von WebSphere 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 in der [WebSphere MQ-Scriptbefehlsreferenz \(MQSC\)](#).

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

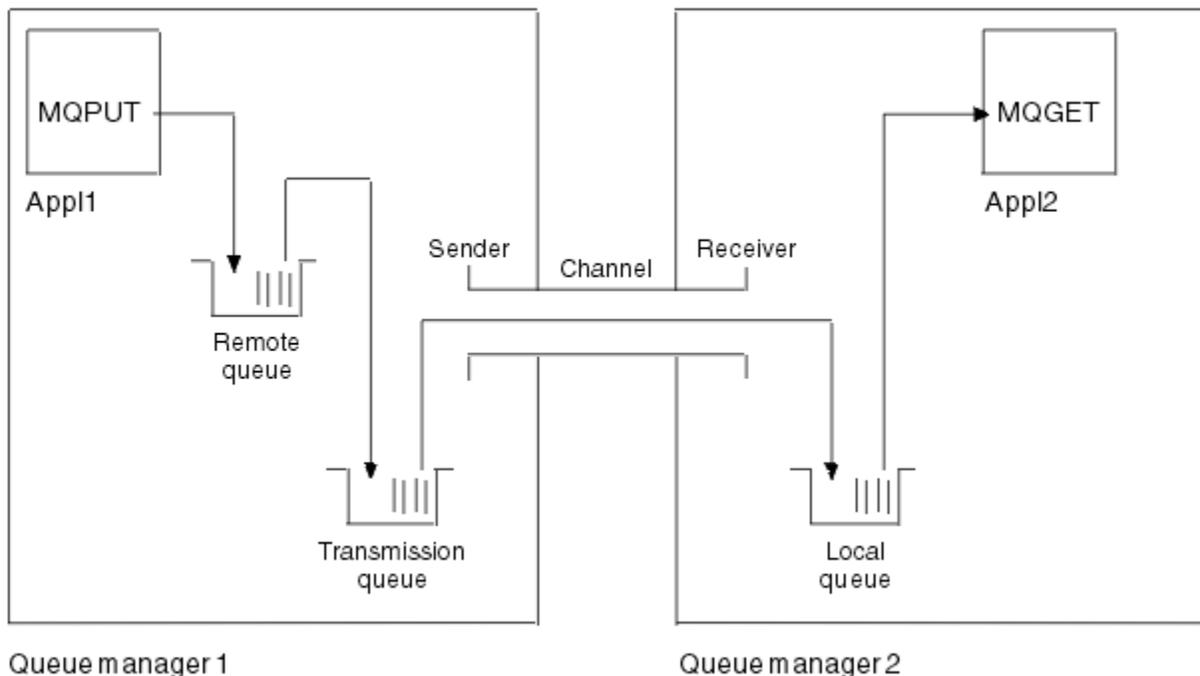


Abbildung 1. In der Beispielkonfiguration einzurichtender WebSphere MQ-Kanal

Dies ist ein einfaches Beispiel, das eine Einführung in die grundlegenden Elemente des WebSphere 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
- ein 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 einen 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 V5.9
 - IBM Communications Server for AIX V6.3
 - Hewlett-Packard SNAplus2
 - IBM i
 - Data Connection SNAP-IX Version 7 oder höher
 - OS/390 Version 2 Release 4

- TCP
 - Microsoft Windows
 - AIX Version 4 Release 1.4
 - HP-UX Version 10.2 oder höher
 - Sun Solaris Release 2.4 oder höher
 - IBM i
 - TCP for z/OS
 - HP Tru64 UNIX
- NetBIOS
- SPX

Zugehörige Tasks

Konfiguration

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.

Wenn Sie ein Windows-, UNIX- oder Linux®-System verwenden, müssen Sie verschiedene Umgebungsvariablen festlegen, bevor Sie beginnen, die Anweisungen für Ihre Plattform zu befolgen. Legen Sie die Umgebungsvariablen fest, indem Sie einen der folgenden Befehle eingeben:

- Unter Windows:

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

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

- Auf UNIX and Linux-Systemen:

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

Dabei steht *MQ_INSTALLATION_PATH* für das Verzeichnis, in dem IBM WebSphere 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 Warteschlangenmanagercluster konfigurieren. Die hier angegebenen Werte für die Konfiguration der Kommunikation gelten weiterhin.

Es sind Beispielkonfigurationen für die folgenden Plattformen vorhanden:

- „Beispielkonfiguration - IBM WebSphere MQ for Windows“ auf Seite 8
- „Beispielkonfiguration - IBM WebSphere MQ for AIX“ auf Seite 18

- [„Beispielkonfiguration - IBM WebSphere MQ for HP-UX“](#) auf Seite 25
- [„Beispielkonfiguration - IBM WebSphere MQ for Solaris“](#) auf Seite 31
- [„Beispielkonfiguration- IBM WebSphere MQ für Linux“](#) auf Seite 36

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 WebSphere MQ-Plattform und WebSphere 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, z. B. 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

[„Beispielkonfigurationsdaten“](#) auf Seite 5

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

Zugehörige Verweise

[setmqenv](#)

Beispielkonfiguration - IBM WebSphere MQ for Windows

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM WebSphere MQ for Windows und IBM WebSphere 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 [„IBM WebSphere MQ-Konfiguration“](#) auf Seite 12 finden Sie eine Beschreibung der Beispielprogramme und Befehle.

Unter „[Beispielkonfigurationsdaten](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

Eine LU 6.2-Verbindung einrichten

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

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP von AnyNet finden Sie online in den folgenden IBM-Dokumentationen: [AnyNet SNA over TCP/IP](#), [SNA-Knoten-Operationen](#) und [Communications Server for Windows](#)

Eine TCP-Verbindung einrichten

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

Der WebSphere MQ-Befehl zum Starten des Empfangsprogramms von WebSphere MQ for TCP lautet:

```
runmqclsr -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. Gehen Sie zu „[IBM WebSphere MQ-Konfiguration](#)“ auf Seite 12.

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 WebSphere 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 WebSphere 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 WebSphere 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. Siehe [LAN-Adaptornummer einrichten](#).

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
```

Die Option MCATYPE (THREAD) muss angegeben werden, weil unter Windows die Senderkanäle als Threads ausgeführt werden müssen.

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 WebSphere 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](#).

SPX-Verbindung aufbauen

Eine SPX-Verbindung findet nur bei einem Client und Server Anwendung, auf denen Windows XP und Windows 2003 Server ausgeführt werden.

Dieser Abschnitt enthält Informationen über

- IPX-/SPX-Parameter
- SPX-Adressierung
- Empfangen über SPX

IPX-/SPX-Parameter

Ausführliche Informationen zur Verwendung und Einstellung der NWLink IPX- und SPX-Parameter finden Sie in der Microsoft-Dokumentation. Die IPX-/SPX-Parameter befinden sich unter folgenden Pfaden in der Registrierung:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkSPX\Parameters  
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkIPX\Parameters
```

SPX-Adressierung

WebSphere MQ verwendet bei der Verbindungsherstellung die SPX-Adresse der einzelnen Maschinen. Die SPX-Adresse wird in folgendem Format angegeben:

```
network.node(socket)
```

Dabei gilt Folgendes:

network

die Netzadresse (4 Byte) des Netzes, in dem sich die ferne Maschine befindet,

node

die Knotenadresse (6 Byte), die der LAN-Adresse des LAN-Adapters auf der fernen Maschine entspricht.

socket

die Socketnummer (2 Byte), über die die ferne Maschine empfangsbereit ist.

WebSphere MQ verwendet standardmäßig die Socketnummer 5E86. Die Standardsocketnummer kann in der Windows-Registrierung oder in der Konfigurationsdatei des Warteschlangenmanagers "qm.ini" geändert werden. Die Zeilen in der Windows-Registrierung können wie folgt lauten:

```
SPX:  
SOCKET=n
```

Weitere Informationen zu Werten, die in der Datei "qm.ini" festgelegt werden können, finden Sie in „[Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen](#)“ auf Seite 57.

Die SPX-Adresse wird später im Parameter CONNAME der Senderkanaldefinition angegeben. Wenn sich die zu verbindenden WebSphere MQ-Systeme im gleichen Netz befinden, muss die Netzadresse nicht angegeben werden. Entsprechend muss die Netzadresse auch nicht angegeben werden, wenn das ferne System über die Standardsocketnummer (5E86) empfangsbereit ist. Eine vollständig qualifizierte SPX-Adresse im CONNAME-Parameter lautet:

```
CONNAME('network.node(socket)')
```

, wenn sich jedoch die Systeme im gleichen Netz befinden und die Standardsocketnummer verwendet wird, lautet der Parameter:

```
CONNAME(node)
```

Ein ausführliches Beispiel für die Kanalkonfigurationsparameter finden Sie in „[IBM WebSphere MQ-Konfiguration](#)“ auf Seite 12.

Empfangen über SPX

Das Empfangen von Kanalprogrammen wird als Antwort auf eine Startanforderung vom sendenden Kanal gestartet. Hierzu muss zunächst ein Listenerprogramm gestartet werden, damit eingehende Netzanforderungen erkannt werden und der zugehörige Kanal gestartet wird.

Sie sollten das WebSphere MQ-Empfangsprogramm verwenden.

WebSphere MQ-Empfangsprogramm verwenden

Verwenden Sie den Befehl RUNMQLSR zur Ausführung des durch WebSphere MQ bereitgestellten Empfangsprogramms, das neue Kanäle als Threads startet. Beispiel:

```
RUNMQLSR -t spx
```

Wenn Sie nicht die Standardwerte verwenden, können Sie einen alternativen Warteschlangenmanagernamen bzw. die zutreffende Socketnummer festlegen.

IBM WebSphere MQ-Konfiguration

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 WebSphere MQ Explorer benutzen.

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

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

3. 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 WebSphere MQ installiert ist.

4. 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.

Standardkonfiguration

Das Programm WebSphere MQ-PostCard führt Sie durch die Erstellung von Standardkonfigurationen.

Informationen zur Verwendung der PostCard-Anwendung finden Sie unter [Installation mit der PostCard-Anwendung überprüfen](#).

Basiskonfiguration

Warteschlangenmanager können aus IBM WebSphere MQ Explorer oder aus der Eingabeaufforderung erstellt und gestartet werden.

Bei Verwendung der Eingabeaufforderung:

1. 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.

2. Starten Sie den Warteschlangenmanager über den folgenden Befehl:

```
strmqm winnt
```

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

Kanalkonfiguration

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

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte des Windows-Warteschlangenmanagers zur Implementierung des in [Abbildung 1 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.

Es sind Beispiele für die Verbindung von WebSphere MQ for Windows und WebSphere MQ for AIX enthalten. Um eine Verbindung zu WebSphere MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle denen im Beispiel für Windows ein.

Anmerkung: Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von WebSphere 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 1. Arbeitsblatt zur Konfiguration für WebSphere MQ for Windows				
	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
Definition für lokalen Knoten				
A	Name des Warteschlangenmanagers		WINNT	
B	Name der lokalen Warteschlange		WINNT.LOCALQ	
Verbindung zu WebSphere MQ for AIX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 2 auf Seite 20 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	AIX	
D	Name der fernen Warteschlange		AIXREMOTEQ	
E	Name der Warteschlange auf fernem System	B	AIXLOCALQ	
F	Name der Übertragungswarteschlange		AIX	
G	Name des Senderkanals (SNA)		WINNT.AIX.SNA	
H	Name des Senderkanals (TCP)		WINNT.AIX.TCP	
I	Name des Empfängerkanals (SNA)	G	AIX.WINNT.SNA	

Tabelle 1. Arbeitsblatt zur Konfiguration für WebSphere MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
J	Name des Empfängerkanals (TCP)	H	AIX.WINNT.TCP	
Verbindung mit WebSphere MQ for HP Tru64 UNIX				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem HP Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	DECUX	
D	Name der fernen Warteschlange		DECUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	DECUX.LOCALQ	
F	Name der Übertragungswarteschlange		DECUX	
H	Name des Senderkanals (TCP)		DECUX.WINNT.TCP	
J	Name des Empfängerkanals (TCP)	H	WINNT.DECUX.TCP	
Verbindung zu WebSphere MQ for HP-UX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 3 auf Seite 27 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	HPUX	
D	Name der fernen Warteschlange		HPUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	HPUX.LOCALQ	
F	Name der Übertragungswarteschlange		HPUX	
G	Name des Senderkanals (SNA)		WINNT.HPUX.SNA	
H	Name des Senderkanals (TCP)		WINNT.HPUX.TCP	
I	Name des Empfängerkanals (SNA)	G	HPUX.WINNT.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	HPUX.WINNT.TCP	
Verbindung zu WebSphere MQ for Solaris				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 4 auf Seite 33 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	SOLARIS	
D	Name der fernen Warteschlange		SOLARIS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	SOLARIS.LOCALQ	
F	Name der Übertragungswarteschlange		SOLARIS	
G	Name des Senderkanals (SNA)		WINNT.SOLARIS.SNA	
H	Name des Senderkanals (TCP)		WINNT.SOLARIS.TCP	
I	Name des Empfängerkanals (SNA)	G	SOLARIS.WINNT.SNA	

Tabelle 1. Arbeitsblatt zur Konfiguration für WebSphere MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
J	Name des Empfängerkanals (TCP)	H	SOLARIS.WINNT.TCP	
Verbindung zu WebSphere MQ für Linux				
Die Werte in diesem Abschnitt der Tabelle müssen den in <u>Tabelle 5 auf Seite 40</u> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	LINUX	
D	Name der fernen Warteschlange		LINUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	LINUX.LOCALQ	
F	Name der Übertragungswarteschlange		LINUX	
G	Name des Senderkanals (SNA)		WINNT.LINUX.SNA	
H	Name des Senderkanals (TCP)		WINNT.LINUX.TCP	
I	Name des Empfängerkanals (SNA)	G	LINUX.WINNT.SNA	
J	Name des Empfängerkanals (TCP)	H	LINUX.WINNT.TCP	
C	Name des fernen Warteschlangenmanagers	A	AS400	
D	Name der fernen Warteschlange		AS400.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	AS400.LOCALQ	
F	Name der Übertragungswarteschlange		AS400	
G	Name des Senderkanals (SNA)		WINNT.AS400.SNA	
H	Name des Senderkanals (TCP)		WINNT.AS400.TCP	
I	Name des Empfängerkanals (SNA)	G	AS400.WINNT.SNA	
C	Name des fernen Warteschlangenmanagers	A	MVS	
D	Name der fernen Warteschlange		MVS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	MVS.LOCALQ	
F	Name der Übertragungswarteschlange		MVS	
G	Name des Senderkanals (SNA)		WINNT.MVS.SNA	
H	Name des Senderkanals (TCP)		WINNT.MVS.TCP	
I	Name des Empfängerkanals (SNA)	G	MVS.WINNT.SNA	
C	Name des fernen Warteschlangenmanagers	A	QSG	
D	Name der fernen Warteschlange		QSG.REMOTEQ	

Tabelle 1. Arbeitsblatt zur Konfiguration für WebSphere MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
E	Name der Warteschlange auf fernem System	B	QSG.SHAREDQ	
F	Name der Übertragungswarteschlange		QSG	
G	Name des Senderkanals (SNA)		WINNT.QSG.SNA	
H	Name des Senderkanals (TCP)		WINNT.QSG.TCP	
I	Name des Empfängerkanals (SNA)	G	QSG.WINNT.SNA	
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	VSE	
D	Name der fernen Warteschlange		VSE.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	VSE.LOCALQ	
F	Name der Übertragungswarteschlange		VSE	
G	Name des Senderkanals		WINNT.VSE.SNA	
I	Name des Empfängerkanals	G	VSE.WINNT.SNA	

Senderkanaldefinitionen unter Verwendung der SNA in IBM WebSphere MQ für 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.SNA) chltype(sdr) +             G
  trptype(lu62) +
  conname(AIXCPIC) +                               18
  xmitq(AIX) +                                     F
  replace

```

Empfängerkanaldefinitionen unter Verwendung der SNA in IBM WebSphere MQ for Windows
Codebeispiel:

```

def ql (WINNT.LOCALQ) replace                       B

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

```

Senderkanaldefinitionen über TCP/IP in IBM WebSphere 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.TCP) chltype(sdr) + trptype(tcp) + conname(remote_tcpip_hostname) + xmitq(AIX) + replace H F

```

Empfängerkanaldefinitionen unter Verwendung von TCP in IBM WebSphere MQ for Windows
Codebeispiel:

```

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

```

Automatischer Start

In WebSphere 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 WebSphere 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 WebSphere MQ-Service gelesen werden.

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

Kanäle als Prozesse oder Threads ausführen

WebSphere MQ for Windows bietet die Flexibilität, sendende Kanäle als Windows-Prozesse oder Windows-Threads auszuführen. Dies wird im Parameter MCATYPE 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.

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.

Unter WebSphere MQ for Windows verwenden Sie den WebSphere MQ Explorer, um in der Registrierung den Parameter *PipeLineLength* festzulegen.

Anmerkung:

1. *PipeLineLength* findet nur in Version 5.2 und späteren Produkten Anwendung.

2. 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.

Beispielkonfiguration - IBM WebSphere MQ for AIX

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

In den Beispielen werden die folgenden Plattformen behandelt:

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

Unter „[Beispielkonfigurationsdaten](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

Eine LU 6.2-Verbindung einrichten

Beschreibt die für eine LU 6.2-Verbindung erforderlichen Parameter.

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP finden Sie online in der IBM 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 WebSphere MQ-Befehl zum Starten des Empfangsprogramms von WebSphere MQ for TCP lautet:

```
runmqldr -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, ersetzen Sie dabei `MQ_INSTALLATION_PATH` durch das übergeordnete Verzeichnis, in dem WebSphere 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. Gehen Sie zu [„IBM WebSphere MQ for AIX -Konfiguration“](#) auf Seite 19.

IBM WebSphere MQ for AIX -Konfiguration

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 sna` erhöhen. (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 WebSphere MQ installiert ist.
5. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
6. Unter AIX können Sie einen Trace der WebSphere MQ-Komponenten durch Eingabe der Standard-Trace-Befehle für WebSphere MQ oder mithilfe des AIX-Systemtrace starten. Weitere Informationen zu WebSphere 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 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 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

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 [Abbildung 1 auf Seite 5](#) beschriebenen Kanals.

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

Zur Herstellung einer Verbindung mit WebSphere MQ for AIX und WebSphere MQ for Windows sind Beispiele angeführt. Um eine Verbindung zu WebSphere MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle denen im Beispiel für Windows ein.

Anmerkung: Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von WebSphere 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.

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
Definition für lokalen Knoten				
A	Name des Warteschlangenmanagers		AIX	
B	Name der lokalen Warteschlange		AIXLOCALQ	
Verbindung zu WebSphere MQ for Windows				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 1 auf Seite 13 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	WINNT	

Tabelle 2. Arbeitsblatt zur Konfiguration von WebSphere MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
D	Name der fernen Warteschlange		WINNT.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	WINNT.LOCALQ	
F	Name der Übertragungswarteschlange		WINNT	
G	Name des Senderkanals (SNA)		AIX.WINNT.SNA	
H	Name des Senderkanals (TCP/IP)		AIX.WINNT.TCP	
I	Name des Empfängerkanals (SNA)	G	WINNT.AIX.SNA	
J	Name des Empfängerkanals (TCP)	H	WINNT.AIX.TCP	
Verbindung mit WebSphere MQ for HP Tru64 UNIX				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem HP Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	DECUX	
D	Name der fernen Warteschlange		DECUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	DECUX.LOCALQ	
F	Name der Übertragungswarteschlange		DECUX	
H	Name des Senderkanals (TCP)		DECUX.AIX.TCP	
J	Name des Empfängerkanals (TCP)	H	AIX.DECUX.TCP	
Verbindung zu WebSphere MQ for HP-UX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 3 auf Seite 27 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	HPUX	
D	Name der fernen Warteschlange		HPUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	HPUX.LOCALQ	
F	Name der Übertragungswarteschlange		HPUX	
G	Name des Senderkanals (SNA)		AIX.HPUX.SNA	
H	Name des Senderkanals (TCP)		AIX.HPUX.TCP	
I	Name des Empfängerkanals (SNA)	G	HPUX.AIX.SNA	
J	Name des Empfängerkanals (TCP)	H	HPUX.AIX.TCP	
Verbindung zu WebSphere MQ for Solaris				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 4 auf Seite 33 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	SOLARIS	

Tabelle 2. Arbeitsblatt zur Konfiguration von WebSphere MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
D	Name der fernen Warteschlange		SOLARIS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	SOLARIS.LOCALQ	
F	Name der Übertragungswarteschlange		SOLARIS	
G	Name des Senderkanals (SNA)		AIX.SOLARIS.SNA	
H	Name des Senderkanals (TCP/IP)		AIX.SOLARIS.TCP	
I	Name des Empfängerkanals (SNA)	G	SOLARIS.AIX.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	SOLARIS.AIX.TCP	
Verbindung zu WebSphere MQ für Linux				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 5 auf Seite 40 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	LINUX	
D	Name der fernen Warteschlange		LINUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	LINUX.LOCALQ	
F	Name der Übertragungswarteschlange		LINUX	
G	Name des Senderkanals (SNA)		AIX.LINUX.SNA	
H	Name des Senderkanals (TCP/IP)		AIX.LINUX.TCP	
I	Name des Empfängerkanals (SNA)	G	LINUX.AIX.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	LINUX.AIX.TCP	
C	Name des fernen Warteschlangenmanagers	A	AS400	
D	Name der fernen Warteschlange		AS400.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	AS400.LOCALQ	
F	Name der Übertragungswarteschlange		AS400	
G	Name des Senderkanals (SNA)		AIXAS400.SNA	
H	Name des Senderkanals (TCP)		AIXAS400TCP-	
I	Name des Empfängerkanals (SNA)	G	AS400.AIX.SNA	
C	Name des fernen Warteschlangenmanagers	A	MVS	
D	Name der fernen Warteschlange		MVS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	MVS.LOCALQ	

Tabelle 2. Arbeitsblatt zur Konfiguration von WebSphere MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
F	Name der Übertragungswarteschlange		MVS	
G	Name des Senderkanals (SNA)		AIX.MVS.SNA	
H	Name des Senderkanals (TCP)		AIX.MVS.TCP	
I	Name des Empfängerkanals (SNA)	G	MVS.AIX.SNA	
C	Name des fernen Warteschlangenmanagers	A	QSG	
D	Name der fernen Warteschlange		QSG.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	QSG.SHAREDQ	
F	Name der Übertragungswarteschlange		QSG	
G	Name des Senderkanals (SNA)		AIX.QSG.SNA	
H	Name des Senderkanals (TCP)		AIX.QSG.TCP	
I	Name des Empfängerkanals (SNA)	G	QSG.AIX.SNA	
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	VSE	
D	Name der fernen Warteschlange		VSE.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	VSE.LOCALQ	
F	Name der Übertragungswarteschlange		VSE	
G	Name des Senderkanals		AIX.VSE.SNA	
I	Name des Empfängerkanals	G	VSE.AIX.SNA	

Senderkanaldefinitionen unter Verwendung der SNA in IBM WebSphere MQ

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) ch1type(sdr) +               G
  tr1type(lu62) +
  conname('WINNTCPIC') +                             17
  xmitq(WINNT) +                                       F
  replace
  
```

Empfängerkanaldefinitionen unter Verwendung der SNA in IBM WebSphere MQ for AIX

Beispielbefehle:

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

Konfiguration von IBM WebSphere MQ for AIX TPN

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 steht *aix* für den Namen des Warteschlangenmanagers (A) und *MQ_INSTALLATION_PATH* für das übergeordnete Verzeichnis, in dem WebSphere 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.

Senderkanaldefinitionen unter Verwendung von TCP in IBM WebSphere 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
```

Empfängerkanaldefinitionen unter Verwendung von TCP in IBM WebSphere MQ for AIX

Beispielbefehle:

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

Beispielkonfiguration - IBM WebSphere MQ for HP-UX

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

Die folgenden Plattformen werden behandelt:

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

Unter „Beispielkonfigurationsdaten“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

Eine LU 6.2-Verbindung einrichten

Beschreibt die für eine LU 6.2-Verbindung erforderlichen Parameter.

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP finden Sie online in der IBM Dokumentation [Communications Server](#) und in der HP-Dokumentation [HP-UX SNAplus2 Installation Guide](#).

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, ersetzen Sie dabei `MQ_INSTALLATION_PATH` durch das übergeordnete Verzeichnis, in dem WebSphere 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. Gehen Sie zu „[Konfiguration von IBM WebSphere MQ for HP-UX](#)“ auf Seite 26.

Konfiguration von IBM WebSphere 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 WebSphere 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 aus der UNIX-Befehlszeile mithilfe 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 aus der UNIX-Befehlszeile mithilfe folgenden Befehls:

```
strmqm hpux
```

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

Kanalkonfiguration

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 [Abbildung 1 auf Seite 5](#) beschriebenen Kanals.

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

Zur Herstellung einer Verbindung mit WebSphere MQ for HP-UX und WebSphere MQ for Windows sind Beispiele angeführt. Um eine Verbindung zu WebSphere MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle denen im Beispiel für Windows ein.

Anmerkung: Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von WebSphere 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 3. Arbeitsblatt zur Konfiguration für WebSphere MQ for HP-UX</i>				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
Definition für lokalen Knoten				
A	Name des Warteschlangenmanagers		HPUX	
B	Name der lokalen Warteschlange		HPUX.LOCALQ	
Verbindung zu WebSphere MQ for Windows				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 1 auf Seite 13 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	WINNT	
D	Name der fernen Warteschlange		WINNT.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	WINNT.LOCALQ	
F	Name der Übertragungswarteschlange		WINNT	
G	Name des Senderkanals (SNA)		HPUX.WINNT.SNA	
H	Name des Senderkanals (TCP/IP)		HPUX.WINNT.TCP	
I	Name des Empfängerkanals (SNA)	G	WINNT.HPUX.SNA	
J	Name des Empfängerkanals (TCP)	H	WINNT.HPUX.TCP	
Verbindung zu WebSphere MQ for AIX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 2 auf Seite 20 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	AIX	
D	Name der fernen Warteschlange		AIXREMOTEQ	
E	Name der Warteschlange auf fernem System	B	AIXLOCALQ	
F	Name der Übertragungswarteschlange		AIX	
G	Name des Senderkanals (SNA)		HPUX.AIX.SNA	
H	Name des Senderkanals (TCP)		HPUX.AIX.TCP	
I	Name des Empfängerkanals (SNA)	G	AIX.HPUX.SNA	
J	Name des Empfängerkanals (TCP)	H	AIX.HPUX.TCP	

Tabelle 3. Arbeitsblatt zur Konfiguration für WebSphere MQ for HP-UX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
Verbindung mit WebSphere MQ for HP Tru64 UNIX				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem HP Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	DECUX	
D	Name der fernen Warteschlange		DECUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	DECUX.LOCALQ	
F	Name der Übertragungswarteschlange		DECUX	
H	Name des Senderkanals (TCP)		DECUX.HPUX.TCP	
J	Name des Empfängerkanals (TCP)	H	HPUX.DECUX.TCP	
Verbindung zu WebSphere MQ for Solaris				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 4 auf Seite 33 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	SOLARIS	
D	Name der fernen Warteschlange		SOLARIS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	SOLARIS.LOCALQ	
F	Name der Übertragungswarteschlange		SOLARIS	
G	Name des Senderkanals (SNA)		HPUX.SOLARIS.SNA	
H	Name des Senderkanals (TCP/IP)		HPUX.SOLARIS.TCP	
I	Name des Empfängerkanals (SNA)	G	SOLARIS.HPUX.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	SOLARIS.HPUX.TCP	
Verbindung zu WebSphere MQ für Linux				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 5 auf Seite 40 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	LINUX	
D	Name der fernen Warteschlange		LINUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	LINUX.LOCALQ	
F	Name der Übertragungswarteschlange		LINUX	
G	Name des Senderkanals (SNA)		HPUX.LINUX.SNA	
H	Name des Senderkanals (TCP/IP)		HPUX.LINUX.TCP	
I	Name des Empfängerkanals (SNA)	G	LINUX.HPUX.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	LINUX.HPUX.TCP	

Tabelle 3. Arbeitsblatt zur Konfiguration für WebSphere MQ for HP-UX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
C	Name des fernen Warteschlangenmanagers	A	AS400	
D	Name der fernen Warteschlange		AS400.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	AS400.LOCALQ	
F	Name der Übertragungswarteschlange		AS400	
G	Name des Senderkanals (SNA)		HPUX.AS400.SNA	
H	Name des Senderkanals (TCP/IP)		HPUX.AS400.TCP	
I	Name des Empfängerkanals (SNA)	G	AS400.HPUX.SNA	
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	VSE	
D	Name der fernen Warteschlange		VSE.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	VSE.LOCALQ	
F	Name der Übertragungswarteschlange		VSE	
G	Name des Senderkanals		HPUX.VSE.SNA	
I	Name des Empfängerkanals	G	VSE.HPUX.SNA	

Senderkanaldefinitionen unter Verwendung der SNA in IBM WebSphere 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.SNA) chltype(sdr) +           G
  trptype(lu62) +
  conname('WINNTCPIC') +                          16
  xmitq(WINNT) +                                    F
  replace

```

Empfängerkanaldefinitionen unter Verwendung der SNA in IBM WebSphere MQ for HP-UX
 Beispielbefehle:

```

def ql (HPUX.LOCALQ) replace                       B

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

```

Aufrufbares Transaktionsprogramm für IBM WebSphere 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/interops/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* für das übergeordnete Verzeichnis, in dem WebSphere MQ installiert ist.

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

Senderkanaldefinitionen unter Verwendung von TCP in IBM WebSphere 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

```

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

Beispielbefehle:

```

def ql (HPUX.LOCALQ) replace                       B

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

```

Beispielkonfiguration - IBM WebSphere MQ for Solaris

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

Für folgende Plattformen stehen Beispiele zur Verfügung:

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

Unter „Beispielkonfigurationsdaten“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

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 online in der IBM IBM Dokumentation [Communications Server](#), in der MetaSwitch-Dokumentation [SNAP-IX Administration Guide](#) und in der Oracle-Dokumentation [Configuring Intersystem Communications \(ISC\)](#).

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/amqcrsta amqcrsta  
[-m queue.manager.name]
```

`MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem WebSphere 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:

Wie geht es weiter?

Die TCP/IP-Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Gehen Sie zu „Konfiguration von IBM WebSphere MQ for Solaris“ auf Seite 32.

Konfiguration von IBM WebSphere 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 WebSphere 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 aus der UNIX-Befehlszeile mithilfe 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 aus der UNIX-Befehlszeile mithilfe folgenden Befehls:

```
strmqm solaris
```

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

Kanalkonfiguration

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** aus einer UNIX-Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Zur Herstellung einer Verbindung mit WebSphere MQ for Solaris und WebSphere MQ for Windows sind Beispiele angeführt. Um eine Verbindung zu WebSphere MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle denen im Beispiel für Windows ein.

Anmerkung: Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von WebSphere 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 4. Arbeitsblatt zur Konfiguration von WebSphere MQ for Solaris</i>				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
Definition für lokalen Knoten				
A	Name des Warteschlangenmanagers		SOLARIS	
B	Name der lokalen Warteschlange		SOLARIS.LOCALQ	
Verbindung zu WebSphere MQ for Windows				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 1 auf Seite 13 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	WINNT	
D	Name der fernen Warteschlange		WINNT.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	WINNT.LOCALQ	
F	Name der Übertragungswarteschlange		WINNT	
G	Name des Senderkanals (SNA)		SOLARIS.WINNT.SNA	
H	Name des Senderkanals (TCP/IP)		SOLARIS.WINNT.TCP	
I	Name des Empfängerkanals (SNA)	G	WINNT.SOLARIS.SNA	
J	Name des Empfängerkanals (TCP)	H	WINNT.SOLARIS.TCP	
Verbindung zu WebSphere MQ for AIX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 2 auf Seite 20 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	AIX	
D	Name der fernen Warteschlange		AIXREMOTEQ	
E	Name der Warteschlange auf fernem System	B	AIXLOCALQ	
F	Name der Übertragungswarteschlange		AIX	
G	Name des Senderkanals (SNA)		SOLARIS.AIX.SNA	
H	Name des Senderkanals (TCP)		SOLARIS.AIX.TCP	

Tabelle 4. Arbeitsblatt zur Konfiguration von WebSphere MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
I	Name des Empfängerkanals (SNA)	G	AIX.SOLARIS.SNA	
J	Name des Empfängerkanals (TCP)	H	AIX.SOLARIS.TCP	
Verbindung zu MQSeries for Compaq Tru64 Unix				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem Compaq Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	DECUX	
D	Name der fernen Warteschlange		DECUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	DECUX.LOCALQ	
F	Name der Übertragungswarteschlange		DECUX	
H	Name des Senderkanals (TCP)		DECUX.SOLARIS.TCP	
J	Name des Empfängerkanals (TCP)	H	SOLARIS.DECUX.TCP	
Verbindung zu WebSphere MQ für HP-UX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 3 auf Seite 27 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	HPUX	
D	Name der fernen Warteschlange		HPUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	HPUX.LOCALQ	
F	Name der Übertragungswarteschlange		HPUX	
G	Name des Senderkanals (SNA)		SOLARIS.HPUX.SNA	
H	Name des Senderkanals (TCP)		SOLARIS.HPUX.TCP	
I	Name des Empfängerkanals (SNA)	G	HPUX.SOLARIS.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	HPUX.SOLARIS.TCP	
Verbindung zu WebSphere MQ für Linux				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 5 auf Seite 40 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	LINUX	
D	Name der fernen Warteschlange		LINUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	LINUX.LOCALQ	
F	Name der Übertragungswarteschlange		LINUX	
G	Name des Senderkanals (SNA)		SOLARIS.LINUX.SNA	
H	Name des Senderkanals (TCP/IP)		SOLARIS.LINUX.TCP	

Tabelle 4. Arbeitsblatt zur Konfiguration von WebSphere MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
I	Name des Empfängerkanals (SNA)	G	LINUX.SOLARIS.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	LINUX.SOLARIS.TCP	
C	Name des fernen Warteschlangenmanagers	A	AS400	
D	Name der fernen Warteschlange		AS400.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	AS400.LOCALQ	
F	Name der Übertragungswarteschlange		AS400	
G	Name des Senderkanals (SNA)		SOLARIS.AS400.SNA	
H	Name des Senderkanals (TCP)		SOLARIS.AS400.TCP	
I	Name des Empfängerkanals (SNA)	G	AS400.SOLARIS.SNA	
C	Name des fernen Warteschlangenmanagers	A	MVS	
D	Name der fernen Warteschlange		MVS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	MVS.LOCALQ	
F	Name der Übertragungswarteschlange		MVS	
G	Name des Senderkanals (SNA)		SOLARIS.MVS.SNA	
H	Name des Senderkanals (TCP)		SOLARIS.MVS.TCP	
I	Name des Empfängerkanals (SNA)	G	MVS.SOLARIS.SNA	
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	VSE	
D	Name der fernen Warteschlange		VSE.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	VSE.LOCALQ	
F	Name der Übertragungswarteschlange		VSE	
G	Name des Senderkanals		SOLARIS.VSE.SNA	
I	Name des Empfängerkanals	G	VSE.SOLARIS.SNA	

Senderkanaldefinitionen unter Verwendung von SNAP-IX SNA in IBM WebSphere MQ for Solaris
Mustercode:

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

F

```

replace

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

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

```

Empfängerkanaldefinitionen unter Verwendung der SNA in IBM WebSphere MQ for Solaris
Mustercode:

```

def ql (SOLARIS.LOCALQ) replace
B

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

```

Senderkanaldefinitionen unter Verwendung von TCP in IBM WebSphere MQ for Solaris
Mustercode:

```

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

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

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

```

Empfängerkanaldefinitionen über TCP/IP in IBM WebSphere MQ for Solaris
Mustercode:

```

def ql (SOLARIS.LOCALQ) replace
B

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

```

Beispielkonfiguration- IBM WebSphere MQ für Linux

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

Für folgende Plattformen stehen Beispiele zur Verfügung:

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

- VSE/ESA

Unter „[Beispielkonfigurationsdaten](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

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 WebSphere MQ for Linux (x86 -Plattform). Es gilt nicht für WebSphere MQ for Linux (Plattformx86-64), WebSphere MQ for Linux (Plattform zSeries s390x) oder WebSphere MQ for Linux (Plattform Power).

Die neuesten Informationen zur Konfiguration von SNA über TCP/IP finden Sie im Administratorhandbuch für Ihre Version von Linux in der folgenden Dokumentation: [Communications Server für Linux -Bibliothek](#).

TCP-Verbindung unter Linux aufbauen

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 WebSphere 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/amqcrista amqcrista
[-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/amqcrista amqcrista -m QM1
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrista amqcrista -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 WebSphere 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 `WebSphere MQ`. Fügen Sie die folgende Zeilengruppe zur Datei hinzu:

```
# WebSphere 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 WebSphere 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. Gehen Sie zu „Konfiguration von IBM WebSphere MQ for Linux“ auf Seite 39.

Konfiguration von IBM WebSphere MQ for Linux

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 WebSphere 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 aus der UNIX-Befehlszeile mithilfe 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 aus der UNIX-Befehlszeile mithilfe folgenden Befehls:

```
strmqm linux
```

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

Kanalkonfiguration

Im folgenden Abschnitt wird die Konfiguration beschrieben, die auf dem Linux -Warteschlangenmanager ausgeführt werden muss, um den in [Abbildung 1](#) auf Seite 5 beschriebenen Kanal zu implementieren.

Es ist der MQSC-Befehl zum Erstellen der einzelnen Objekte angegeben. Starten Sie entweder **runmqsc** von einer UNIX -Eingabeaufforderung und geben Sie nacheinander jeden Befehl ein oder erstellen Sie die Befehle in einer Befehlsdatei.

Es werden Beispiele für die Verbindung von WebSphere MQ für Linux und WebSphere MQ für HP-UX angegeben. Um eine Verbindung zu WebSphere MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle denen im Beispiel für HP-UX ein.

Anmerkung: Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von WebSphere 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 WebSphere MQ for Linux</i>				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
Definition für lokalen Knoten				
A	Name des Warteschlangenmanagers		LINUX	
B	Name der lokalen Warteschlange		LINUX.LOCALQ	
Verbindung zu WebSphere MQ for Windows				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 1 auf Seite 13 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	WINNT	
D	Name der fernen Warteschlange		WINNT.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	WINNT.LOCALQ	
F	Name der Übertragungswarteschlange		WINNT	
G	Name des Senderkanals (SNA)		LINUX.WINNT.SNA	
H	Name des Senderkanals (TCP/IP)		LINUX.WINNT.TCP	
I	Name des Empfängerkanals (SNA)	G	WINNT.LINUX.SNA	
J	Name des Empfängerkanals (TCP)	H	WINNT.LINUX.TCP	
Verbindung zu WebSphere MQ for AIX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 2 auf Seite 20 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	AIX	
D	Name der fernen Warteschlange		AIXREMOTEQ	
E	Name der Warteschlange auf fernem System	B	AIXLOCALQ	
F	Name der Übertragungswarteschlange		AIX	
G	Name des Senderkanals (SNA)		LINUX.AIX.SNA	
H	Name des Senderkanals (TCP)		LINUX.AIX.TCP	
I	Name des Empfängerkanals (SNA)	G	AIX.LINUX.SNA	
J	Name des Empfängerkanals (TCP)	H	AIX.LINUX.TCP	
Verbindung mit MQSeries MQ for Compaq Tru64 UNIX				
Die Werte in diesem Tabellenabschnitt müssen denen in Ihrem Compaq Tru64 UNIX-System entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	DECUX	
D	Name der fernen Warteschlange		DECUX.REMOTEQ	

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

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
E	Name der Warteschlange auf fernem System	B	DECUX.LOCALQ	
F	Name der Übertragungswarteschlange		DECUX	
H	Name des Senderkanals (TCP)		DECUX.LINUX.TCP	
J	Name des Empfängerkanals (TCP)	H	LINUX.DECUX.TCP	
Verbindung zu WebSphere MQ for HP-UX				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 3 auf Seite 27 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	HPUX	
D	Name der fernen Warteschlange		HPUX.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	HPUX.LOCALQ	
F	Name der Übertragungswarteschlange		HPUX	
G	Name des Senderkanals (SNA)		LINUX.HPUX.SNA	
H	Name des Senderkanals (TCP)		LINUX.HPUX.TCP	
I	Name des Empfängerkanals (SNA)	G	HPUX.LINUX.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	HPUX.LINUX.TCP	
Verbindung zu WebSphere MQ for Solaris				
Die Werte in diesem Abschnitt der Tabelle müssen den in Tabelle 4 auf Seite 33 angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	SOLARIS	
D	Name der fernen Warteschlange		SOLARIS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	SOLARIS.LOCALQ	
F	Name der Übertragungswarteschlange		GIS	
G	Name des Senderkanals (SNA)		LINUX.SOLARIS.SNA	
H	Name des Senderkanals (TCP/IP)		LINUX.SOLARIS.TCP	
I	Name des Empfängerkanals (SNA)	G	SOLARIS.LINUX.SNA	
J	Name des Empfängerkanals (TCP/IP)	H	SOLARIS.LINUX.TCP	
C	Name des fernen Warteschlangenmanagers	A	AS400	
D	Name der fernen Warteschlange		AS400.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	AS400.LOCALQ	

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

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
F	Name der Übertragungswarteschlange		AS400	
G	Name des Senderkanals (SNA)		LINUX.AS400.SNA	
H	Name des Senderkanals (TCP)		LINUX.AS400.TCP	
I	Name des Empfängerkanals (SNA)	G	AS400.LINUX.SNA	
Verbindung zu MVS				
C	Name des fernen Warteschlangenmanagers	A	MVS	
D	Name der fernen Warteschlange		MVS.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	MVS.LOCALQ	
F	Name der Übertragungswarteschlange		MVS	
G	Name des Senderkanals (SNA)		LINUX.MVS.SNA	
H	Name des Senderkanals (TCP)		LINUX.MVS.TCP	
I	Name des Empfängerkanals (SNA)	G	MVS.LINUX.SNA	
Verbindung zu MQSeries für VSE/ESA (nur WebSphere 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	VSE	
D	Name der fernen Warteschlange		VSE.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	VSE.LOCALQ	
F	Name der Übertragungswarteschlange		VSE	
G	Name des Senderkanals		LINUX.VSE.SNA	
I	Name des Empfängerkanals	G	VSE.LINUX.SNA	

Senderkanaldefinitionen über SNA in IBM WebSphere MQ for Linux (x86 -Plattform)

Mustercode:

```

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

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

def chl (LINUX.HPUX.SNA) chltype(sdr) +             G
  trptype(lu62) +
  conname('HPUXCPIC') +                             14
  xmitq(HPUX) +                                     F
  replace

```

Empfängerkanaldefinitionen über SNA in IBM WebSphere MQ for Linux (x86 -Plattform)

Mustercode:

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

Senderkanaldefinitionen über TCP in IBM WebSphere MQ for Linux

Mustercode:

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

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

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

Empfängerkanaldefinitionen unter Verwendung von TCP/IP in IBM WebSphere MQ for Linux

Mustercode:

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

Warteschlangennamen

In diesem Abschnitt erhalten Sie Informationen zu den Einschränkungen für Warteschlangennamen sowie Informationen zu reservierten Warteschlangennamen.

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. Sie können diese Warteschlangendefinitionen mit den Befehlen **ALTER** oder **DEFINE REPLACE** an Ihre Installation anpassen. Die folgenden Namen sind für IBM WebSphere 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

Name der Warteschlange	Beschreibung
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
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 WebSphere MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	Warteschlange für die Initialisierung (nicht unter z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange für die Übertragung von Repository-Änderungen zwischen Warteschlangenmanagern (nur unter AIX, HP-UX, Linux, IBM i, Solaris, Windows und z/OS)
SYSTEM.CLUSTER.HISTORY.QUEUE	Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange mit Informationen zum Repository (nur unter AIX, HP-UX, Linux, IBM i, Solaris, Windows und z/OS)
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Mit der Warteschlange werden einzelne Übertragungswarteschlangen für jeden Clustersenderkanal erstellt.
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle Ziele, die durch Clusterunterstützung verwaltet werden (nur unter AIX, HP-UX, Linux, IBM i, Solaris, Windows und z/OS)
SYSTEM.COMMAND.INPUT	Warteschlange, an die unter z/OS Befehlsnachrichten 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

Name der Warteschlange	Beschreibung
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen.
SYSTEM.INTERNAL.REPLY.QUEUE	IBM WebSphere 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
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 für IBM WebSphere 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 gruppeninternen Warteschlangenagenten zur Übertragung von Nachrichten zwischen Warteschlangenmanagern derselben 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 WebSphere MQ-interne Warteschlange für Auswahlbewertungen (nicht unter z/OS)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM WebSphere MQ-interne Warteschlange für Auswahlprüfungen (nicht unter z/OS)

Weitere Objektnamen

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 entsprechend Ihrer Installation anpassen. Die folgenden Namen sind für IBM WebSphere MQ definiert:

Objektname	Beschreibung
SYSTEM.ADMIN.SVRCONN	Serververbindungskanal für die ferne Verwaltung eines Warteschlangenmanagers
SYSTEM.AUTO.RECEIVER	Standardempfängerkanal für automatische Definition (nurWindows, UNIX and Linux -Systeme)
SYSTEM.AUTO.SVRCONN	Standardserververbindungskanal für automatische Definition (nurIBM i, Windows, UNIX and Linux -Systeme)
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	Standard-SNA-Listener (nurWindows)
SYSTEM.DEFAULT.LISTENER.NETBIOS	Standardlistener NetBIOS (nurWindows)
SYSTEM.DEFAULT.LISTENER.SPX	Standard-SPX-Listener (nurWindows)
SYSTEM.DEFAULT.LISTENER.TCP	Standard-TCP/IP-Listener (nurIBM i-, Windows-, UNIX and Linux -Systeme)
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.DEFAULT.PROCESS	Standarddefinition für Prozesse
SYSTEM.DEFAULT.SERVICE	Standardservice (nurIBM i-, Windows-und UNIX and Linux -Systeme)
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.

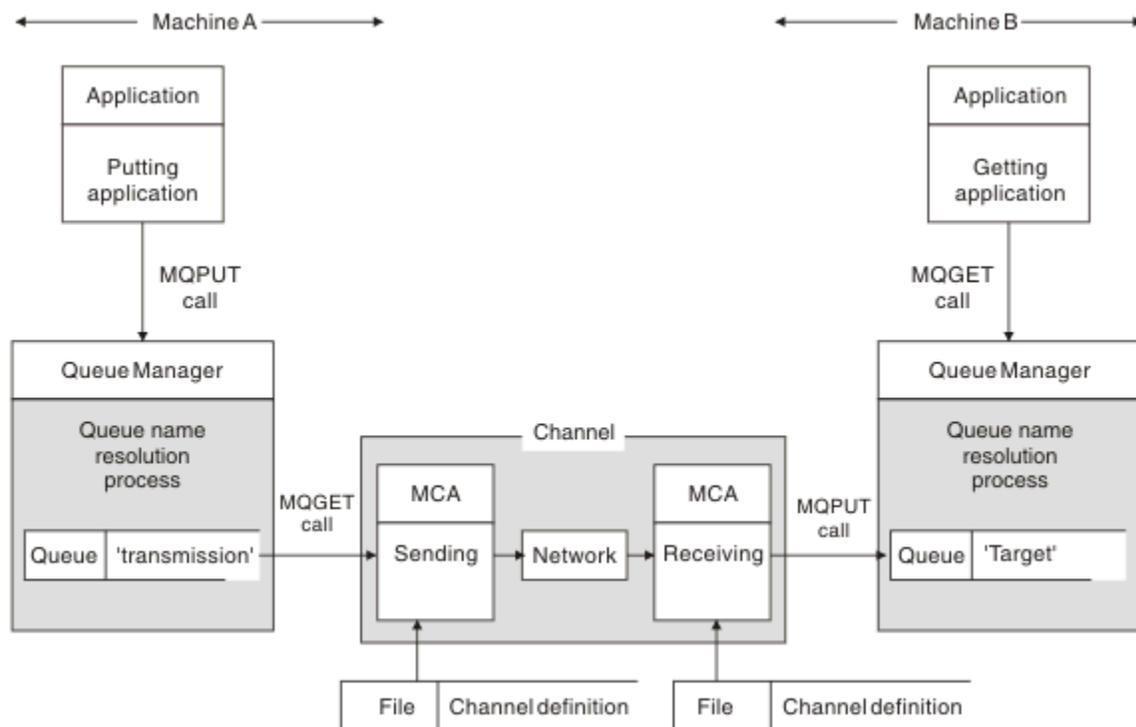


Abbildung 2. Namensauflösung

Wie in [Abbildung 2 auf Seite 47](#) 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

Listet die vom Befehl **crtmqm** erstellten System- und Standardobjekte auf.

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 WebSphere 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 6 auf Seite 49](#) enthält die System- und Standardwarteschlangenobjekte.
- In [Tabelle 7 auf Seite 51](#) sind die System- und Standardthemenobjekte aufgeführt.
- In [Tabelle 8 auf Seite 51](#) sind die System- und Standardkanalobjekte aufgeführt.
- In [Tabelle 9 auf Seite 52](#) sind die System- und Standardauthentifizierungsdatenobjekte aufgeführt.
- In [Tabelle 10 auf Seite 52](#) sind die System- und Standardempfangsprogrammobjekte aufgeführt.
- In [Tabelle 11 auf Seite 52](#) sind die System- und Standardnamenslistenobjekte aufgeführt.
- In [Tabelle 12 auf Seite 52](#) sind die System- und Standardprozessobjekte aufgeführt.
- In [Tabelle 13 auf Seite 52](#) sind die System- und Standardserviceobjekte aufgeführt.

Objektname	Beschreibung
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 6. 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	Kanalinitialisierungwarteschlange
SYSTEM.CHANNEL.SYNCQ	Warteschlange für die Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM WebSphere MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	Standardinitialisierungwarteschlange für CICS
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 6. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.MQEXPLORER.REPLY.MODEL	Die IBM WebSphere 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 WebSphere 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	Unterstützung verzögerter Nachrichten in JMS

Tabelle 7. 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 8. 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.

<i>Tabelle 9. System- und Standardobjekte: Authentifizierungsdatenobjekte</i>	
Objektname	Beschreibung
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 10. System- und Standardobjekte: Empfangsprogramme</i>	
Objektname	Beschreibung
SYSTEM.DEFAULT.LISTENER.TCP	TCP-Standardempfangsprogramm.
SYSTEM.DEFAULT.LISTENER.LU62 ¹	LU62-Standardempfangsprogramm.
SYSTEM.DEFAULT.LISTENER.NETBIOS ¹	NETBIOS-Standardempfangsprogramm.
SYSTEM.DEFAULT.LISTENER.SPX ¹	SPX-Standardempfangsprogramm.

1. Nur Windows

<i>Tabelle 11. System- und Standardobjekte: Namenslisten</i>	
Objektname	Beschreibung
SYSTEM.DEFAULT.NAMELIST	Standardnamensliste.

<i>Tabelle 12. System- und Standardobjekte: Prozesse</i>	
Objektname	Beschreibung
SYSTEM.DEFAULT.PROCESS	Standardprozessdefinition.

<i>Tabelle 13. System- und Standardobjekte: Services</i>	
Objektname	Beschreibung
SYSTEM.DEFAULT.SERVICE	Standardservice.
SYSTEM.BROKER	Publish/Subscribe-Broker

Objekte der Windows-Standardkonfiguration

Auf Windows-Systemen können Sie mithilfe der Anwendung WebSphere 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 WebSphere MQ-Objektnamen nicht gültig sind (beispielsweise Bindestriche) werden durch ein Unterstreichungszeichen ersetzt.

Gültige Zeichen für WebSphere 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 14 auf Seite 53](#) aufgeführten Objekte erstellt.

<i>Tabelle 14. Von der Anwendung für die Windows-Standardkonfiguration erstellte Objekte</i>	
Objekt	Name
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 standardmäßigen WebSphere MQ-Port, der die Portnummer 1414 hat. 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. Nachdem WebSphere 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 WebSphere MQ Postcard	<p>Die lokale Nachrichtenwarteschlange zur Verwendung durch die Anwendung WebSphere 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 14 auf Seite 53](#) 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 15. Standardwerte von SYSTEM.BASE.TOPIC</i>	
Parameter	Wert
TOPICSTR	"
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
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

Falls dieses Objekt nicht vorhanden ist, werden seine Standardwerte von IBM WebSphere 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 subscribieren.

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.

Informationen zu Zeilengruppen

Anhand der folgenden Informationen können Sie die Angaben innerhalb von Zeilengruppen konfigurieren. Zudem finden Sie hier eine Auflistung des Inhalts der Dateien `mqs.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 WebSphere 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:
 - Zeilengruppe *AccessMode* (nur unter Windows)
 - Zeilengruppe *Service* - für installierbare Services
 - Zeilengruppe *Log*
 - 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*
 - 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 57 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_s.ini](#)
- Datei [qm.ini](#)
- Datei [mqclient.ini](#)

Dort finden Sie eine Liste der möglichen Zeilengruppen in den einzelnen Konfigurationsdateien.

mq_s.ini-Datei

Ein Beispiel für eine Datei `mqs.ini` finden Sie im Abschnitt [Beispiel einer IBM WebSphere MQ-Konfigurationsdatei für UNIX and Linux-Systeme](#).

Eine Datei `mqs.ini` 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 WebSphere MQ für Systeme mit UNIX and Linux](#) finden Sie ein Beispiel für die Datei `qm.ini`.

Eine Datei `qm.ini` kann die folgenden Zeilengruppen enthalten:

- [ExitPath](#)
- [Protokoll](#)
- [QMErrorLog](#)
- [QueueManager](#)
- [Sicherheit](#)
- [Service](#) und [ServiceComponent](#)

So konfigurieren Sie [InstallableServices](#):

- Verwenden Sie auf UNIX and Linux-Plattformen die Zeilengruppen [Service](#) und [ServiceComponent](#).
- Unter Windows erfolgt die Konfiguration über die Bearbeitung der Registrierung (Befehl **regedit**).

- [Connection](#) für [DefaultBindType](#)



Achtung: Falls Sie eine [Connection](#)-Zeilengruppe benötigen, müssen Sie diese erstellen.

- [SSL und TLS](#)
- [TCP, LU62 und NETBIOS](#)
- [XAResourceManager](#)

Darüber hinaus können Sie mit dem Befehl [crtmqm](#) die folgenden Eigenschaften ändern:

- [AccessMode](#) (nur unter Windows)
- [RestrictedMode](#) (nur auf Systemen mit UNIX and Linux)

Datei 'mqclient.ini'

Eine Datei `mqclient.ini` kann die folgenden Zeilengruppen enthalten:

- [CHANNELS](#)
- [ClientExitPath](#)
- [LU62, NETBIOS und SPX](#)
- [MessageBuffer](#)
- [SSL](#)
- [TCP](#)

Darüber hinaus ist zur Konfiguration eines [PreConnect](#)-Exits möglicherweise die Zeilengruppe [PreConnect](#) erforderlich.

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 gilt für die Konfigurationsdatei des Warteschlangenmanagers für IBM WebSphere MQ auf Windows- und UNIX and Linux -Systemen. Der Name der Datei lautet auf allen Plattformen 'qm.ini'.

Zeilengruppen für die verteilte Steuerung von Warteschlangen sind:

- Kanäle
- TCP
- LU62
- NETBIOS

- SPX (nur Windows XP und Windows 2003 Server)
- EXITPATH

In [Abbildung 3 auf Seite 58](#) 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".
  ADOPTNEWMCA=ch1type  ; 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.
  ADOPTNEWMCACHHECK=    ; Specifies the type checking required.
  typecheck              ; The default is "NAME", "ADDRESS", and "QM".
TCP:
  PORT=n                 ; Port number, the default is 1414
  KEEPALIVE=Yes          ; Switch TCP/IP KeepAlive on
  LIBRARY2=DLLName2    ; Used if code is in two libraries
  EXITPATH:2           ; Location of user exits (MQSeries for AIX,
                        ; HP-UX, and Solaris only)
  EXITPATHS=             ; String of directory paths.
```

Abbildung 3. Zeilengruppen der Datei *qm.ini* für die verteilte Steuerung von Warteschlangen

Anmerkung:

1. MQIBINDTYPE gilt nur für IBM WebSphere MQ für AIX, IBM WebSphere MQ für HP-UX und IBM WebSphere MQ für Solaris.
2. EXITPATH gilt nur für IBM WebSphere MQ für AIX, IBM WebSphere MQ für HP-UX und IBM WebSphere MQ für Solaris.

Zugehörige Informationen

[Konfiguration](#)

[Konfigurationsdaten auf Windows-, UNIX- und Linux -Systemen ändern](#)

Kanalattribute

In diesem Abschnitt werden die in den Kanaldefinitionen angegebenen Kanalattribute beschrieben.

Bei diesen Informationen handelt es sich um produktabhängige Informationen zur Programmierschnittstelle.

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.

Anmerkung: In WebSphere 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 WebSphere MQ-Kanalattribute werden im Abschnitt [Tabelle 16 auf Seite 59](#) aufgelistet.

Tabelle 16. Kanalattribute für die Kanaltypen

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
Datum ändern	ALTDATE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Zeit ändern	ALTTIME	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Intervall der Überwachungssignale für Stapel	BATCHHB	Ja	Ja					Ja	Ja
Stapelintervall	BATCHINT	Ja	Ja					Ja	Ja
Stapelgröße	BATCHSZ	Ja	Ja	Ja	Ja			Ja	Ja
Channel Name	CHANNEL	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Kanalstatistik	STATCHL	Ja	Ja	Ja	Ja			Ja	Ja
Channel Type	CHLTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Clientkanalgewichtung	CLNTWGHT					Ja			
Cluster	CLUSTER							Ja	Ja
Clusternamensliste	CLUSNL							Ja	Ja
Clusterauslastungspriorität	CLWLPRTY							Ja	Ja
Clusterauslastungsrangordnung	CLWLRANK							Ja	Ja
Clusterauslastungsgewichtung	CLWLWGHT							Ja	Ja
Verbindungsaffinität	AFFINITY					Ja			
Verbindungsname	CONNNAME	Ja	Ja		Ja	Ja		Ja	Ja
Nachricht umwandeln	CONVERT	Ja	Ja					Ja	Ja
Datenkomprimierung	COMPMSG	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Beschreibung	DESCR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Unterbrechungsintervall	DISCINT	Ja	Ja					Ja	Ja
Header-Komprimierung	COMPHDR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Intervall der Überwachungssignale	HBINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Keepalive-Intervall	KAINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Lokale Adresse	LOCLADDR	Ja	Ja		Ja	Ja		Ja	Ja

Tabelle 16. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
Zähler für Wiederholungsversuche nach langem Intervall	LONGRTY	Ja	Ja					Ja	Ja
Langes Wiederholungsintervall	LONGTMR	Ja	Ja					Ja	Ja
Gibt den LU 6.2-Modusnamen an.	MODENAME	Ja	Ja		Ja	Ja		Ja	Ja
Gibt das LU 6.2-Transaktionsprogramm an.	TPNAME	Ja	Ja		Ja	Ja		Ja	Ja
Maximale Instanzen	MAXINST						Ja		
Maximale Instanzen pro Client	MAXINSTC						Ja		
Maximale Nachrichtenlänge	MAXMSG	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Name des Nachrichtenkanalagenten	MCANAME	Ja	Ja		Ja			Ja	Ja
Typ des Nachrichtenkanalagenten	MCATYPE	Ja	Ja		Ja			Ja	Ja
Benutzer des Nachrichtenkanalagenten	MCAUSER	Ja	Ja	Ja	Ja		Ja	Ja	Ja
Name des Nachrichtenexits	MSGEXIT	Ja	Ja	Ja	Ja			Ja	Ja
Benutzerdaten des Nachrichtenexits	MSGDATA	Ja	Ja	Ja	Ja			Ja	Ja
Name des Exits für Nachrichtenwiederholung	MREXIT			Ja	Ja				Ja
Benutzerdaten des Exits für Nachrichtenwiederholungen	MRDATA			Ja	Ja				Ja
Zähler für Nachrichtenwiederholungen	MRRTY			Ja	Ja				Ja
Intervall für Nachrichtenwiederholungen	MRTMR			Ja	Ja				Ja
Monitoring	MONCHL	Ja	Ja	Ja	Ja		Ja	Ja	Ja
Netzverbindungspriorität	NETPRTY								Ja
Gibt die Geschwindigkeit nicht permanenter Nachrichten an.	NPMSPEED	Ja	Ja	Ja	Ja			Ja	Ja
Kennwort	KENNWORT	Ja	Ja		Ja	Ja		Ja	

Tabelle 16. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Be- fehlsparame- ter	SDR	SVR	RCVR	RQSTR	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR
Eigenschaftensteue- rung	PROPCTL	Ja	Ja					Ja	Ja
PUT-Berechtigung	PUTAUT			Ja	Ja				Ja
Warteschlangenmana- gername	QMNAME					Ja			
Name des Empfangs- exits	RCVEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Benutzerdaten des Empfangsexits	RCVDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Name des Sicherheits- exits	SCYEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Benutzerdaten des Si- cherheitsexits	SCYDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Name des Sendeexits	SENDEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Benutzerdaten des Sendeexits	SENDDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Folgenummernserie	SEQWRAP	Ja	Ja	Ja	Ja			Ja	Ja
Gemeinsam genutzte Verbindungen	SHARECNV					Ja	Ja		
Zähler für Wiederho- lungsversuche nach kurzem Intervall	SHORTRTY	Ja	Ja					Ja	Ja
Kurzes Wiederholungs- intervall	SHORTTMR	Ja	Ja					Ja	Ja
SSL-Verschlüsselungs- spezifikation	SSLCIPH	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
SSL-Clientauthentifi- zierung	SSLCAUTH		Ja	Ja	Ja		Ja		Ja
SSL-Peer	SSLPEER	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Name der Übertra- gungswarteschlange	XMITQ	Ja	Ja						
Transport Type	TRPTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Warteschlange für nicht zustellbare Nach- richten verwenden	USEDLQ	Ja	Ja	Ja	Ja			Ja	Ja
Benutzer-ID	USERID	Ja	Ja		Ja	Ja		Ja	

Zugehörige Konzepte

„Kanalattribute in alphabetischer Reihenfolge“ auf Seite 62

In diesem Abschnitt werden alle Attribute eines Kanalobjekts mit ihren gültigen Werten und ggf. Hinweisen zu ihrer Verwendung beschrieben.

Zugehörige Verweise

[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.

WebSphere 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 (ALTDAT)

Dieses Attribut ist das Datum, an dem die Definition zuletzt geändert wurde, im Format yyyy-mm-dd.

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.

Stapelintervall der Überwachungssignale (BATCHEB)

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 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 (BATCHEI)

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 geschlossen, sobald die in BATCHEI angegebene Anzahl von Nachrichten gesendet wurde oder die Übertragungswarteschlange leer ist. Bei Kanälen mit ge-

ringer 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.
- 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

Stapelgröße (BATCHSZ)

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 wird für Produkte der Version 1 mit Ausnahme von WebSphere MQ for MVS verwendet.
- 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

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:

Buchstaben	(A-Z, a-z; Groß-/Kleinschreibung wird beachtet)
Ziffern	(0-9)
Zeitraum	(.)
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)

Dieses Attribut steuert 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](#).

Dieses Attribut wird unter z/OS nicht unterstützt.

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:

Nachrichtkanaltypen:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

MQI-Kanaltypen:

- Clientverbindung (nur WebSphere MQ für Windows -Systeme und UNIX -Systeme)

Anmerkung: Sie können Clientverbindungskanäle unter z/OS auch für die Verwendung auf anderen Plattformen definieren.

- Serververbindung

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

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.

Gemäß den Vorgaben für die Benennung von WebSphere MQ-Objekten beträgt die maximale Länge 48 Zeichen.

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

Clusterauslastungspriorität (CLWLPRTY)

Dieses Attribut gibt die Priorität des Kanals an.

Der Wert muss im Bereich 0 bis 9 liegen, wobei 0 die niedrigste Priorität und 9 die höchste Priorität ist.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

Clusterauslastungsrangordnung (CLWLRANK)

Dieses Attribut gibt den Rang des Kanals an.

Der Wert muss im Bereich 0 bis 9 liegen, wobei 0 der niedrigste Rang und 9 der höchste Rang ist.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

Clusterauslastungsgewichtung (CLWLWGHT)

Dieses Attribut weist dem Kanal einen Gewichtungsfaktor zu, sodass der Anteil der Nachrichten, der über diesen Kanal gesendet wird, gesteuert werden kann.

Der Wert muss im Bereich 1 und 99 liegen, wobei 1 die geringste Gewichtung und 99 die höchste Gewichtung ist.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

Verbindungsaffinität (AFFINITY)

Dieses Attribut gibt an, ob Clientanwendungen, bei denen mehrfach Verbindungen mit dem gleichen Warteschlangenmanagernamen 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.

Für Clientanwendungen, die in C, C++ oder dem .NET Programming Framework (einschließlich vollständig verwaltetem .NET) geschrieben sind, und für Anwendungen, die die IBM WebSphere MQ -Klassen für Java und IBM WebSphere MQ -Klassen für JMS verwenden, wird die Liste aktualisiert, wenn die 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.

Für Clientanwendungen, die in C, C++ oder dem .NET Programming Framework (einschließlich vollständig verwaltetem .NET) geschrieben sind, und für Anwendungen, die die IBM WebSphere MQ

-Klassen für Java und IBM WebSphere MQ -Klassen für JMS verwenden, wird die Liste aktualisiert, wenn die CCDT seit der Erstellung der Liste geändert wurde.

Dieses Attribut ist nur für den Kanaltyp Clientverbindungskanal gültig.

Verbindungsname (CONNNAME)

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 CONNNAME 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 CLNTWGHT-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 Version 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 lassen sich Probleme vermeiden, die bei der Herstellung einer Kanalverbindung über einen Client oder Warteschlangenmanager auftreten können, der mit einer Version vor IBM WebSphere MQ Version 7.0.1 ausgeführt wird.

Auf AIX, HP-UX, IBM i, Linux, Solaris, and Windows-Plattformen ist der Parameter für den TCP/IP-Verbindungsnamen eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM WebSphere 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 immer in Schreibweise mit Trennzeichen (IPv4) oder in hexadezimaler Form (IPv6) angegeben, nicht als alphanumerischer DNS-Hostname.

Der Name ist bei z/OS bis zu 48 Zeichen (siehe Hinweis 1) und bei anderen Plattformen bis zu 264 Zeichen lang. Darüber hinaus gilt Folgendes:

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)). Er kann den IP-Namen (IP_name) einer dynamischen DNS-Gruppe in z/OS 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 finden Sie im Abschnitt „Lokale Adresse (LOCLADDR)“ auf Seite 74.

Transporttyp LU 6.2

Bei WebSphere MQ for IBM i, Windows-Systemen und UNIX-Systemen verwenden Sie den vollständig qualifizierten Namen der Partner-LU, wenn TPNAME und MODENAME angegeben sind. Bei anderen

Versionen oder wenn TPNAME und MODENAME leer sind, geben Sie den Namen des CPI-C-Nebeninformationsobjekts für die bestimmte Plattform an.

Unter z/OS kann dieser Wert in zwei Formaten 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.

Anmerkung: 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.

Anmerkung: Bei Clusterempfängerkanälen befinden sich die Nebeninformationen auf den anderen Warteschlangenmanagern innerhalb des Clusters. In diesem Fall kann es sich alternativ auch um einen Namen handeln, den ein Exit für die automatische Kanaldefinition in die entsprechenden LU-Informationen für den lokalen Warteschlangenmanager auflösen kann.

Bei der explizit bzw. implizit angegebenen LU kann es sich um eine generische VTAM-Ressourcen-Gruppe handeln.

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 Socketnummer übergangen, wird die SPX-Standardsocketnummer von WebSphere 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. Die Begrenzung auf 48 Zeichen kann wie folgt umgangen werden:
 - 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 [90](#).

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, die die Datenumwandlung nicht unterstützt, geben Sie mit diesem Kanalattribut an, dass die Nachricht **vor** der Übertragung in das vom empfangenden System benötigte Format umgewandelt werden muss.

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 [72](#).

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.

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.

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 Serververbindungen gilt zwischen WebSphere 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 wird nur unter z/OS unterstützt.

Folgende Werte sind möglich:

QMGR

Der Kanal wird in der Seitengruppe des Warteschlangenmanagers definiert, der den Befehl ausführt. Dieser Wert stellt den Standardwert dar.

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 WebSphere MQ classes for Java-, JMS- oder .NET-APIs verwenden, wird der HBINT-Wert auf eine der folgenden Arten ermittelt:

- 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 Version 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 Verweise

[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. Auf anderen Plattformen als z/OS 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 73.

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 anderen Plattformen wird die TCP/IP-Keepalive-Funktion aktiviert, indem der Parameter KEEPALIVE in der TCP-Zeilengruppe der Konfigurationsdatei für verteilte Warteschlangensteuerung (qm.ini) auf YES gesetzt wird (KEEPALIVE=YES), oder sie wird über den IBM WebSphere 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):

<i>Tabelle 17. Vereinbarter HBINT-Wert und entsprechender KAINTE-Wert</i>	
Vereinbarter HBINT-Wert	KAINTE
>0	Vereinbarter HBINT-Wert + 60 Sekunden
0	0

Wenn für den KAINTE-Wert die Option AUTO angegeben ist und es sich um einen Serververbindungskanal handelt, wird anstelle des Keepalive-Intervalls der Wert TCP INTERVAL verwendet.

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.

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 einen Kanal auch zwingen, einen IPv4- oder IPv6-Stack in einem Dual Stack-System oder einen Dualmodus-Stack in einem Einzelstack-System zu verwenden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Wenn LOCLADDR eine Netzadresse enthält, muss diese zu einer Netzschnittstelle 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 Netzschnittstellen 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, beträgt MQ_LOCAL_ADDRESS_LENGTH.

Wenn Sie LOCLADDR nicht angeben, 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-Schreibweise mit Trennzeichen

Beispiel: 192.0.2.1

IPv6-Hexadezimalschreibweise

Beispiel: 2001:DB8:0:0:0:0:0:0

Alphanumerisches Hostnamensformat

Beispiel: WWW.EXAMPLE.COM

low-port and 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 18. Beispiele für die Verwendung des Parameters LOCLADDR</i>	
LOCLADDR	Bedeutet
9.20.4.98	Kanal wird lokal an diese Adresse gebunden.
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 (CONNNAME) 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 Sie eine lokale Adresse (LOCLADDR) als IPv4-Netzadresse angeben, wird ein Fehler generiert und der Start des Kanals schlägt fehl. Wenn Sie auf Plattformen, die die IPv6-Adresszuordnung unterstützen, einen Verbindungsnamen (CONNAME) als IPv4-Netzadresse angeben, wird die Adresse einer IPv6-Adresse zugeordnet. Beispiel: xxx.xxx.xxx.xxxist : :ffff:xxx.xxx.xxx.xxxzugeordnet. 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 die lokale Adresse (LOCLADDR) als Hostname angegeben wird, der in IPv4 -und IPv6 -Adressen aufgelöst wird, bestimmt der Verbindungsname (CONNAME), welcher der Stacks verwendet wird. Wenn Sie die lokale Adresse (LOCLADDR) und den Verbindungsnamen (CONNAME) als Hostnamen angeben, die in IPv4- und IPv6-Adressen aufgelöst werden, bestimmt das Warteschlangenmanagerattribut IPADDRV den verwendeten Stack.
- Wenn die IPv4- und IPv6-Stacks für das System konfiguriert sind und keine lokale Adresse (LOCLADDR) für einen Kanal angegeben wird, bestimmt der für den Kanal angegebene Verbindungsname (CONNAME), welcher IP-Stack verwendet wird. Wenn Sie den Verbindungsnamen (CONNAME) als Hostname angeben, der in IPv4- und IPv6-Adressen aufgelöst wird, bestimmt das Warteschlangenmanagerattribut IPADDRV den verwendeten Stack.

distributed Auf verteilten Plattformen kann ein Standardwert als lokale Adresse gesetzt werden, der für alle Senderkanäle übernommen 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 dieses Werts entspricht dem des MQSC-Attributs LOCLADDR.

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 potenziellen 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.

Zugehörige Konzepte

[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.

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).

(Es erfolgt kein Wiederholungsversuch, wenn die Ursache des Fehlschlagens darauf schließen lässt, dass eine Wiederholung wahrscheinlich nicht erfolgreich sein wird).

Wenn der Kanalinitiator (unter z/OS) bzw. der Kanal (auf verteilten Plattformen) 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 verteilten Plattformen) 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.

Anmerkung: Für IBM i-, UNIX und Windows-Systeme:

1. Wenn der Status des Kanals von RETRYING zu RUNNING wechselt, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht unverzüglich zurückgesetzt. Die Zähler werden erst zurückgesetzt, nachdem die erste Nachricht erfolgreich über den Kanal versendet wurde, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., nachdem 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.

Das Attribut *Zähler für lange Wiederholungsversuche* kann auf einen Wert zwischen null und 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Anmerkung: Auf UNIX-Systemen sowie Windows-Systemen erfolgt ein Wiederholungsversuch nur, wenn ein Kanalinitiator ausgeführt wird. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

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, *long retry count* Mal in diesem langen Intervall eine Verbindung herzustellen, nachdem er *short retry count* Mal im kurzen Wiederholungsintervall versucht hat.

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 Weise festgelegt, weitere Informationen zum Einrichten der Kommunikation für Ihre Plattform finden Sie im Abschnitt [Anwendungen mithilfe der verteilten Steuerung von Warteschlangen verbinden](#).

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 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.

Die Clientanschlusseinrichtung CAF (Client Attachment Feature) ist eine Option von WebSphere MQ for z/OS, die den Anschluss von Clients unter z/OS unterstützt. Wenn CAF nicht installiert ist, kann das Attribut für den Kanal SYSTEM.ADMIN.SVRCONN nur auf 0 bis fünf gesetzt werden. Bei nicht installiertem CAF wird ein Wert größer als fünf als 0 interpretiert.

Wenn der Wert kleiner als die Anzahl der Instanzen des Serververbindungskanals, die aktuell ausgeführt werden, oder sogar 0 ist, hat dies keine Auswirkung auf die aktiven Kanäle. Neue Instanzen können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

Dieses Attribut ist nur für Serververbindungskanäle gültig.

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.

Die Clientanschlusseinrichtung CAF (Client Attachment Feature) ist eine Option von WebSphere MQ for z/OS, die den Anschluss von Clients unter z/OS unterstützt. Wenn CAF nicht installiert ist, kann das Attribut für den Kanal SYSTEM.ADMIN.SVRCONN nur auf 0 bis fünf gesetzt werden. Bei nicht installiertem CAF wird ein Wert größer als fünf als 0 interpretiert.

Wenn der Wert kleiner als die Anzahl der Instanzen des Serververbindungskanals, die aktuell auf einzelnen Clients ausgeführt werden, oder sogar 0 ist, hat dies keine Auswirkung 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.

Maximale Nachrichtenlänge (MAXMSGL)

Dieses Attribut gibt die maximale Länge von Nachrichten an, die über den Kanal übertragen werden können.

Geben Sie auf WebSphere MQ für UNIX und Windows -Systemen einen Wert größer-gleich null und kleiner-gleich der maximalen Nachrichtenlänge für den Warteschlangenmanager an. Weitere Informationen finden Sie im Abschnitt zum MAXMSGL-Parameter des Befehls ALTER QMGR in [ALTER QMGR](#).

Da es auf WebSphere 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.

Durch Hinzufügen der digitalen Signatur und des Schlüssels zur Nachricht erhöht [WebSphere MQ 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 WebSphere MQ for z/OS wird dieses Attribut nur für den Kanaltyp 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 RUNMSLSR als Thread ausgeführt. Bei Eingabe von **inetd** wird er als Prozess ausgeführt.

Unter WebSphere 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 WebSphere 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.

Unter IBM WebSphere 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 Konzepte

Kanalauthentifizierungsdatensätze

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 85.

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 hierzu „Benutzerdaten des Empfangsexits (RCVDATA)“ auf Seite 86.

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 85. 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 für diesen Kanal wird ausgeschaltet.

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 unter [Warteschlangen- und Kanalüberwachungsdaten anzeigen](#).

Netzverbindungspriorität (NETPRTY)

Dieses Attribut gibt die Priorität für die Netzverbindung an.

Bei der verteilten Steuerung von Warteschlangen wird, sofern mehrere Pfade verfügbar sind, der Pfad mit der höchsten Priorität gewählt. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste Priorität bezeichnet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clusterempfänger

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](#).

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 WebSphere 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

PUT-Berechtigung (PUTAUT)

Dieses Attribut gibt den Typ der vom MCA auszuführenden Sicherheitsverarbeitung an.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- 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).

Sie können eine der folgenden Optionen wählen:

Prozesssicherheit, auch als Standardberechtigung (DEF) bezeichnet

Die Standard-Benutzer-ID wird verwendet.

Auf allen Plattformen wird die Berechtigung zum Öffnen für die Warteschlange unter Verwendung der Benutzer-ID des Prozesses oder Benutzers überprüft, unter der der MCA auf der Empfangsseite des Nachrichtenkanals ausgeführt wird.

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 wird die Berechtigung zum Öffnen für die Warteschlange für MQOO_SET_ALL_CONTEXT und MQOO_ALTERNATE_USER_AUTHORITY unter Verwendung der Benut-

zer-ID des Prozesses oder Benutzers überprüft, unter der der MCA auf der Empfangsseite des Nachrichtenkanals ausgeführt wird. Die Benutzer-ID, die zum Überprüfen der Öffnungsberechtigung für die Warteschlange für MQOO_OUTPUT verwendet wird, ist die *UserIdentifier* im Nachrichtendes- kriptor.

Kontextsicherheit (CTX) wird für Serververbindungskanäle nicht unterstützt.

Weitere Details zu Kontextfeldern und Öffnungsoptionen finden Sie im Abschnitt [Kontextinformationen steuern](#).

Weitere Informationen zu Sicherheit finden Sie unter:

- [Sicherheit](#)
- [Sicherheit auf Windows, UNIX and Linux-Systemen](#) für WebSphere MQ-UNIX-Systeme und Windows-Systeme einrichten

Name des Warteschlangenmanagers (QMNAME)

Dieses Attribut gibt den Namen des Warteschlangenmanagers bzw. der Warteschlangenmanagergruppe an, zu dem bzw. der eine WebSphere MQ-Client-Anwendung eine Verbindung anfordern 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.
- Unter IBM i wird folgendes Format verwendet:

```
libname/progname
```

wenn es in CL-Befehlen verwendet wird.

Bei Angabe in WebSphere 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.

- Unter Windows hat das Attribut folgendes Format:

```
dllname(functionname)
```

Dabei wird *dllname* ohne das Suffix `.DLL` angegeben. Die maximale Länge der Zeichenfolge beträgt 40 Zeichen.

- In UNIX-Systemen hat es das Format:

```
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 Nachrichtensexitprogrammen 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 WebSphere MQ für IBM i können Sie bis zu 10 Exitnamen auflisten. In WebSphere MQ für z/OS ist die Angabe von bis zu acht Exitnamen möglich.

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)
```

In WebSphere MQ für UNIX-Systemen und Windows-Systemen ist die Länge von Zeichenfolgen der Exitnamen und Zeichenfolgen von Benutzerdaten auf 500 Zeichen beschränkt. In WebSphere MQ für IBM i können Sie bis zu 10 Exitnamen angeben, und die Länge der Benutzerdaten ist jeweils auf 32 Zeichen beschränkt. In WebSphere MQ für z/OS können Sie bis zu acht Zeichenfolgen von Benutzerdaten mit jeweils 32 Zeichen angeben.

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 85. 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 85.

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 hierzu „Benutzerdaten des Empfangsexits (RCVDATA)“ auf Seite 86.

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.

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. Ist auch dies erfolglos, beendet der Kanal seine Zuordnungsversuche.

(Es erfolgt kein Wiederholungsversuch, wenn die Ursache des Fehlschlagens darauf schließen lässt, dass eine Wiederholung wahrscheinlich nicht erfolgreich sein wird).

Wenn der Kanalinitiator (unter z/OS) bzw. der Kanal (auf verteilten Plattformen) 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 verteilten Plattformen) 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.

Anmerkung: Für UNIX-Systeme und Windows-Systeme:

1. Wenn der Status des Kanals von RETRYING zu RUNNING wechselt, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht unverzüglich zurückgesetzt. Die Zähler werden erst zurückgesetzt, nachdem die erste Nachricht erfolgreich über den Kanal versendet wurde, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., nachdem 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.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Anmerkung: In UNIX- und Windows-Systemen muss ein Kanalinitiator ausgeführt werden, damit Wiederholungsversuche erfolgen. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

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- oder SSL-Verbindung an.

Jede WebSphere 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 weder SSL noch TLS 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 dagegen einen nicht leeren Wert aufweist, wird vom Kanal versucht, die angegebene Verschlüsselung für die Nutzung von SSL oder TLS zu verwenden. Auch in diesem Fall müssen beide Enden des Kanals denselben SSLCIPH-Wert aufweisen.

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 SSL-Zertifikat von einem SSL-Client empfangen und authentifizieren muss.

Mögliche Werte:

OPTIONAL

Wenn der SSL-Peer-Client ein Zertifikat sendet, wird das Zertifikat normal verarbeitet, die Authentifizierung schlägt aber nicht fehl, wenn kein Zertifikat gesendet wird.

ERFORDERLICH

Wenn der SSL-Client kein Zertifikat sendet, schlägt die Authentifizierung fehl.

Der Standardwert ist REQUIRED.

Auch in einer Nicht-SSL-Kanaldefinition, in der SSLCIPH fehlt oder leer ist, kann für SSLCAUTH ein Wert eingegeben werden.

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 unter [WebSphere MQ-Scriptbefehlsreferenz \(MQSC\) 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 WebSphere MQ-Kanals aus verwendet.

Anmerkung: Alternativ dazu können zur Beschränkung von Verbindungen zu Kanälen durch Abgleich des definierten SSL- oder TLS-Namens Kanalauthentifizierungsdatensätze verwendet werden. Über die Authentifizierungsdatensätze für Kanäle können verschiedene Muster für definierte Namen des Zertifikatinhabers in SSL bzw. 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.

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.

Auch in einer Nicht-SSL-Kanaldefinition, in der SSLCIPH fehlt oder leer ist, kann für SSLPEER ein Wert eingegeben werden. In diesem Fall können Sie SSL vorübergehend für Debugging-Zwecke inaktivieren, ohne die SSL-Parameter entfernen und später wieder einsetzen zu müssen.

Weitere Informationen zur Verwendung von SSLPEER finden Sie unter [WebSphere MQ-Scriptbefehlsreferenz \(MQSC\)](#) und [Sicherheit](#).

Dieses Attribut ist für alle Kanaltypen gültig.

Zugehörige Konzepte

[Kanalauthentifizierungsdatensätze](#)

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 90)
SPX	SPX („1“ auf Seite 90)
Anmerkungen: 1. Zur Verwendung unter Windows. Kann auch unter z/OS zur Definition von Clientverbindungskanälen zur Verwendung unter Windows eingesetzt werden.	

Dieses Attribut gilt für alle Kanaltypen, wird jedoch von antwortenden Nachrichtenkanalagenten ignoriert.

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:

Nein

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 Warteschlangenmanager-Attribut DEADQ den Namen für eine Warteschlange für nicht zustellbare Nachrichten vorgibt, wird diese verwendet, andernfalls erfolgt dieselbe Aktion wie bei der Angabe von 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 WebSphere 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 WebSphere 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 WebSphere MQ-Clusterbefehle

Die zu den IBM WebSphere 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:

- Unter IBM WebSphere MQ for IBM 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. Verwenden Sie zum Erstellen einer Clusternamensliste den Befehl **DEFINE NAMELIST**.

IBM WebSphere MQ Explorer

Unter IBM WebSphere MQ for z/OS Version 6 oder höher kann ein Cluster mit Repository-Warteschlangenmanagern in der grafischen Benutzerschnittstelle des Explorers verwaltet werden. Die Benennung

eines weiteren Repositorys auf einem separaten System ist nicht erforderlich. In frühere Versionen von WebSphere MQ for z/OS kann der IBM WebSphere MQ Explorer keinen Cluster mit Repository-Warteschlangenmanagern verwalten. In diesem Fall müssen Sie daher ein weiteres Repository auf einem System benennen, das von IBM WebSphere MQ Explorer verwaltet werden kann.

Unter IBM WebSphere MQ für Windows und WebSphere MQ für Linux können Sie auch IBM WebSphere MQ Explorer verwenden, um mit Clustern zu arbeiten. Ebenso können Sie den eigenständigen IBM WebSphere MQ Explorer-Client verwenden.

In IBM WebSphere MQ Explorer können Sie Clusterwarteschlangen anzeigen und den Status von Cluster-sender- und Clusterempfängerkanälen abfragen. IBM WebSphere MQ Explorer stellt zwei Assistenten bereit, die Sie durch die folgenden Tasks führen:

- Cluster erstellen
- Unabhängigen Warteschlangenmanager mit einem Cluster verbinden

Programmable Command Formats (PCF)

<i>Tabelle 19. PCF-Entsprechungen zu MQSC-Befehlen für die Verwaltung von Clustern</i>	
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 Konzepte

„IBM WebSphere MQ-Clusterbefehle“ auf Seite 91

Die zu den IBM WebSphere 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.

Befehle für die Warteschlangenmanagerdefinition

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Um anzugeben, dass ein Warteschlangenmanager ein vollständiges Repository für ein Cluster enthält, verwenden Sie den Befehl ALTER QMGR unter Angabe des Attributs REPOS(*clustername*). Um eine Liste mit mehreren Clusternamen anzugeben, definieren Sie eine Clusternamensliste und verwenden Sie dann das Attribut REPOSNL(*namelist*) im Befehl ALTER QMGR :

```
DEFINE NAMELIST(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(*data*)

Gibt die Daten an, die an den Benutzerexit für Clusterauslastung übergeben werden sollen.

CLWLLEN(*length*)

Gibt die Höchstmenge an Nachrichtendaten an, die an den Benutzerexit für Clusterauslastung übergeben werden sollen.

CLWLMRUC (channels)

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 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

Use the SUSPEND QMGR and RESUME QMGR command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

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: Forcibly removing a queue manager from a cluster

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 WebSphere MQ den Namen des Kanals und ersetzt +QMNAME+ durch den korrekten Namen des vollständigen Warteschlangenmanager-Repositorys. Dieser 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 Warteschlangenmanager 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 WebSphere 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 im Befehl DEFINE CHANNEL und im Befehl ALTER CHANNEL gelten auch 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 Clustersenderkanal 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 automatisch definierter Clustersenderkanäle zu untersuchen.

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 CONNAME in einer CLUSRCVR-Definition

Unter Umständen können Sie den Wert CONNAME in einer CLUSRCVR-Definition weglassen. Unter z/OS darf der Wert CONNAME jedoch nicht weggelassen werden.

Auf AIX, HP-UX, IBM i, Linux, Solaris, and Windows-Plattformen ist der Parameter für den TCP/IP-Verbindungsnamen eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM WebSphere 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 CONNAME wird immer in Schreibweise mit Trennzeichen (IPv4) oder in hexadezimaler Form (IPv6) angegeben, nicht als alphanumerischer DNS-Hostname.

Diese Funktion ist hilfreich, wenn Ihre Systeme Dynamic Host Configuration Protocol (DHCP) verwenden. Wenn Sie für den Parameter CONNAME 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 CONNAME in der CLUSRCVR-Definition ein Leerzeichen angeben, generiert WebSphere MQ einen Wert für CONNAME aus der IP-Adresse des Systems. Nur der generierte CONNAME wird in den Repositorys gespeichert. Andere Warteschlangenmanager im Cluster wissen nicht, dass für CONNAME ursprünglich ein Leerzeichen angegeben war.

Wenn Sie den Befehl `DISPLAY CLUSQMGR` ausgeben, wird der generierte CONNAME angezeigt. Wenn Sie jedoch den Befehl `DISPLAY CHANNEL` vom lokalen Warteschlangenmanager absetzen, sehen Sie, dass CONNAME leer ist.

Wenn der Warteschlangenmanager aufgrund von DHCP gestoppt und mit einer anderen IP-Adresse neu gestartet wird, generiert WebSphere MQ CONNAME erneut und aktualisiert die Repositorys dementsprechend.

Zugehörige Verweise

[Befehle für die Warteschlangenmanagerdefinition](#)

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

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

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: Forcibly removing a queue manager from a cluster

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 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 Version 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 Version 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.

Die Attribute der Befehle DEFINE QLOCAL, DEFINE QREMOTE und DEFINE QALIAS gelten auch für den Befehl DISPLAY QUEUE .

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.

Der Befehl DISPLAY QUEUE oder DISPLAY QCLUSTER gibt den Namen des Warteschlangenmanagers zurück, der die Warteschlange enthält (oder die Namen aller Warteschlangenmanager, wenn mehrere Instanzen der Warteschlange vorhanden sind). Außerdem werden der Systemname für jeden Warteschlangenmanager, 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 Verweise

Befehle für die Warteschlangenmanagerdefinition

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

Use the SUSPEND QMGR and RESUME QMGR command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

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: Forcibly removing a queue manager from a cluster

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. Dazu gehören alle Warteschlangenmanager, an die versucht wurde, eine Nachricht zu senden, sowie alle 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:

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.

CLUSDATE

Das Datum, an dem die Definition dem lokalen Warteschlangenmanager zur Verfügung gestellt wurde.

CLUSTIME

Zeit, zu der die Definition für den lokalen Warteschlangenmanager verfügbar war.

STATUS

Gibt den Status des Clustersenderkanals für diesen Warteschlangenmanager an.

SUSPEND

Gibt an, ob der Warteschlangenmanager ausgesetzt wurde.

CLUSTER

Gibt an, zu welchen Clustern der Warteschlangenmanager gehört.

CHANNEL

Der Name des Clusterempfängerkanals für den Warteschlangenmanager.

XMITQ

Die vom Warteschlangenmanager verwendete Clusterübertragungswarteschlange. Die Eigenschaft ist nur auf anderen Plattformen als z/OS verfügbar.

Zugehörige Verweise

Befehle für die Warteschlangenmanagerdefinition

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

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

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: Forcibly removing a queue manager from a cluster

Verwenden Sie den Befehl `RESET CLUSTER`, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

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 Tasks

Verwalten eines Warteschlangenmanagers

Zugehörige Verweise

Befehle für die Warteschlangenmanagerdefinition

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: Forcibly removing a queue manager from a cluster

Verwenden Sie den Befehl `RESET CLUSTER`, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

SUSPEND QMGR

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 CLUSSDR-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 CLUSSDR-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 CLUSSDR-Definitionen erneut durchzuführen. Die Suche wird auch dann durchgeführt, wenn der CLUSSDR-Kanal den Warteschlangenmanager mit mehreren Clustern verbindet.

Anmerkung: Die Verwendung des Befehls **REFRESH CLUSTER** kann den Cluster zerstören, während er in Bearbeitung ist, indem er beispielsweise eine plötzliche Zunahme der Arbeit für die vollständigen Repositorys bewirkt, während sie die erneute Weitergabe von Clusterressourcen eines Warteschlangenmanagers verarbeiten. Aus diesem Grund wird empfohlen, die Verwendung des Befehls bei der täglichen Arbeit möglichst zu vermeiden und alternative Methoden zur Korrektur bestimmter Inkonsistenzen einzusetzen.

Zugehörige Konzepte

Clustering: [REFRESH CLUSTER und die Verlaufswarteschlange](#)

Zugehörige Verweise

[Befehle für die Warteschlangenmanagerdefinition](#)

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](#)

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

RESET CLUSTER: Forcibly removing a queue manager from a cluster

Verwenden Sie den Befehl `RESET CLUSTER`, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

RESET CLUSTER: Forcibly removing a queue manager from a cluster

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.

Die Angabe von `QUEUES(NO)` in einem `RESET CLUSTER`-Befehl ist 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` erzwungen 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 WebSphere 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. Wenn der beschädigte Warteschlangenmanager später wieder funktioniert, verwenden Sie den Befehl **REFRESH CLUSTER**, um den Effekt von **RESET CLUSTER** umzukehren und den Warteschlangenmanager an den Cluster zurückzugeben. Wenn sich der Warteschlangenmanager in einem Publish/Subscribe-Cluster befindet, müssen Sie dann den Befehl `REFRESH QMGR TYPE (PROXYSUB)` ausgeben, um alle erforderlichen Proxy-Subskriptionen wiederherzustellen. Siehe [Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster](#).

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 auszugeben, 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 richtige Vorgehensweise zum 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 Verweise

[Befehle für die Warteschlangenmanagerdefinition](#)

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](#)

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

[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

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt WebSphere MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei wird der Verwaltungsalgorithmus für Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager der am besten geeignete ist. Sie können den Algorithmus für Lastausgleich, mit dem der Warteschlangenmanager ausgewählt werden soll, angeben, indem Sie ein Exitprogramm für Clusterauslastung schreiben.

Passende Zieladressen werden auf Basis der Verfügbarkeit von Warteschlangenmanager und Warteschlange ausgewählt, sowie auf Basis der Anzahl spezifischer Attribute zur Clusterauslastung, die Warteschlangenmanagern, Warteschlangen und Kanälen zugeordnet sind.

Wenn die Ergebnisse des Algorithmus für Lastausgleich nicht Ihren Bedürfnissen entsprechen, können Sie ein Benutzerexitprogramm für Clusterauslastung schreiben. Verwenden Sie den Exit, um Nachrichten zu der Warteschlange Ihrer Wahl im Cluster weiterzuleiten.

Zugehörige Konzepte

[Aufruf des Exits für Clusterauslastung und Datenstrukturen](#)

Dieser Abschnitt enthält Referenzinformationen zum Exit für Clusterauslastung und zu den Datenstrukturen. Dies sind Informationen zur allgemeinen Programmierschnittstelle.

Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

In diesem 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:

<i>Tabelle 20. Attribute für das Clusterauslastungsmanagement</i>		
Warteschlangen	Warteschlangenmanager	Kanäle
<ul style="list-style-type: none"> • CLWLPRTY • CLWLRANK • CLWLUSEQ 	<ul style="list-style-type: none"> • CLWLUSEQ • CLWLMRUC 	<ul style="list-style-type: none"> • CLWLPRTY • CLWLRANK • CLWLWGHT • NETPRTY

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 bzw. Kanäle zu einem Warteschlangenmanager, der eine Warteschlange mit dem Namen `ObjectName` bereitstellt, die sich in einem der Cluster befindet, zu denen der Warteschlangenmanager gehört.

Durch die nachfolgenden Algorithmusschritte wird die Liste der möglichen Ziele reduziert.

1. Bei Angabe eines Warteschlangennamens:
 - a. Nicht aktivierte Warteschlangen werden aus der Liste der möglichen Ziele ausgeschlossen.
 - b. Ferne Instanzen von Warteschlangen, die nicht zum gleichen Cluster wie der lokale Warteschlangenmanager gehören, werden ausgeschlossen.
 - c. Ferne CLUSRCVR-Kanäle, die nicht zum gleichen Cluster wie die Warteschlange gehören, werden ausgeschlossen.
2. Bei Angabe eines Warteschlangenmanagernamens:
 - a. Nicht aktivierte Warteschlangenmanager-Aliasnamen werden ausgeschlossen.
 - b. Ferne CLUSRCVR-Kanäle, die nicht zum gleichen Cluster wie der lokale Warteschlangenmanager gehören, werden ausgeschlossen.
3. Wenn die verbleibende Gruppe der Warteschlangen die lokale Instanz der Warteschlange enthält, wird in der Regel diese Instanz verwendet. Die lokale Instanz einer Warteschlange wird verwendet, wenn eine der folgenden drei Bedingungen zutrifft:
 - Das Attribut `CLWLUSEQ` der Warteschlange ist auf `LOCAL` gesetzt.
 - Beide der folgenden Bedingungen treffen zu:
 - a. Das Attribut `CLWLUSEQ` der Warteschlange ist auf `QMGR` gesetzt.
 - b. Das Attribut `CLWLUSEQ` des Warteschlangenmanagers ist auf `LOCAL` gesetzt.
 - Die Nachricht wurde nicht direkt von einer lokalen Anwendung eingereicht, sondern über einen Clusterkanal empfangen.

Anmerkung: In einem Benutzerexit erkennen Sie eine über einen Clusterkanal übertragene Nachricht daran, dass weder das Flag MQWXP_PUT_BY_CLUSTER_CH noch das Flag MQQF_CLWL_USEQ_ANY gesetzt ist:

- MQWXP.Flags Flag MQWXP_PUT_BY_CLUSTER_CH.
 - MQWQR.QFlags Flag MQQF_CLWL_USEQ_ANY.
4. Bei einer Cluster-PCF-Nachricht wird jeder Warteschlangenmanager ausgeschlossen, an den bereits eine Veröffentlichung oder Subskription gesendet wurde.
 5. Alle Kanäle zu Warteschlangenmanagern bzw. Warteschlangenmanager-Aliasnamen mit einem CLWLRANK unter dem Maximalrang aller verbleibenden Kanäle oder Warteschlangenmanager-Aliasnamen werden ausgeschlossen.
 6. Alle Warteschlangen (nicht Warteschlangenmanager-Aliasnamen) mit einem CLWLRANK unter dem Maximalrang aller verbleibenden Warteschlangen werden ausgeschlossen.
 7. Wenn nur ferne Instanzen einer Warteschlange verbleiben, werden wieder aufgenommene Warteschlangenmanager gegenüber ausgesetzten Warteschlangenmanagern bevorzugt.
 8. Wenn mehrere ferne Instanzen einer Warteschlange 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 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 ferne Instanz einer Warteschlange verbleibt, werden alle Kanäle mit dem Status MQCHS_RETRYING aufgelistet.
 11. Wenn keine ferne Instanz einer Warteschlange verbleibt, werden alle Kanäle mit den Status REQUESTING, PAUSED oder STOPPED aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
 - MQCHS_REQUESTING
 - MQCHS_PAUSED
 - MQCHS_STOPPED
 12. Wenn mehrere ferne Instanzen einer Warteschlange verbleiben und es sich um eine Cluster-PCF-Nachricht handelt, werden lokal definierte CLUSSDR-Kanäle ausgewählt.
 13. Wenn mehrere ferne Instanzen einer Warteschlange eines beliebigen Warteschlangenmanagers verbleiben, wird für jeden Warteschlangenmanager die Warteschlange mit dem jeweils höchsten NETPRTY-Wert ausgewählt.
 14. Bei Auswahl eines Warteschlangenmanagers:
 - 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.
 15. Bei Auswahl einer Warteschlange:
 - Alle Warteschlangen mit Ausnahme der Warteschlangen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen, Kanäle bleiben aber erhalten.
 16. Alle Kanäle mit Ausnahme der Kanäle mit den höchsten Werten für MQWDR.DestSeqNumber werden ausgeschlossen. Die Eliminierung wird gestoppt, wenn die Anzahl der verbleibenden Kanäle nicht größer ist als die maximal zulässige Anzahl der zuletzt verwendeten Kanäle, CLWLMRUC.

17. Wenn mehrere ferne Instanzen einer Warteschlange verbleiben, wird der am längsten nicht mehr verwendete Kanal ausgewählt. Dieser Kanal hat den niedrigsten MQWDR.DestSeqFactor-Wert.
- Falls mehrere Kanäle mit diesem Wert vorliegen, wird einer der Kanäle mit dem niedrigsten MQWDR.DestSeqNumber-Wert ausgewählt.
 - Der ausgewählte Zielsequenzfaktor wird vom Warteschlangenmanager um ca. 1000/CLWLWGHT erhöht.

Anmerkung:

- a. Bei einer Änderung der Clusterauslastungsattribute der verfügbaren CLUSRCVR-Kanäle werden die Zielsequenzfaktoren aller Ziele auf null zurückgesetzt. Außerdem werden die Sequenzfaktoren auf null zurückgesetzt, sobald neue CLUSRCVR-Kanäle verfügbar werden.
- b. Bei einer Änderung der Auslastungsattribute manuell definierter CLUSSDR-Kanäle werden die Zielsequenzfaktoren nicht zurückgesetzt.

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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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 von CLUSSDR- und CLUSRCVR-Kanälen 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 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. WebSphere MQ wählt Ziele mit höherer Priorität innerhalb des Clusters vor Zielen 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.

WebSphere 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. Sie könnten getrennte Warteschlangen verwenden oder CLWLRANK mit manuellem Umschalten vom primären zum Sicherungsziel.

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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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 von CLUSSDR- und CLUSRCVR-Kanälen 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.

WebSphere 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.

Würden Sie hingegen das Prioritätsattribut 'CLWLPRTY' verwenden, würde WebSphere MQ nur zwischen verfügbaren Zielen auswählen. 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** gibt die Priorität von **CLUSDDR-** und **CLUSRCVR-**Kanälen für eine gleichmäßige Clusterauslastung 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 von **CLUSDDR-** und **CLUSRCVR-**Kanälen 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.

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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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 von CLUSSDR- und CLUSRCVR-Kanälen 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.

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** gibt die Priorität von **CLUSDR-** und **CLUSRCVR-**Kanälen für eine gleichmäßige Clusterauslastung 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 von **CLUSDR-** und **CLUSRCVR-**Kanälen 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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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 von CLUSSDR- und CLUSRCVR-Kanälen 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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. 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 Priorität eines CLUSSDR- oder eines CLUSRCVR-Kanals fest. IBM WebSphere 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.

WebSphere MQ ruft die Priorität von Kanälen 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 (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. Sie könnten getrennte Warteschlangen verwenden oder CLWLRANK mit manuellem Umschalten vom primären zum Sicherheitsziel.

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 von `CLUSDR`- und `CLUSRCVR`-Kanälen 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.

Kanalattribut `CLWLRANK`

Das Kanalattribut `CLWLRANK` gibt den Rang von `CLUSDR`- und `CLUSRCVR`-Kanälen 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.

WebSphere 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.

Würden Sie hingegen das Prioritätsattribut '`CLWLPRTY`' verwenden, würde WebSphere MQ nur zwischen verfügbaren Zielen auswählen. 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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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.

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.

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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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 von CLUSSDR- und CLUSRCVR-Kanälen 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.

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 gibt die Priorität von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung 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 von CLUSSDR- und CLUSRCVR-Kanälen 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.

Aufruf des Exits für Clusterauslastung und Datenstrukturen

Dieser Abschnitt enthält Referenzinformationen zum Exit für Clusterauslastung und zu den Datenstrukturen. Dies sind Informationen zur allgemeinen Programmierschnittstelle.

Exits für Clusterauslastung können in folgenden Programmiersprachen geschrieben werden:

- C
- System/390 Assembler (WebSphere MQ for z/OS)

Eine Beschreibung des Aufrufs finden Sie in:

- [„MQ_CLUSTER_WORKLOAD_EXIT - Beschreibung des Aufrufs“](#) auf Seite 117

Eine Beschreibung der vom Exit verwendeten Datenstrukturen finden Sie in:

- [„MQXCLWLN - Navigieren in den Clusterauslastungsdatensätzen“](#) auf Seite 119
- [„MQWXP - Parameterstruktur des Exits für Clusterauslastung“](#) auf Seite 122
- [„MQWDR - Zieldatensatzstruktur für Clusterauslastung“](#) auf Seite 130
- [„MQWQR - Struktur des Warteschlangendatensatzes für Clusterauslastung“](#) auf Seite 134
- [„MQWCR - Struktur des Clusterdatensatzes für Clusterauslastung“](#) auf Seite 140
-

In diesem Abschnitt werden Warteschlangenmanagerattribute und Warteschlangenattribute immer mit vollständigem Namen angegeben. Die entsprechenden Namen, die im Buch mit den MQSC-Befehlen verwendet werden, sind unten aufgeführt. Details zu MQSC-Befehlen finden Sie unter [MQSC-Referenz](#) .

<i>Tabelle 21. Warteschlangenmanagerattribute</i>	
Vollständiger Name	Name im MQSC
<i>ClusterWorkloadData</i>	CLWLDATA
<i>ClusterWorkloadExit</i>	CLWLEXIT
<i>ClusterWorkloadLength</i>	CLWLLEN

<i>Tabelle 22. Warteschlangenattribute</i>	
Vollständiger Name	Name im MQSC
<i>DefBind</i>	DEFBIND
<i>DefPersistence</i>	DEFPSIST
<i>DefPriority</i>	DEFPRTY
<i>InhibitPut</i>	EINREIHEN
<i>QDesc</i>	DESCR

Zugehörige Konzepte

Lastausgleich

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt WebSphere MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei wird der Verwaltungsalgorithmus für Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager der am besten geeignete ist. Sie können den Algorithmus für Lastausgleich, mit dem der Warteschlangenmanager ausgewählt werden soll, angeben, indem Sie ein Exitprogramm für Clusterauslastung schreiben.

MQ_CLUSTER_WORKLOAD_EXIT - Beschreibung des Aufrufs

Der Exit für Clusterauslastung wird vom Warteschlangenmanager zur Weiterleitung einer Nachricht an einen verfügbaren Warteschlangenmanager aufgerufen.

Anmerkung: Vom Warteschlangenmanager wird kein Einstiegspunkt mit dem Namen MQ_CLUSTER_WORKLOAD_EXIT bereitgestellt. Stattdessen wird der Name des Exits für Clusterauslastung durch das Attribut `ClusterWorkloadExit` des Warteschlangenmanagers definiert.

Der Exit MQ_CLUSTER_WORKLOAD_EXIT wird auf allen Plattformen unterstützt.

Syntax

```
MQ_CLUSTER_WORKLOAD_EXIT (ExitParms)
```

Zugehörige Verweise

MQXCLWLN - Navigieren in den Clusterauslastungsdatensätzen

Der Aufruf MQXCLWLN wird verwendet, um durch die im Cluster-Cache gespeicherten Ketten der MQWDR-, MQWQR- und MQWCR -Datensätze zu navigieren.

MQWXP - Parameterstruktur des Exits für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP.

MQWDR - Zieldatensatzstruktur für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Zieldatensatzstruktur für Clusterauslastung MQWDR.

MQWQR - Struktur des Warteschlangendatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

MQWCR - Struktur des Clusterdatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Parameter für MQ_CLUSTER_WORKLOAD_EXIT

Beschreibung der Parameter im Aufruf MQ_CLUSTER_WORKLOAD_EXIT.

ExitParms(MQWXP) - Ein-/Ausgabe

Exit-Parameterblock.

- Der Exit gibt Informationen in MQWXP an, um anzugeben, wie die Auslastung verwaltet werden sollte.

Zugehörige Verweise

Hinweise zur Verwendung

Die Funktion des Exits für Clusterauslastung wird vom Exit-Provider definiert. Der Exit muss jedoch den im zugehörigen Steuerblock MQWXP definierten Regeln entsprechen.

Aufrufe in Programmiersprachen für MQ_CLUSTER_WORKLOAD_EXIT

MQ_CLUSTER_WORKLOAD_EXIT unterstützt die beiden Programmiersprachen C und High Level Assembler.

Hinweise zur Verwendung

Die Funktion des Exits für Clusterauslastung wird vom Exit-Provider definiert. Der Exit muss jedoch den im zugehörigen Steuerblock MQWXP definierten Regeln entsprechen.

Vom Warteschlangenmanager wird kein Einstiegspunkt mit dem Namen MQ_CLUSTER_WORKLOAD_EXIT bereitgestellt. Allerdings wird typedef für den Namen MQ_CLUSTER_WORKLOAD_EXIT in der Programmiersprache C bereitgestellt. Verwenden Sie typedef zur Deklaration des benutzerdefinierten Exits, um sicherzustellen, dass die Parameter stimmen.

Zugehörige Verweise

Parameter für MQ_CLUSTER_WORKLOAD_EXIT

Beschreibung der Parameter im Aufruf MQ_CLUSTER_WORKLOAD_EXIT.

Aufrufe in Programmiersprachen für MQ_CLUSTER_WORKLOAD_EXIT

MQ_CLUSTER_WORKLOAD_EXIT unterstützt die beiden Programmiersprachen C und High Level Assembler.

Aufrufe in Programmiersprachen für MQ_CLUSTER_WORKLOAD_EXIT

MQ_CLUSTER_WORKLOAD_EXIT unterstützt die beiden Programmiersprachen C und High Level Assembler.

C-Aufruf

```
MQ_CLUSTER_WORKLOAD_EXIT (&ExitParms);
```

Ersetzen Sie *MQ_CLUSTER_WORKLOAD_EXIT* durch den Namen Ihrer Exitfunktion für Clusterauslastung.

Deklarieren Sie die *MQ_CLUSTER_WORKLOAD_EXIT*-Parameter wie folgt:

```
MQWXP ExitParms; /* Exit parameter block */
```

Aufruf von High Level Assembler

```
CALL EXITNAME, (EXITPARMS)
```

Deklarieren Sie die Parameter wie folgt:

```
EXITPARMS          CMQWXP          Exit parameter block
```

Zugehörige Verweise

Parameter für MQ_CLUSTER_WORKLOAD_EXIT

Beschreibung der Parameter im Aufruf MQ_CLUSTER_WORKLOAD_EXIT.

Hinweise zur Verwendung

Die Funktion des Exits für Clusterauslastung wird vom Exit-Provider definiert. Der Exit muss jedoch den im zugehörigen Steuerblock MQWXP definierten Regeln entsprechen.

MQXCLWLN - Navigieren in den Clusterauslastungsdatensätzen

Der Aufruf MQXCLWLN wird verwendet, um durch die im Cluster-Cache gespeicherten Ketten der MQWDR-, MQWQR- und MQWCR -Datensätze zu navigieren.

Der Clustercache ist ein Bereich des Hauptspeichers, in dem Daten zum Cluster gespeichert werden.

Wenn der Clustercache statisch ist, hat er eine festgelegte Größe. Wenn Sie ihn auf dynamisch (DYNAMIC) setzen, kann der Clustercache je nach Bedarf erweitert werden.

Legen Sie für den Clustercachetyp STATIC oder DYNAMIC fest, indem Sie einen Systemparameter oder ein Makro verwenden.

- Verwenden Sie den Systemparameter `ClusterCacheType` auf allen Plattformen außer auf z/OS.
- Verwenden Sie den Parameter `CLCACHE` im `CSQ6SYSP`-Makro unter z/OS.

Syntax

```
MQXCLWLN (ExitParms, CurrentRecord, NextOffset, NextRecord, Compcode, Reason)
```

Zugehörige Verweise

[MQ_CLUSTER_WORKLOAD_EXIT](#) - Beschreibung des Aufrufs

Der Exit für Clusterauslastung wird vom Warteschlangenmanager zur Weiterleitung einer Nachricht an einen verfügbaren Warteschlangenmanager aufgerufen.

[MQWXP](#) - Parameterstruktur des Exits für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP.

[MQWDR](#) - Zieldatensatzstruktur für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Zieldatensatzstruktur für Clusterauslastung MQWDR.

[MQWQR](#) - Struktur des Warteschlangendatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

[MQWCR](#) - Struktur des Clusterdatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Parameter für MQXCLWLN für das Navigieren in den Clusterauslastungsdatensätzen

Beschreibung der Parameter im MQXCLWLN-Aufruf.

ExitParms (MQWXP) - Ein-/Ausgabe

Exit-Parameterblock.

Diese Struktur enthält Informationen zum Aufruf des Exits. Der Exit gibt Informationen in dieser Struktur an, um anzugeben, wie die Auslastung verwaltet werden sollte.

CurrentRecord (MQPTR) - Eingabe

Adresse des aktuellen Datensatzes.

Diese Struktur enthält Informationen zu der Adresse des Datensatzes, der derzeit vom Exit geprüft wird. Der Datensatz muss einer der folgenden Typen sein:

- Zieldatensatz für Clusterauslastung (MQWDR)
- Warteschlangendatensatz für Clusterauslastung (MQWQR)
- Clusterdatensatz für Clusterauslastung (MQWCR)

NextOffset (MQLONG) - Eingabe

Relative Adresse des nächsten Datensatzes.

Diese Struktur enthält Informationen zu der relativen Adresse des nächsten Datensatzes oder der nächsten Struktur. *NextOffset* ist der Wert des zugeordneten Felds für die relative Adresse im aktuellen Datensatz. Dieses muss eins der folgenden Felder sein:

- Feld `ChannelDefOffset` in MQWDR
- Feld `ClusterRecOffset` in MQWDR
- Feld `ClusterRecOffset` in MQWQR
- Feld `ClusterRecOffset` in MQWCR

NextRecord (MQPTR) - Ausgabe

Adresse des nächsten Datensatzes oder der nächsten Struktur.

Diese Struktur enthält Informationen zur Adresse des nächsten Datensatzes oder der nächsten Struktur. Wenn *CurrentRecord* die Adresse einer MQWDR-Struktur und *NextOffset* der Wert des Felds `ChannelDefOffset` ist, dann ist *NextRecord* die Adresse der Kanaldefinitionsstruktur (MQCD).

Wenn es keinen nächsten Datensatz oder nächste Struktur gibt, setzt der Warteschlangenmanager für *NextRecord* den Leerzeiger, und der Aufruf gibt den Beendigungscode `MQCC_WARNING` und den Ursachencode `MQRC_NO_RECORD_AVAILABLE` zurück.

CompCode (MQLONG) - Ausgabe

Beendigungscode.

Der Beendigungscode hat einen der folgenden Werte:

MQCC_OK

Erfolgreiche Fertigstellung.

MQCC_WARNING

Warnung (teilweise Ausführung)

MQCC_FAILED

Aufruf fehlgeschlagen.

Reason (MQLONG) - Ausgabe

Ursachencode, der CompCode qualifiziert.

Wenn CompCode auf `MQCC_OK` gesetzt ist:

MQRC_NONE

(0, X'0000')

Keine Ursache zurückzumelden.

Wenn CompCode auf `MQCC_WARNING` gesetzt ist:

MQRC_NO_RECORD_AVAILABLE

(2359, X'0937')

Kein Datensatz verfügbar. Ein MQXCLWLN-Aufruf wurde von einem Exit für Clusterauslastung ausgegeben, um die Adresse des nächsten Datensatzes in der Kette anzufordern. Der aktuelle Datensatz ist der letzte Datensatz in der Kette. Korrekturmaßnahme: Keine.

Wenn CompCode auf `MQCC_FAILED` gesetzt ist:

MQRC_CURRENT_RECORD_ERROR

(2357, X'0935')

Der Parameter *CurrentRecord* ist nicht gültig. Ein MQXCLWLN-Aufruf wurde von einem Exit für Clusterauslastung ausgegeben, um die Adresse des nächsten Datensatzes in der Kette anzufordern. Die im Parameter *CurrentRecord* angegebene Adresse ist nicht die Adresse eines gültigen Datensatzes.

CurrentRecord muss die Adresse eines Zieldatensatzes (MQWDR), eines Warteschlangendatensatzes (MQWQR) oder eines Clusterdatensatzes (MQWCR) innerhalb des Clustercaches angeben. Korrekturmaßnahme: Sicherstellen, dass der Exit für Clusterauslastung die Adresse eines gültigen Datensatzes innerhalb des Clustercaches übergibt.

MQRC_ENVIRONMENT_ERROR

(2012, X'07DC')

Aufruf ist in dieser Umgebung nicht gültig. Ein MQXCLWLN-Aufruf wurde ausgegeben, aber nicht von einem Exit für Clusterauslastung.

MQRC_NEXT_OFFSET_ERROR

(2358, X'0936')

Der Parameter *NextOffset* ist nicht gültig. Ein MQXCLWLN-Aufruf wurde von einem Exit für Clusterauslastung ausgegeben, um die Adresse des nächsten Datensatzes in der Kette anzufordern. Die durch den Parameter *NextOffset* angegebene relative Adresse ist nicht gültig. Für *NextOffset* muss der Wert eines der folgenden Felder angegeben werden:

- Feld `ChannelDefOffset` in MQWDR
- Feld `ClusterRecOffset` in MQWDR
- Feld `ClusterRecOffset` in MQWQR
- Feld `ClusterRecOffset` in MQWCR

Korrekturmaßnahme: Sicherstellen, dass der für den Parameter *NextOffset* angegebene Wert dem Wert eines der zuvor aufgeführten Felder entspricht.

MQRC_NEXT_RECORD_ERROR

(2361, X'0939')

Der Parameter *NextRecord* ist nicht gültig.

MQRC_WXP_ERROR

(2356, X'0934')

Die Parameterstruktur des Auslastungsexits ist nicht gültig. Ein MQXCLWLN-Aufruf wurde von einem Exit für Clusterauslastung ausgegeben, um die Adresse des nächsten Datensatzes in der Kette anzufordern. Die Parameterstruktur des Exits für Clusterauslastung *ExitParms* ist nicht gültig; dies kann eine der folgenden Ursachen haben:

- Der Parameterzeiger ist nicht gültig. Es ist nicht immer möglich, ungültige Parameterzeiger zu erkennen; andernfalls treten unvorhersehbare Ergebnisse auf.
- Das Feld `StrucId` enthält nicht den Wert `MQWXP_STRUC_ID`.
- Das Feld `Version` enthält nicht den Wert `MQWXP_VERSION_2`.
- Das Feld `Context` enthält nicht den durch den Warteschlangenmanager an den Exit übergebenen Wert.

Korrekturmaßnahme: Sicherstellen, dass der für *ExitParms* angegebene Parameter die MQWXP-Struktur ist, die beim Aufruf des Exits an den Exit übergeben wurde.

Zugehörige Verweise

Hinweise zu MQXCLWLN für das Navigieren in den Clusterauslastungsdatensätzen

Verwenden Sie MQXCLWLN, um auch in einem statischen Cache durch Clusterdatensätze zu navigieren.

Aufrufe in Programmiersprachen für MQXCLWLN

MQXCLWLN unterstützt die beiden Programmiersprachen C und High Level Assembler.

Hinweise zu MQXCLWLN für das Navigieren in den Clusterauslastungsdatensätzen

Verwenden Sie MQXCLWLN, um auch in einem statischen Cache durch Clusterdatensätze zu navigieren.

Wenn der Clustercache dynamisch ist, muss der MQXCLWLN-Aufruf verwendet werden, um in den Datensätzen zu navigieren. Der Exit wird abnormal beendet, wenn einfache Arithmetik mit Zeiger und Offset verwendet wird, um in den Datensätzen zu navigieren.

Wenn der Clustercache statisch ist, muss der Aufruf MQXCLWLN nicht verwendet werden, um durch die Datensätze zu navigieren. Üblicherweise wird MQXCLWLN jedoch auch dann verwendet, wenn der Cache statisch ist. In diesem Fall können Sie den Clustercache in den dynamischen Zustand versetzen, ohne Änderungen am Auslastungsexit vornehmen zu müssen.

Zugehörige Verweise

Parameter für MQXCLWLN für das Navigieren in den Clusterauslastungsdatensätzen

Beschreibung der Parameter im MQXCLWLN-Aufruf.

Aufrufe in Programmiersprachen für MQXCLWLN

MQXCLWLN unterstützt die beiden Programmiersprachen C und High Level Assembler.

Aufrufe in Programmiersprachen für MQXCLWLN

MQXCLWLN unterstützt die beiden Programmiersprachen C und High Level Assembler.

C-Aufruf

```
MQXCLWLN (&ExitParms, CurrentRecord, NextOffset, &NextRecord, &CompCode, &Reason) ;
```

Deklarieren Sie die Parameter wie folgt:

```
Typedef struct tagMQXCLWLN {
    MQWXP      ExitParms;          /* Exit parameter block */
    MQPTR      CurrentRecord;      /* Address of current record*/
    MQLONG     NextOffset;        /* Offset of next record */
    MQPTR      NextRecord;        /* Address of next record or structure */
    MQLONG     CompCode;         /* Completion code */
    MQLONG     Reason;           /* Reason code qualifying CompCode */
}
```

Aufruf von High Level Assembler

```
CALL MQXCLWLN, (CLWLEXITPARMS, CURRENTRECORD, NEXTOFFSET, NEXTRECORD, COMPCODE, REASON)
```

Deklarieren Sie die Parameter wie folgt:

CLWLEXITPARMS	CMQWXP,	Cluster workload exit parameter block
CURRENTRECORD	CMQWDRA,	Current record
NEXTOFFSET	DS F	Next offset
NEXTRECORD	DS F	Next record
COMPCODE	DS F	Completion code
REASON	DS F	Reason code qualifying COMPCODE

Zugehörige Verweise

Parameter für MQXCLWLN für das Navigieren in den Clusterauslastungsdatensätzen

Beschreibung der Parameter im MQXCLWLN-Aufruf.

Hinweise zu MQXCLWLN für das Navigieren in den Clusterauslastungsdatensätzen

Verwenden Sie MQXCLWLN, um auch in einem statischen Cache durch Clusterdatensätze zu navigieren.

MQWXP - Parameterstruktur des Exits für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP.

Tabelle 23. Felder in MQWXP		
Feld	Beschreibung	Seite
<i>StrucId</i>	Struktur-ID	StrucId
<i>Version</i>	Strukturversionsnummer	Version
<i>ExitId</i>	Exittyp	ExitId
<i>ExitReason</i>	Grund für das Aufrufen des Exits	ExitReason
<i>ExitResponse</i>	Antwort vom Exit	ExitResponse
<i>ExitResponse2</i>	Sekundäre Exit-Antwort	ExitResponse2
<i>Feedback</i>	Rückmeldungscode	Rückmeldung

Tabelle 23. Felder in MQWXP (Forts.)

Feld	Beschreibung	Seite
<i>Flags</i>	Flagwerte. Diese Bit-Flags werden verwendet, um Informationen zu der Nachricht anzuzeigen, die eingereicht wird.	Flags
<i>ExitUserArea</i>	Exit-Benutzerbereich	ExitUserArea
<i>ExitData</i>	Exitdaten	ExitData
<i>MsgDescPtr</i>	Adresse des Nachrichtendeskriptors (MQMD)	MsgDescPtr
<i>MsgBufferPtr</i>	Adresse des Puffers, der alle oder einige der Nachrichtendaten enthält	MsgBufferPtr
<i>MsgBufferLength</i>	Länge des Puffers, der die Nachrichtendaten enthält	MsgBufferLength
<i>MsgLength</i>	Länge der gesamten Nachricht	MsgLength
<i>QName</i>	Name der Warteschlange	QName
<i>QMgrName</i>	Name des lokalen Warteschlangenmanagers	QMgrName
<i>DestinationCount</i>	Anzahl der möglichen Zieladressen	DestinationCount
<i>DestinationChosen</i>	Ausgewählte Zieladresse	DestinationChosen
<i>DestinationArrayPtr</i>	Adresse eines Bereichs von Zeigern auf Zieldatensätze (MQWDR)	DestinationArrayPtr
<i>QArrayPtr</i>	Adresse eines Bereichs von Zeigern auf Warteschlangendatensätze (MQWQR)	QArrayPtr
Anmerkung: Die übrigen Felder werden ignoriert, wenn 'Version' kleiner als MQWXP_VERSION_2 ist.		
<i>CacheContext</i>	Kontextinformationen	CacheContext
<i>CacheType</i>	Clustercachetyp	CacheType
Anmerkung: Die übrigen Felder werden ignoriert, wenn 'Version' kleiner als MQWXP_VERSION_3 ist.		
<i>CLWLMRUChannels</i>	Maximale Anzahl an zulässigen aktiven abgehenden Clusterkanälen	CLWLMRUChannels
Anmerkung: Die übrigen Felder werden ignoriert, wenn 'Version' kleiner als MQWXP_VERSION_4 ist.		
<i>pEntryPoints</i>	Adresse der MQIEP-Struktur, um MQI- und DCI-Aufrufe zu ermöglichen	pEntryPoints

Die Parameterstruktur des Exits für Clusterauslastung beschreibt Daten, die an den Exit für Clusterauslastung übermittelt werden.

Die Parameterstruktur des Exits für Clusterauslastung wird auf allen Plattformen unterstützt.

Zusätzlich sind die MQWXP1-, MQWXP2- und MQWXP3-Strukturen für Abwärtskompatibilität verfügbar.

Zugehörige Verweise

[MQ_CLUSTER_WORKLOAD_EXIT](#) - Beschreibung des Aufrufs

Der Exit für Clusterauslastung wird vom Warteschlangenmanager zur Weiterleitung einer Nachricht an einen verfügbaren Warteschlangenmanager aufgerufen.

[MQXCLWLN](#) - Navigieren in den Clusterauslastungsdatensätzen

Der Aufruf MQXCLWLN wird verwendet, um durch die im Cluster-Cache gespeicherten Ketten der MQWDR-, MQWQR- und MQWCR -Datensätze zu navigieren.

MQWDR - Zieldatensatzstruktur für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Zieldatensatzstruktur für Clusterauslastung MQWDR.

MQWQR - Struktur des Warteschlangendatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

MQWCR - Struktur des Clusterdatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP

Beschreibung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP

StrucId (MQCHAR4) - Eingabe

Die Struktur-ID für die Parameterstruktur des Exits für Clusterauslastung.

- Der Wert für StrucId ist MQWXP_STRUC_ID.
- Für die Programmiersprache C wird die Konstante MQWXP_STRUC_ID_ARRAY ebenfalls definiert. Der Wert entspricht MQWXP_STRUC_ID. Es handelt sich um eine Gruppe von Zeichen, nicht um eine Zeichenfolge.

Version (MQLONG) - Eingabe

Gibt die Strukturversionsnummer an. Version hat einen der folgenden Werte:

MQWXP_VERSION_1

Parameterstruktur des Exits für Clusterauslastung Version-1.

MQWXP_VERSION_1 wird in allen Umgebungen unterstützt.

MQWXP_VERSION_2

Parameterstruktur des Exits für Clusterauslastung Version-2.

MQWXP_VERSION_2 wird in den folgenden Umgebungen unterstützt: AIX, HP-UX, Linux, IBM i, Solaris und Windows.

MQWXP_VERSION_3

Parameterstruktur des Exits für Clusterauslastung Version-3.

MQWXP_VERSION_3 wird in den folgenden Umgebungen unterstützt: AIX, HP-UX, Linux, IBM i, Solaris und Windows.

MQWXP_VERSION_4

Parameterstruktur des Exits für Clusterauslastung Version-4.

MQWXP_VERSION_4 wird in den folgenden Umgebungen unterstützt: AIX, HP-UX, Linux, IBM i, Solaris und Windows.

MQWXP_CURRENT_VERSION

Aktuelle Version der Parameterstruktur des Exits für Clusterauslastung.

ExitId (MQLONG) - Eingabe

Gibt den Exittyp an, der aufgerufen wird. Der Exit für Clusterauslastung ist der einzige Exit, der unterstützt wird.

- Der Wert von ExitId muss MQXT_CLUSTER_WORKLOAD_EXIT lauten.

ExitReason (MQLONG) - Eingabe

Gibt den Grund für den Aufruf des Exits für Clusterauslastung an. ExitReason hat einen der folgenden Werte:

MQXR_INIT

Gibt an, dass der Exit zum ersten Mal aufgerufen wird.

Anforderung und Initialisierung von Ressourcen, die der Exit möglicherweise benötigt, z. B. Hauptspeicher.

MQXR_TERM

Gibt an, dass der Exit in Kürze beendet wird.

Freigabe von Ressourcen, die der Exit möglicherweise seit der Initialisierung angefordert hat, z. B. Hauptspeicher.

MQXR_CLWL_OPEN

Aufgerufen durch MQOPEN.

MQXR_CLWL_PUT

Aufgerufen durch MQPUT oder MQPUT1.

MQXR_CLWL_MOVE

Aufgerufen durch Nachrichtenkanalagent, wenn sich der Kanalstatus geändert hat.

MQXR_CLWL_REPOS

Aufgerufen durch MQPUT oder MQPUT1 für eine PCF-Nachricht des Repository-Managers.

MQXR_CLWL_REPOS_MOVE

Aufgerufen durch Nachrichtenkanalagent für eine PCF-Nachricht des Repository-Managers, wenn sich der Kanalstatus geändert hat.

ExitResponse (MQLONG) - Ausgabe

Legen Sie einen Wert für ExitResponse fest, um anzugeben, ob die Verarbeitung der Nachricht fortgesetzt wird. Folgende Werte sind zulässig:

MQXCC_OK

Normale Fortsetzung der Nachrichtenverarbeitung.

- DestinationChosen identifiziert die Zieladresse, an die die Nachricht gesendet werden soll.

MQXCC_SUPPRESS_FUNCTION

Einstellung der Nachrichtenverarbeitung.

- Die vom Warteschlangenmanager durchgeführten Aktionen basieren auf dem Grund für den Aufruf des Exits:

ExitReason	Durchgeführte Aktion
– MQXR_CLWL_OPEN – MQXR_CLWL_REPOS – MQXR_CLWL_PUT	Der Aufruf MQOPEN, MQPUT oder MQPUT1 schlägt mit dem Beendigungscode MQCC_FAILED und dem Ursachencode MQRC_STOPPED_BY_CLUSTER_EXIT fehl.
– MQXR_CLWL_MOVE – MQXR_CLWL_REPOS_MOVE	Die Nachricht wird in die Warteschlange für nicht zustellbare Nachrichten eingereiht.

MQXCC_SUPPRESS_EXIT

Normale Fortsetzung der aktuellen Nachricht. Rufen Sie den Exit nicht mehr auf, bis der Warteschlangenmanager beendet wurde.

Der Warteschlangenmanager verarbeitet nachfolgende Nachrichten als wäre das Warteschlangenmanagerattribut ClusterWorkloadExit nicht belegt. DestinationChosen identifiziert die Zieladresse, an die die aktuelle Nachricht gesendet wird.

Alle anderen Werte

Verarbeitung der Nachricht als wäre MQXCC_SUPPRESS_FUNCTION angegeben.

ExitResponse2 (MQLONG) - Ein-/Ausgabe

Legen Sie einen Wert für ExitResponse2 fest, um dem Warteschlangenmanager weitere Informationen bereitzustellen.

- MQXR2_STATIC_CACHE ist der Standardwert, der bei Zugriff auf den Exit festgelegt wird.

- Wenn `ExitReason` den Wert `MQXR_INIT` hat, kann der Exit einen der folgenden Werte in `ExitResponse2` festlegen:

MQXR2_STATIC_CACHE

Der Exit erfordert einen statischen Cluster-Cache.

- Wenn der Cluster-Cache statisch ist, muss der Exit nicht den Aufruf `MQXCLWLN` verwenden, um durch die Datensatzketten im Cluster-Cash zu navigieren.
- Wenn der Cluster-Cache dynamisch ist, kann der Exit nicht ordnungsgemäß durch die Datensätze im Cache navigieren.

Anmerkung: Der Warteschlangenmanager verarbeitet die Rückgabe des Aufrufs `MQXR_INIT` als hätte der Exit `MQXCC_SUPPRESS_EXIT` im `ExitResponse`-Feld zurückgegeben.

MQXR2_DYNAMIC_CACHE

Der Exit kann mit einem statischen oder dynamischen Cache arbeiten.

- Wenn der Exit diesen Wert zurückgibt, muss der Exit den Aufruf `MQXCLWLN` verwenden, um durch die Datensatzketten im Cluster-Cache zu navigieren.

Feedback (MQLONG) - Eingabe

Ein reserviertes Feld. Der Wert ist 0.

Flags (MQLONG) - Eingabe

Gibt Informationen zur eingereichten Nachricht an.

- Der Wert von `Flags` ist `MQWXP_PUT_BY_CLUSTER_CHL`. Die Nachricht stammt von einem Clusterkanal und ist nicht lokal oder von einem anderen Kanal. Demnach kommt die Nachricht von einem anderen Cluster-Warteschlangenmanager.

Reserved (MQLONG) - Eingabe

Ein reserviertes Feld. Der Wert ist 0.

ExitUserArea (MQBYTE16) - Ein-/Ausgabe

Legen Sie einen Wert für `ExitUserArea` fest, um zwischen Aufrufen des Exits zu kommunizieren.

- `ExitUserArea` wird vor dem ersten Aufruf des Exits mit einer binären Null initialisiert. Alle Änderungen, die der Exit an diesem Feld vornimmt, werden für sämtliche Aufrufe des Exits zwischen dem Aufruf `MQCONN` und dem zugehörigen Aufruf `MQDISC` gespeichert. Das Feld wird auf eine binäre Null zurückgesetzt, wenn der Aufruf `MQDISC` erfolgt.
- Beim ersten Aufruf des Exits hat das `ExitReason`-Feld den Wert `MQXR_INIT`.
- Die folgenden Konstanten werden definiert:

MQXUA_NONE - Zeichenfolge

MQXUA_NONE_ARRAY - Zeichenbereich

Keine Benutzerinformationen. Beide Konstanten haben eine binäre Null als Feldlänge.

MQ_EXIT_USER_AREA_LENGTH

Die Länge von `ExitUserArea`.

ExitData (MQCHAR32) - Eingabe

Der Wert des Warteschlangenmanagerattributs `ClusterWorkloadData`. Wenn für dieses Attribut kein Wert festgelegt wurde, bleibt dieses Feld leer.

- Die Länge von `ExitData` wird durch `MQ_EXIT_DATA_LENGTH` angegeben.

MsgDescPtr (PMQMD) - Eingabe

Die Adresse einer Kopie des Nachrichtendesktors (MQMD) für die Nachricht, die gerade verarbeitet wird.

- Alle Änderungen am Nachrichtendesktors durch den Exit werden vom Warteschlangenmanager ignoriert.
- Wenn `ExitReason` einen der folgenden Werte hat, ist `MsgDescPtr` auf den Nullzeiger gesetzt und es wird kein Nachrichtendesktors an den Exit weitergeleitet:
 - `MQXR_INIT`

- MQXR_TERM
- MQXR_CLWL_OPEN

MsgBufferPtr (PMQVOID) - Eingabe

Die Adresse eines Puffers, der eine Kopie der ersten MsgBufferLength-Bytes der Nachrichtendaten enthält.

- Alle Änderungen an den Nachrichtendaten durch den Exit werden vom Warteschlangenmanager ignoriert.
- Es werden keine Nachrichtendaten an den Exit weitergeleitet, wenn:
 - MsgDescPtr der Nullzeiger ist.
 - Die Nachricht keine Daten enthält.
 - Das Warteschlangenmanagerattribut ClusterWorkloadLength 0 ist.

In diesen Fällen ist MsgBufferPtr der Nullzeiger.

MsgBufferLength (MQLONG) - Eingabe

Die Länge des Puffers, der die an den Exit weitergeleiteten Nachrichtendaten enthält.

- Die Länge wird durch das Warteschlangenmanagerattribut ClusterWorkloadLength bestimmt.
- Die Länge kann kleiner als die Länge der vollständigen Nachricht sein, vgl. MsgLength.

MsgLength (MQLONG) - Eingabe

Die Länge der an den Exit weitergeleiteten vollständigen Nachricht.

- MsgBufferLength kann kleiner als die Länge der vollständigen Nachricht sein.
- MsgLength ist 0, wenn ExitReason den Wert MQXR_INIT, MQXR_TERM oder MQXR_CLWL_OPEN hat.

QName (MQCHAR48) - Eingabe

Der Name der Zielwarteschlange. Die Warteschlange ist eine Clusterwarteschlange.

- Die Länge von QName ist MQ_Q_NAME_LENGTH.

QMgrName (MQCHAR48) - Eingabe

Der Name des lokalen Warteschlangenmanagers, der den Exit für Clusterauslastung aufgerufen hat.

- Die Länge von QMgrName ist MQ_Q_MGR_NAME_LENGTH.

DestinationCount (MQLONG) - Eingabe

Die Anzahl möglicher Zieladressen. Zieladressen sind Instanzen der Zielwarteschlange und werden durch Zieldatensätze beschrieben.

- Ein Zieldatensatz ist eine MQWDR-Struktur. Es gibt eine Struktur für jede mögliche Route zu den einzelnen Instanzen der Warteschlange.
- MQWDR-Strukturen werden durch einen Zeigerbereich adressiert, vgl. DestinationArrayPtr.

DestinationChosen (MQLONG) - Ein-/Ausgabe

Die ausgewählte Zieladresse.

- Die Zahl der MQWDR-Struktur, die Route und Warteschlangeninstanz identifiziert, an die die Nachricht gesendet werden soll.
- Der Wert liegt im Bereich von 1 bis DestinationCount.
- Bei der Eingabe für den Exit zeigt DestinationChosen die Route und Warteschlangeninstanz an, die der Warteschlangenmanager ausgewählt hat. Der Exit kann diese Auswahl annehmen oder eine andere Route und Warteschlangeninstanz auswählen.
- Der vom Exit festgelegte Wert muss im Bereich von 1 bis DestinationCount liegen. Wenn ein anderer Wert zurückgegeben wird, verwendet der Warteschlangenmanager den Wert DestinationChosen bei der Eingabe für den Exit.

DestinationArrayPtr (PPMQWDR) - Eingabe

Die Adresse eines Bereichs von Zeigern auf Zieldatensätze (MQWDR).

- Es gibt DestinationCount Zieldatensätze.

QArrayPtr (PPMQWQR) - Eingabe

Die Adresse eines Bereichs von Zeigern auf Warteschlangendatensätze (MQWQR).

- Wenn Warteschlangendatensätze verfügbar sind, gibt es DestinationCount davon.
- Sollten keine Warteschlangendatensätze verfügbar sein, ist QArrayPtr der Nullzeiger.

Anmerkung: QArrayPtr kann der Nullzeiger sein, auch wenn DestinationCount größer als 0 ist.

CacheContext (MQPTR): Version 2 - Eingabe

Das CacheContext-Feld ist für die Verwendung durch den Warteschlangenmanager reserviert. Der Exit darf den Wert dieses Felds nicht ändern.

CacheType (MQLONG): Version 2 - Eingabe

Der Cluster-Cache ist einer der folgenden Typen:

MQCLCT_STATIC

Der Cache ist statisch.

- Die Größe des Cache ist festgelegt und kann nicht erweitert werden, während der Warteschlangenmanager ausgeführt wird.
- Sie müssen nicht den Aufruf MQXCLWLN verwenden, um die Datensätze in diesem Cachetyp anzusteuern.

MQCLCT_DYNAMIC

Der Cache ist dynamisch.

- Die Größe des Cache kann erweitert werden, um die unterschiedlichen Clusterinformationen aufzunehmen.
- Sie müssen den Aufruf MQXCLWLN verwenden, um die Datensätze in diesem Cachetyp anzusteuern.

CLWLMRUChannels (MQLONG): Version 3 - Eingabe

Zeigt die maximale Anzahl aktiver abgehender Clusterkanäle an, die für die Verwendung durch den Algorithmus zur Auswahl der Clusterauslastung berücksichtigt werden muss.

- CLWLMRUChannels ist ein Wert von 1 bis 999 999 999.

pEntryPoints (PMQIEP): Version 4

Die Adresse einer MQIEP-Struktur, über die MQI- und DCI-Aufrufe möglich sind.

Anfangswerte und Sprachendeklarationen für MQWXP

Anfangswerte und Deklarationen in den Programmiersprachen C und High Level Assembler für die Parameterstruktur des Exits für Clusterauslastung MQWXP.

<i>Tabelle 25. Anfangswerte der Felder in MQWXP</i>		
Name des Felds	Name der Konstante	Wert der Konstanten
<i>StrucId</i>	MQWXP_STRUC_ID	'WXP? '
<i>Version</i>	MQWXP_VERSION_2	2
<i>ExitId</i>	--	0
<i>ExitReason</i>	MQXCC_OK	0
<i>ExitResponse</i>	--	0
<i>ExitResponse2</i>	--	0
<i>Flags</i>	--	0
<i>ExitUserArea</i>	{MQXUA_NONE_ARRAY}	0
<i>ExitData</i>	--	" "

Tabelle 25. Anfangswerte der Felder in MQWXP (Forts.)

Name des Felds	Name der Konstante	Wert der Konstanten
<i>MsgDescPtr</i>	--	NULL
<i>MsgBufferPtr</i>	--	NULL
<i>MsgBufferLength</i>	--	0
<i>MsgBufferPtr</i>	--	0
<i>QName</i>	--	" "
<i>QMgrName</i>	--	" "
<i>DestinationCount</i>	--	0
<i>DestinationChosen</i>	--	0
<i>DestinationArrayPtr</i>	--	NULL
<i>QArrayPtr</i>	--	NULL
<i>CacheContext</i>	--	NULL
<i>CacheType</i>	MQCLCT_DYNAMIC	1
<i>CLWLMRUChannels</i>	--	0
<i>pEntryPoints</i>	--	NULL

Anmerkungen:

1. Das Symbol ? stellt ein einzelnes Leerzeichen dar.
2. In der Programmiersprache C enthält die Makrovariable MQWXP_DEFAULT die Standardwerte. Verwenden Sie sie folgendermaßen, um Anfangswerte für die Felder in der Struktur anzugeben:

```
MQWDR MyWXP = {MQWXP_DEFAULT};
```

Deklaration in Programmiersprache C

```
typedef struct tagMQWXP {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    ExitId;           /* Type of exit */
    MQLONG    ExitReason;       /* Reason for invoking exit */
    MQLONG    ExitResponse;     /* Response from exit */
    MQLONG    ExitResponse2;    /* Reserved */
    MQLONG    Feedback;         /* Reserved */
    MQLONG    Flags;            /* Flags */
    MQBYTE16  ExitUserArea;     /* Exit user area */
    MQCHAR32  ExitData;         /* Exit data */
    PMQMD     MsgDescPtr;       /* Address of message descriptor */
    PMQVOID   MsgBufferPtr;     /* Address of buffer containing some
                                or all of the message data */
    MQLONG    MsgBufferLength;  /* Length of buffer containing message
                                data */
    MQLONG    MsgLength;        /* Length of complete message */
    MQCHAR48  QName;           /* Queue name */
    MQCHAR48  QMgrName;        /* Name of local queue manager */
    MQLONG    DestinationCount; /* Number of possible destinations */
    MQLONG    DestinationChosen; /* Destination chosen */
    PPMQWDR   DestinationArrayPtr; /* Address of an array of pointers to
                                destination records */
    PPMQWQR   QArrayPtr;       /* Address of an array of pointers to
                                queue records */

    /* version 1 */
    MQPTR     CacheContext;     /* Context information */
    MQLONG    CacheType;       /* Type of cluster cache */
};
```

```

/* version 2 */
MQLONG    CLWLMRUChannels;    /* Maximum number of most recently
                               used cluster channels */

/* version 3 */
PMQIEP    pEntryPoints;      /* Address of the MQIEP structure */
/* version 4 */
};

```

High Level Assembler

```

MQWXP                                DSECT
MQWXP_STRUCID                        DS    CL4      Structure identifier
MQWXP_VERSION                        DS    F        Structure version number
MQWXP_EXITID                         DS    F        Type of exit
MQWXP_EXITREASON                     DS    F        Reason for invoking exit
MQWXP_EXITRESPONSE                   DS    F        Response from exit
MQWXP_EXITRESPONSE2                 DS    F        Reserved
MQWXP_FEEDBACK                       DS    F        Reserved
MQWXP_RESERVED                       DS    F        Reserved
MQWXP_EXITUSERAREA                   DS    XL16     Exit user area
MQWXP_EXITDATA                       DS    CL32     Exit data
MQWXP_MSGDESCPTR                     DS    F        Address of message
*                                     descriptor
MQWXP_MSGBUFFERPTR                   DS    F        Address of buffer containing
*                                     some or all of the message
*                                     data
MQWXP_MSGBUFFERLENGTH                DS    F        Length of buffer containing
*                                     message data
MQWXP_MSGLENGTH                       DS    F        Length of complete message
MQWXP_QNAME                           DS    CL48     Queue name
MQWXP_QMGRNAME                        DS    CL48     Name of local queue manager
MQWXP_DESTINATIONCOUNT              DS    F        Number of possible
*                                     destinations
MQWXP_DESTINATIONCHOSEN               DS    F        Destination chosen
MQWXP_DESTINATIONARRAYPTR            DS    F        Address of an array of
*                                     pointers to destination
*                                     records
MQWXP_QARRAYPTR                       DS    F        Address of an array of
*                                     pointers to queue records
MQWXP_CACHECONTEXT                   DS    F        Context information
MQWXP_CACHETYPE                       DS    F        Type of cluster cache
MQWXP_CLWLMRUCHANNELS                DS    F        Number of most recently used
*                                     channels for workload balancing

MQWXP_LENGTH                          EQU    *-MQWXP  Length of structure
MQWXP_AREA                            ORG    MQWXP
MQWXP_AREA                            DS    CL(MQWXP_LENGTH)

```

MQWDR - Zieldatensatzstruktur für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Zieldatensatzstruktur für Clusterauslastung MQWDR.

Tabelle 26. Felder in MQWDR		
Feld	Beschreibung	Seite
<i>StrucId</i>	Struktur-ID	StrucId
<i>Version</i>	Strukturversionsnummer	Version
<i>StrucLength</i>	Länge der MQWDR-Struktur	StrucLength
<i>QMgrFlags</i>	Warteschlangenmanagerflags	QMgrFlags
<i>QMgrIdentifizier</i>	Warteschlangenmanager-ID	QMgrIdentifizier
<i>QMgrName</i>	Name des Warteschlangenmanagers	QMgrName
<i>ClusterRecOffset</i>	Logische relative Adresse des ersten Clusterdatensatzes (MQWCR)	ClusterRecOffset
<i>ChannelState</i>	Kanalstatus	ChannelState

Tabelle 26. Felder in MQWDR (Forts.)		
Feld	Beschreibung	Seite
<i>ChannelDefOffset</i>	Logische relative Adresse der Kanaldefinitionsstruktur (MQCD)	ChannelDefOffset
Anmerkung: Die übrigen Felder werden ignoriert, wenn 'Version' kleiner als MQWDR_VERSION_2 ist.		
<i>DestSeqNumber</i>	Folgenummer der Kanalzieladresse	DestSeqNumber
<i>DestSeqFactor</i>	Folgefaktor der Kanalzieladresse für Gewichtung	DestSeqFactor

Die Zieldatensatzstruktur für Clusterauslastung enthält Daten zu einer der möglichen Zieladressen für die Nachricht. Jeder Instanz der Zielwarteschlange ist eine Zieldatensatzstruktur für Clusterauslastung zugeordnet.

Die Zieldatensatzstruktur für Clusterauslastung wird in allen Umgebungen unterstützt.

Zusätzlich sind die MQWDR1- und MQWDR2-Strukturen für Abwärtskompatibilität verfügbar.

Zugehörige Verweise

[MQ_CLUSTER_WORKLOAD_EXIT](#) - Beschreibung des Aufrufs

Der Exit für Clusterauslastung wird vom Warteschlangenmanager zur Weiterleitung einer Nachricht an einen verfügbaren Warteschlangenmanager aufgerufen.

[MQXCLWLN](#) - Navigieren in den Clusterauslastungsdatensätzen

Der Aufruf MQXCLWLN wird verwendet, um durch die im Cluster-Cache gespeicherten Ketten der MQWDR-, MQWQR- und MQWCR -Datensätze zu navigieren.

[MQWXP](#) - Parameterstruktur des Exits für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP.

[MQWQR](#) - Struktur des Warteschlangendatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

[MQWCR](#) - Struktur des Clusterdatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Felder in der Zieldatensatzstruktur für Clusterauslastung MQWDR

Beschreibung der Parameter in der Zieldatensatzstruktur für Clusterauslastung MQWDR.

StrucId (MQCHAR4) - Eingabe

Die Struktur-ID für die Zieldatensatzstruktur für Clusterauslastung.

- Der Wert für StrucId ist MQWDR_STRUC_ID.
- Für die Programmiersprache C wird die Konstante MQWDR_STRUC_ID_ARRAY ebenfalls definiert. Der Wert entspricht MQWDR_STRUC_ID. Es handelt sich um eine Gruppe von Zeichen, nicht um eine Zeichenfolge.

Version (MQLONG) - Eingabe

Die Versionsnummer der Struktur. Version hat einen der folgenden Werte:

MQWDR_VERSION_1

Zieldatensatz für Clusterauslastung Version 1.

MQWDR_VERSION_2

Zieldatensatz für Clusterauslastung Version 2.

MQWDR_CURRENT_VERSION

Aktuelle Version des Zieldatensatzes für Clusterauslastung.

StrucLength (MQLONG)-Eingabe

Die Länge der MQWDR-Struktur. StrucLength hat einen der folgenden Werte:

MQWDR_LENGTH_1

Länge des Zieldatensatzes für Clusterauslastung Version 1.

MQWDR_LENGTH_2

Länge des Zieldatensatzes für Clusterauslastung Version 2.

MQWDR_CURRENT_LENGTH

Länge der aktuellen Version des Zieldatensatzes für Clusterauslastung.

QMgrFlags (MQLONG)-Eingabe

Warteschlangenmanagerflags, die Eigenschaften des Warteschlangenmanagers anzeigen, der die in der MQWDR-Struktur beschriebene Instanz der Zielwarteschlange enthält. Die folgenden Flags sind definiert:

MQQMF_REPOSITORY_Q_MGR

Die Zieladresse ist ein Warteschlangenmanager für ein vollständiges Repository.

MQQMF_CLUSSDR_USER_DEFINED

Der Clustersenderkanal wurde manuell definiert.

MQQMF_CLUSSDR_AUTO_DEFINED

Der Clustersenderkanal wurde automatisch definiert.

MQQMF_AVAILABLE

Der Zielwarteschlangenmanager ist für das Empfangen von Nachrichten verfügbar.

Other values

Andere Flags in diesem Feld können vom Warteschlangenmanager für interne Zwecke gesetzt werden.

QMgrIdentifizier (MQCHAR48) - Eingabe

Die Warteschlangenmanager-ID ist eine eindeutige ID für den Warteschlangenmanager, der die in der MQWDR-Struktur beschriebene Instanz der Zielwarteschlange enthält.

- Die ID wird vom Warteschlangenmanager generiert.
- Die Länge von QMgrIdentifizier wird in MQ_Q_MGR_IDENTIFIER_LENGTH angegeben.

QMgrName (MQCHAR48) - Eingabe

Der Name des Warteschlangenmanagers, der die in der MQWDR-Struktur beschriebene Instanz der Zielwarteschlange enthält.

- QMgrName kann der Name des lokalen Warteschlangenmanagers oder eines anderen Warteschlangenmanagers in dem Cluster sein.
- Die Länge von QMgrName ist MQ_Q_MGR_NAME_LENGTH.

ClusterRecOffset (MQLONG)-Eingabe

Die logische relative Adresse der ersten MQWCR-Struktur, die zu der MQWDR-Struktur gehört.

- Für statische Caches ist ClusterRecOffset die relative Adresse der ersten MQWCR-Struktur, die zu der MQWDR-Struktur gehört.
- Die relative Adresse wird vom Start der MQWDR-Struktur an in Bytes gemessen.
- Verwenden Sie die logische relative Adresse nicht für Zeigerarithmetik mit dynamischen Caches. Um die Adresse des nächsten Datensatzes anzufordern, muss der MQXCLWLN-Aufruf verwendet werden.

ChannelState (MQLONG)-Eingabe

Der Status des Kanals, der den lokalen Warteschlangenmanager mit dem durch die MQWDR-Struktur bestimmten Warteschlangenmanager verbindet. Folgende Werte sind möglich:

MQCHS_BINDING

Kanal trifft eine Vereinbarung mit dem Partner.

MQCHS_INACTIVE

Kanal ist nicht aktiv.

MQCHS_INITIALIZING

Kanal führt Initialisierung durch.

MQCHS_PAUSED

Kanal wurde angehalten.

MQCHS_REQUESTING

Requesterkanal fordert Verbindung an.

MQCHS_RETRYING

Kanal versucht erneut, eine Verbindung herzustellen.

MQCHS_RUNNING

Der Kanal überträgt Nachrichten oder wartet auf diese.

MQCHS_STARTING

Kanal wartet auf seine Aktivierung.

MQCHS_STOPPING

Kanal wird gestoppt.

MQCHS_STOPPED

Kanal wurde gestoppt.

ChannelDefOffset (MQLONG)-Eingabe

Die logische relative Adresse der Kanaldefinition (MQCD) für den Kanal, der den lokalen Warteschlangenmanager mit dem durch die MQWDR-Struktur bestimmten Warteschlangenmanager verbindet.

- ChannelDefOffset ist wie ClusterRecOffset.
- Die logische relative Adresse kann nicht in Zeigerarithmetik verwendet werden. Um die Adresse des nächsten Datensatzes anzufordern, muss der MQXCLWLN-Aufruf verwendet werden.

DestSeqFaktor (MQLONG)-Eingabe

Der Zieladressenfolgefaktor, der eine auf Gewichtung basierende Auswahl des Kanals ermöglicht.

- DestSeqFactor wird verwendet, bevor der Warteschlangenmanager Änderungen daran vornimmt.
- Der Auslastungsmanager erhöht DestSeqFactor in einer Art und Weise, die sicherstellt, dass die Nachrichten entsprechend ihrer Gewichtung auf die Kanäle verteilt werden.

DestSeqZahl (MQLONG)-Eingabe

Der Zielwert des Clusterkanals vor Änderung durch den Warteschlangenmanager.

- Der Auslastungsmanager erhöht DestSeqNumber jedes Mal, wenn eine Nachricht an diesen Kanal übergeben wird.
- Auslastungsexits können DestSeqNumber verwenden, um zu entscheiden, an welchen Kanal eine Nachricht übergeben werden soll.

Anfangswerte und Sprachdeklarationen für MQWDR

Anfangswerte und Deklarationen in den Programmiersprachen C und High Level Assembler für den Ziel-datensatz für Clusterauslastung MQWDR.

Tabelle 27. Anfangswerte der Felder in MQWDR		
Name des Felds	Name der Konstante	Wert der Konstanten
<i>StrucId</i>	MQWDR_STRUC_ID	'WDR? '
<i>Version</i>	MQWDR_VERSION_1	1
<i>StrucLength</i>	MQWDR_CURRENT_LENGTH ³	136
<i>QMgrFlags</i>	MQWDR_NONE	0
<i>QMgrIdentifizier</i>	--	""
<i>QMgrName</i>	--	""
<i>ClusterRecOffset</i>	--	0

Tabelle 27. Anfangswerte der Felder in MQWDR (Forts.)

Name des Felds	Name der Konstante	Wert der Konstanten
<i>ChannelState</i>	--	0
<i>ChannelDefOffset</i>	--	0
<i>DestSeqNumber</i>	--	0
<i>DestSeqFactor</i>	--	0

Anmerkungen:

1. Das Symbol ? stellt ein einzelnes Leerzeichen dar.
2. In der Programmiersprache C enthält die Makrovariable MQWDR_DEFAULT die Standardwerte. Verwenden Sie sie folgendermaßen, um Anfangswerte für die Felder in der Struktur anzugeben:

```
MQWDR MyWDR = {MQWDR_DEFAULT};
```

3. Die Anfangswerte legen die Länge der Struktur absichtlich auf die Länge der aktuellen Version, und nicht auf die Länge von Version 1 fest.

High Level Assembler

```
MQWDR          DSECT
MQWDR_STRUCID  DS    CL4      Structure identifier
MQWDR_VERSION DS    F        Structure version number
MQWDR_STRUCLNTH DS    F        Length of MQWDR structure
MQWDR_QMGRFLGS DS    F        Queue-manager flags
MQWDR_QMGRIDENTIFIER DS CL48  Queue-manager identifier
MQWDR_QMGRNAME DS    CL48  Queue-manager name
MQWDR_CLUSTERRECOFFSET DS    F        Offset of first cluster
*              record
MQWDR_CHANNELSTATE DS    F        Channel state
MQWDR_CHANNELDEFOFFSET DS    F        Offset of channel definition
*              structure
MQWDR_LENGTH   EQU    *-MQWDR Length of structure
MQWDR_AREA     ORG    MQWDR
               DS    CL(MQWDR_LENGTH)
```

Deklaration in Programmiersprache C

```
typedef struct tagMQWDR {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWDR structure */
    MQLONG    QMgrFlags;        /* Queue-manager flags */
    MQCHAR48  QMgrIdentifier;    /* Queue-manager identifier */
    MQCHAR48  QMgrName;         /* Queue-manager name */
    MQLONG    ClusterRecOffset; /* Offset of first cluster record */
    MQLONG    ChannelState;     /* Channel state */
    MQLONG    ChannelDefOffset; /* Offset of channel definition structure */
    /* Ver:1 */
    MQLONG    DestSeqNumber;     /* Cluster channel destination sequence number */
    MQINT64   DestSeqFactor;     /* Cluster channel factor sequence number */
    /* Ver:2 */
};
```

MQWQR - Struktur des Warteschlangendatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

<i>Tabelle 28. Felder in MQWQR</i>		
Feld	Beschreibung	Seite
<i>StrucId</i>	Struktur-ID	StrucId
<i>Version</i>	Strukturversionsnummer	Version
<i>StrucLength</i>	Länge der MQWQR-Struktur	StrucLength
<i>QFlags</i>	Warteschlangenflags	QFlags
<i>QName</i>	Warteschlangenname	QName
<i>QMgrIdentifizier</i>	Warteschlangenmanager-ID	QMgrIdentifizier
<i>ClusterRecOffset</i>	Relative Adresse des ersten Clusterdatensatzes (MQWCR)	ClusterRecOffset
<i>QType</i>	Warteschlangentyp	QType
<i>QDesc</i>	Warteschlangenbeschreibung	QDesc
<i>DefBind</i>	Standardbindung	DefBind
<i>DefPersistence</i>	Standardpersistenz für Nachrichten	DefPersistence
<i>DefPriority</i>	Standardpriorität für Nachr.	DefPriority
<i>InhibitPut</i>	Gibt an, ob PUT-Operationen für diese Warteschlange zulässig sind	InhibitPut
Anmerkung: Die übrigen Felder werden ignoriert, wenn 'Version' kleiner als MQWQR_VERSION_2 ist.		
<i>CLWLQueuePriority</i>	Ein Wert von 0 bis 9, der die Priorität der Warteschlange darstellt	CLWLQueuePriority
<i>CLWLQueueRank</i>	Ein Wert von 0 bis 9, der den Rang der Warteschlange darstellt	CLWLQueueRank
Anmerkung: Die übrigen Felder werden ignoriert, wenn 'Version' kleiner als MQWQR_VERSION_3 ist.		
<i>DefPutResponse</i>	Standard-PUT-Antwort	DefPutResponse

Die Struktur des Warteschlangendatensatzes für Clusterauslastung enthält Daten zu einer der möglichen Zieladressen für die Nachricht. Jeder Instanz der Zielwarteschlange ist eine Struktur des Warteschlangendatensatzes für Clusterauslastung zugeordnet.

Die Struktur des Warteschlangendatensatzes für Clusterauslastung wird in allen Umgebungen unterstützt.

Zusätzlich sind die MQWQR1- und MQWQR2-Strukturen für Abwärtskompatibilität verfügbar.

Zugehörige Verweise

[MQ_CLUSTER_WORKLOAD_EXIT](#) - Beschreibung des Aufrufs

Der Exit für Clusterauslastung wird vom Warteschlangenmanager zur Weiterleitung einer Nachricht an einen verfügbaren Warteschlangenmanager aufgerufen.

[MQXCLWLN](#) - Navigieren in den Clusterauslastungsdatensätzen

Der Aufruf MQXCLWLN wird verwendet, um durch die im Cluster-Cache gespeicherten Ketten der MQWDR-, MQWQR- und MQWCR -Datensätze zu navigieren.

[MQWXP](#) - Parameterstruktur des Exits für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP.

[MQWDR](#) - Zieldatensatzstruktur für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Zieldatensatzstruktur für Cluster-
auslastung MQWDR.

MQWCR - Struktur des Clusterdatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Clusterdatensatzes für
Clusterauslastung MQWCR.

Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR

Beschreibung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

StrucId (MQCHAR4) - Eingabe

Die Struktur-ID für die Struktur des Warteschlangendatensatzes für Clusterauslastung.

- Der Wert für StrucId ist MQWQR_STRUC_ID.
- Für die Programmiersprache C wird die Konstante MQWQR_STRUC_ID_ARRAY ebenfalls definiert. Der Wert entspricht MQWQR_STRUC_ID. Es handelt sich um eine Gruppe von Zeichen, nicht um eine Zeichenfolge.

Version (MQLONG) - Eingabe

Die Versionsnummer der Struktur. Version hat einen der folgenden Werte:

MQWQR_VERSION_1

Warteschlangendatensatz für Clusterauslastung Version 1.

MQWQR_VERSION_2

Warteschlangendatensatz für Clusterauslastung Version 2.

MQWQR_VERSION_3

Warteschlangendatensatz für Clusterauslastung Version 3.

MQWQR_CURRENT_VERSION

Aktuelle Version des Warteschlangendatensatzes für Clusterauslastung.

StrucLength (MQLONG)-Eingabe

Die Länge der MQWQR-Struktur. StrucLength hat einen der folgenden Werte:

MQWQR_LENGTH_1

Länge des Warteschlangendatensatzes für Clusterauslastung Version 1.

MQWQR_LENGTH_2

Länge des Warteschlangendatensatzes für Clusterauslastung Version 2.

MQWQR_LENGTH_3

Länge des Warteschlangendatensatzes für Clusterauslastung Version 3.

MQWQR_CURRENT_LENGTH

Länge der aktuellen Version des Warteschlangendatensatzes für Clusterauslastung.

QFlags (MQLONG)-Eingabe

Die Warteschlangenflags geben Eigenschaften der Warteschlange an. Die folgenden Flags sind definiert:

MQQF_LOCAL_Q

Die Zieladresse ist eine lokale Warteschlange.

MQQF_CLWL_USEQ_ANY

Verwendung von lokalen und fernen Warteschlangen beim Einreihen erlauben.

MQQF_CLWL_USEQ_LOCAL

Nur Einreihungen in eine lokale Warteschlange erlauben.

Other values

Andere Flags in diesem Feld können vom Warteschlangenmanager für interne Zwecke gesetzt werden.

QName (MQCHAR48) - Eingabe

Der Name einer Warteschlange, die eine der möglichen Zieladressen der Nachricht ist.

- Die Länge von QName ist MQ_Q_NAME_LENGTH.

QMgrIdentifizier (MQCHAR48) - Eingabe

Die Warteschlangenmanager-ID ist eine eindeutige ID für den Warteschlangenmanager, der die in der MQWQR-Struktur beschriebene Instanz der Warteschlange enthält.

- Die ID wird vom Warteschlangenmanager generiert.
- Die Länge von QMgrIdentifizier wird in MQ_Q_MGR_IDENTIFIER_LENGTH angegeben.

ClusterRecOffset (MQLONG)-Eingabe

Die logische relative Adresse der ersten MQWCR-Struktur, die zu der MQWQR-Struktur gehört.

- Für statische Caches ist ClusterRecOffset die relative Adresse der ersten MQWCR-Struktur, die zu der MQWQR-Struktur gehört.
- Die relative Adresse wird vom Start der MQWQR-Struktur an in Bytes gemessen.
- Verwenden Sie die logische relative Adresse nicht für Zeigerarithmetik mit dynamischen Caches. Um die Adresse des nächsten Datensatzes anzufordern, muss der MQXCLWLN-Aufruf verwendet werden.

QType (MQLONG)-Eingabe

Der Warteschlangentyp der Zielwarteschlange. Folgende Werte sind möglich:

MQCQT_LOCAL_Q

Lokale Warteschlange.

MQCQT_ALIAS_Q

Aliaswarteschlange.

MQCQT_REMOTE_Q

Ferne Warteschlange

MQCQT_Q_MGR_ALIAS

Aliasname des Warteschlangenmanagers.

QDesc (MQCHAR64) - Eingabe

Das Warteschlangenattribut zur Warteschlangenbeschreibung, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält.

- Die Länge von QDesc wird in MQ_Q_DESC_LENGTH angegeben.

DefBind (MQLONG)-Eingabe

Das Warteschlangenattribut zur Standardbindung, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält. Bei der Verwendung von Gruppen mit Clustern muss entweder MQBND_BIND_ON_OPEN oder MQBND_BIND_ON_GROUP angegeben werden. Folgende Werte sind möglich:

MQBND_BIND_ON_OPEN

Bindung festgelegt durch MQOPEN-Aufruf.

MQBND_BIND_NOT_FIXED

Bindung nicht festgelegt.

MQBND_BIND_ON_GROUP

Mit dieser Option kann eine Anwendung fordern, dass alle Nachrichten einer Nachrichtengruppe an dieselbe Zielinstanz übergeben werden.

DefPersistence (MQLONG)-Eingabe

Das Warteschlangenattribut zur Standardnachrichtenpersistenz, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält. Folgende Werte sind möglich:

MQPER_PERSISTENT

Nachricht ist persistent

MQPER_NOT_PERSISTENT

Nachricht ist nicht persistent

DefPriority (MQLONG)-Eingabe

Das Warteschlangenattribut zur Standardnachrichtepriorität, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält. Der Prioritätsbereich umfasst 0 - MaxPriority.

- 0 ist die niedrigste Priorität.
- MaxPriority ist das Warteschlangenmanagerattribut des Warteschlangenmanagers, der diese Instanz der Zielwarteschlange enthält.

InhibitPut (MQLONG)-Eingabe

Das Warteschlangenattribut zum Sperren von PUT-Operationen, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält. Folgende Werte sind möglich:

MQQA_PUT_INHIBITED

Put-Operationen werden unterdrückt.

MQQA_PUT_ALLOWED

PUT-Operationen werden zugelassen.

CLWLQueuePriority (MQLONG)-Eingabe

Das Attribut zur Warteschlangenpriorität bei der Clusterauslastung, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält.

CLWLQueueRank (MQLONG)-Eingabe

Der Warteschlangenrang bei der Clusterauslastung, der in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält.

DefPutResponse (MQLONG)-Eingabe

Das Warteschlangenattribut zur Standard-PUT-Antwort, das in dem Warteschlangenmanager definiert ist, der die in der MQWQR-Struktur beschriebene Instanz der Zielwarteschlange enthält. Folgende Werte sind möglich:

MQPRT_SYNC_RESPONSE

Synchrone Antwort auf MQPUT- oder MQPUT1-Aufrufe.

MQPRT_ASYNC_RESPONSE

Asynchrone Antwort auf MQPUT- oder MQPUT1-Aufrufe.

Anfangswerte und Sprachendeklarationen für MQWQR

Anfangswerte und Deklarationen in den Programmiersprachen C und High Level Assembler für den Warteschlangendatensatz für Clusterauslastung MQWQR.

Tabelle 29. Anfangswerte der Felder in MQWQR		
Name des Felds	Name der Konstante	Wert der Konstanten
<i>StrucId</i>	MQWQR_STRUC_ID_ARRAY	'WQR? '
<i>Version</i>	MQWQR_VERSION_1	1
<i>StrucLength</i>	MQWQR_CURRENT_LENGTH ³	212
<i>QFlags</i>	--	0
<i>QName</i>	--	" "
<i>QMgrIdentifizier</i>	--	" "
<i>ClusterRecOffset</i>	--	0
<i>QType</i>	--	0
<i>QDesc</i>	--	" "
<i>DefBind</i>	--	0

Tabelle 29. Anfangswerte der Felder in MQWQR (Forts.)

Name des Felds	Name der Konstante	Wert der Konstanten
DefPersistence	--	0
DefPriority	--	0
InhibitPut	--	0
CLWLQueuePriority	--	0
CLWLQueueRank	--	0
DefPutResponse	--	1

Anmerkungen:

1. Das Symbol ? stellt ein einzelnes Leerzeichen dar.
2. In der Programmiersprache C enthält die Makrovariable MQWQR_DEFAULT die Standardwerte. Verwenden Sie sie folgendermaßen, um Anfangswerte für die Felder in der Struktur anzugeben:

```
MQWQR MyWQR = {MQWQR_DEFAULT};
```

3. Die Anfangswerte legen die Länge der Struktur absichtlich auf die Länge der aktuellen Version, und nicht auf die Länge von Version 1 fest.

Deklaration in Programmiersprache C

```
typedef struct tagMQWQR {
    MQCHAR4   StrucId;           /* Structure identifier */
    MQLONG    Version;          /* Structure version number */
    MQLONG    StrucLength;      /* Length of MQWQR structure */
    MQLONG    QFlags;           /* Queue flags */
    MQCHAR48  QName;            /* Queue name */
    MQCHAR48  QMgrIdentifier;    /* Queue-manager identifier */
    MQLONG    ClusterRecOffset; /* Offset of first cluster record */
    MQLONG    QType;            /* Queue type */
    MQCHAR64  QDesc;            /* Queue description */
    MQLONG    DefBind;          /* Default binding */
    MQLONG    DefPersistence;    /* Default message persistence */
    MQLONG    DefPriority;       /* Default message priority */
    MQLONG    InhibitPut;       /* Whether put operations on the queue
                                are allowed */

    /* version 2 */
    MQLONG    CLWLQueuePriority; /* Queue priority */
    MQLONG    CLWLQueueRank;     /* Queue rank */
    /* version 3 */
    MQLONG    DefPutResponse;    /* Default put response */
};
```

High Level Assembler

```
MQWQR          DSECT
MQWQR_STRUCID  DS    CL4      Structure identifier
MQWQR_VERSION  DS    F        Structure version number
MQWQR_STRUCLNGTH DS    F        Length of MQWQR structure
MQWQR_QFLAGS   DS    F        Queue flags
MQWQR_QNAME    DS    CL48     Queue name
MQWQR_QMGRIDENTIFIER DS    CL48 Queue-manager identifier
MQWQR_CLUSTERRECOFFSET DS    F Offset of first cluster
*              record
MQWQR_QTYPE    DS    F        Queue type
MQWQR_QDESC    DS    CL64     Queue description
MQWQR_DEFBIND  DS    F        Default binding
MQWQR_DEFPERSISTENCE DS    F   Default message persistence
MQWQR_DEFPRIORITY DS    F     Default message priority
MQWQR_INHIBITPUT DS    F     Whether put operations on
*              the queue are allowed
MQWQR_DEFPUTRESPONSE DS    F   Default put response
MQWQR_LENGTH   EQU    *-MQWQR Length of structure
```

MQWCR - Struktur des Clusterdatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Tabelle 30. Felder in MQWCR		
Feld	Beschreibung	Seite
<i>ClusterName</i>	Name des Clusters	ClusterName
<i>ClusterRecOffset</i>	Relative Adresse des nächsten Clusterdatensatzes (MQWCR)	ClusterRecOffset
<i>ClusterFlags</i>	Clusterflags	ClusterFlags

Die Struktur des Clusterdatensatzes für Clusterauslastung enthält Daten zu einem Cluster. Für jeden Cluster, zu dem die Zielwarteschlange gehört, gibt es eine Struktur des Clusterdatensatzes für Clusterauslastung.

Die Struktur des Clusterdatensatzes für Clusterauslastung wird in allen Umgebungen unterstützt.

Zugehörige Verweise

[MQ_CLUSTER_WORKLOAD_EXIT](#) - Beschreibung des Aufrufs

Der Exit für Clusterauslastung wird vom Warteschlangenmanager zur Weiterleitung einer Nachricht an einen verfügbaren Warteschlangenmanager aufgerufen.

[MQXCLWLN](#) - Navigieren in den Clusterauslastungsdatensätzen

Der Aufruf MQXCLWLN wird verwendet, um durch die im Cluster-Cache gespeicherten Ketten der MQWDR-, MQWQR- und MQWCR -Datensätze zu navigieren.

[MQWXP](#) - Parameterstruktur des Exits für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Parameterstruktur des Exits für Clusterauslastung MQWXP.

[MQWDR](#) - Zieldatensatzstruktur für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Zieldatensatzstruktur für Clusterauslastung MQWDR.

[MQWQR](#) - Struktur des Warteschlangendatensatzes für Clusterauslastung

Die folgende Tabelle enthält eine Zusammenfassung der Felder in der Struktur des Warteschlangendatensatzes für Clusterauslastung MQWQR.

Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Beschreibung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

ClusterName (MQCHAR48) - Eingabe

Der Name des Clusters, zu dem die Instanz der Zielwarteschlange der MQWCR-Struktur gehört. Die Instanz der Zielwarteschlange wird in einer MQWDR-Struktur beschrieben.

- Die Länge von ClusterName wird in MQ_CLUSTER_NAME_LENGTH angegeben.

ClusterRecOffset (MQLONG)-Eingabe

Die logische relative Adresse der nächsten MQWCR-Struktur.

- Wenn es keine weiteren MQWCR-Strukturen gibt, hat ClusterRecOffset den Wert null.
- Die relative Adresse wird vom Start der MQWCR-Struktur an in Bytes gemessen.

ClusterFlags (MQLONG)-Eingabe

Die Clusterflags geben Eigenschaften des in der MQWCR-Struktur bestimmten Warteschlangenmanagers an. Die folgenden Flags sind definiert:

MQQMF_REPOSITORY_Q_MGR

Die Zieladresse ist ein Warteschlangenmanager für ein vollständiges Repository.

MQQMF_CLUSSDR_USER_DEFINED

Der Clustersenderkanal wurde manuell definiert.

MQQMF_CLUSSDR_AUTO_DEFINED

Der Clustersenderkanal wurde automatisch definiert.

MQQMF_AVAILABLE

Der Zielwarteschlangenmanager ist für das Empfangen von Nachrichten verfügbar.

Other values

Andere Flags in diesem Feld können vom Warteschlangenmanager für interne Zwecke gesetzt werden.

Zugehörige Verweise

[Anfangswerte und Sprachendeklarationen für MQWCR](#)

Anfangswerte und Deklarationen in den Programmiersprachen C und High Level Assembler für die Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Anfangswerte und Sprachendeklarationen für MQWCR

Anfangswerte und Deklarationen in den Programmiersprachen C und High Level Assembler für die Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.

Tabelle 31. Anfangswerte der Felder in MQWCR

Name des Felds	Name der Konstante	Wert der Konstanten
<i>ClusterName</i>	--	" "
<i>ClusterRecOffset</i>	--	0
<i>ClusterFlags</i>	--	0

Deklaration in Programmiersprache C

```
typedef struct tagMQWCR {
    MQCHAR48 ClusterName; /* Cluster name */
    MQLONG ClusterRecOffset; /* Offset of next cluster record */
    MQLONG ClusterFlags; /* Cluster flags */
};
```

High Level Assembler

```
MQWCR          DSECT
MQWCR_CLUSTERNAME DS CL48 Cluster name
MQWCR_CLUSTERRECOFFSET DS F Offset of next cluster
* record
MQWCR_CLUSTERFLAGS DS F Cluster flags
MQWCR_LENGTH EQU *-MQWCR Length of structure
MQWCR_AREA DS CL(MQWCR_LENGTH)
```

Zugehörige Verweise

[Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.](#)

[Beschreibung der Felder in der Struktur des Clusterdatensatzes für Clusterauslastung MQWCR.](#)

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.

Tabelle 32. Kanalprogramme für Windows- und UNIX and Linux-Systeme

Programmname	Verbindungsrichtung	Kommunikation
amqrmppa		Alle
runmqlsr	Eingehend	Alle
amqcrs6a	Eingehend	LU 6.2
amqcrsta	Eingehend	TCP
runmqchl	Ausgehend	Alle
runmqchi	Ausgehend	Alle

Die Steuerbefehle 'runmqlsr' (Ausführung des WebSphere MQ-Empfangsprogramms), 'runmqchl' (Ausführung des WebSphere MQ-Kanals) und 'runmqchi' (Ausführung des WebSphere MQ-Kanalinitiators) 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

- Auf UNIX and Linux -Systemen verwenden Sie `export [environment variable]=filename`.
- Verwenden Sie auf Windows -Systemen `Set [environment variable]=filename`.
-

AMQ_MQS_INI_LOCATION

Auf UNIX and Linux-Systemen können Sie die Speicherposition, die für die Datei `mqc.ini` verwendet wird, ändern, indem Sie die Speicherposition der Datei `mqc.ini` in dieser Variablen 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_SSL_ALLOW_DEFAULT_CERT

Wenn die Umgebungsvariable `AMQ_SSL_ALLOW_DEFAULT_CERT` nicht festgelegt ist, kann eine Anwendung eine Verbindung zu einem WS-Manager mit einem persönlichen Zertifikat im Client-Keystore nur herstellen, wenn das Zertifikat den Kennsatznamen `ibmwebsphermq<userid>` enthält. Wenn die Umgebungsvariable `AMQ_SSL_ALLOW_DEFAULT_CERT` festgelegt ist, erfordert das Zertifikat nicht den Kennsatznamen `ibmwebsphermq<userid>`. 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 `ibmwebsphermq<userid>` 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.

V 7.5.0.9

AMQ_SSL_LDAP_SERVER_VERSION

Diese Variable kann verwendet werden, um sicherzustellen, dass LDAP v2 oder LDAP v3 von IBM WebSphere 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. Legen Sie `AMQ_SSL_LDAP_SERVER_VERSION=2` fest, 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 WebSphere MQ-Warteschlangenmanager für die Benutzerauthentifizierung oder die Benutzerberechtigung erstellt werden.

GMQ_MQ_LIB

Wenn sowohl der IBM WebSphere MQ MQI-Client als auch der IBM WebSphere 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 WebSphere MQ `amqzst.dll` zu laden. Wenn diese DLL nicht vorhanden ist, wie bei reinen Client-Installationen, versucht WebSphere MQ `mqic.dll` zu laden.

HOME

Diese Variable enthält den Namen des Verzeichnisses, in dem nach der Datei `mqclient.ini` gesucht wird. Diese Datei enthält Konfigurationsdaten, die von IBM WebSphere MQ MQI-Clients auf Systemen mit UNIX and Linux verwendet werden.

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. Diese Datei enthält Konfigurationsdaten, die von IBM WebSphere 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 Verzeichnissuche 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. Der Wert entspricht standardmäßig dem des lokalen Hosts, falls nicht anders angegeben.

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.

MQCERTVPOL

Bestimmt den verwendeten Typ der Prüfung der Gültigkeit des Zertifikats:

ANY

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

RFC5280

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

MQCHLLIB

Gibt den Verzeichnispfad zu der Datei an, die die Definitionstabelle für Clientkanäle (Client channel definition table, CCDT) enthält. Diese Datei wird auf dem Server erstellt, kann aber auch zur einer WebSphere MQ-MQI-Client-Workstation kopiert werden.

MQCHLTAB

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

MQC_IPC_HOST

Wenn IBM WebSphere MQ-Dateien gemeinsam genutzt werden und der für myHostName erstellte Wert Probleme verursacht, setzen Sie myHostName mithilfe der Umgebungsvariablen MQC_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 weitere 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 unterdrücken](#).

Die Umgebungsvariable MQ_CHANNEL_SUPPRESS_INTERVAL ist vergleichbar mit [SuppressInterval](#) in der Datei `qm.ini`.

MQ_CHANNEL_SUPPRESS_MSGS

Gibt IBM WebSphere 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 unterdrücken](#).

Die Umgebungsvariable MQ_CHANNEL_SUPPRESS_MSGS ist mit [SuppressMessage](#) in der Datei `qm.ini` vergleichbar, obwohl sie anders angegeben ist.

MQ_CONNECT_TYPE

Auf IBM WebSphere MQ für Windows- und UNIX and Linux-Systemen verwenden Sie diese Umgebungsvariable zusammen mit dem Binding-Typ, der im Feld 'Optionen' für die MQNO-Struktur angegeben ist, die bei MQCONNX-Aufrufen verwendet wird. Weitere Informationen finden Sie im Abschnitt [Umgebungsvariable 'MQCONNX'](#).

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\\FilePath
```

MQIPADDRV

MQIPADDRV gibt an, welches IP-Protokoll für eine Kanalverbindung verwendet werden soll. Die möglichen Zeichenfolgenwerte sind "MQIPADDR_IPV4" oder "MQIPADDR_IPV6". Diese Werte haben dieselbe Bedeutung wie "IPV4" und "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 WebSphere MQ classes for Java installiert sind, wie in den IBM WebSphere MQ classes for Java-Installationsverzeichnissen angezeigt.

MQ_JAVA_LIB_PATH

Gibt das Verzeichnis an, in dem die IBM WebSphere MQ classes for Java-Bibliotheken gespeichert werden. Einige Scripts, die mit IBM WebSphere MQ classes for Java bereitgestellt werden, wie z. B. IVTRun, verwenden diese Umgebungsvariable.

MQNAME

MQNAME gibt den lokalen NetBIOS-Namen an, den der IBM WebSphere 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 Trace-Dateien geschrieben werden sollen. Weitere Informationen finden Sie unter [Publish-Exit-Beispielprogramm](#)

MQSERVER

Mit der Umgebungsvariablen "MQSERVER" wird der Minimalkanal definiert. Sie können MQSERVER nicht verwenden, um einen SSL-Kanal oder einen Kanal mit Kanalexits zu definieren. MQSERVER gibt die Position des WebSphere MQ-Servers sowie die zu verwendende Kommunikationsmethode an.

MQ_SET_NODELAYACK

Wenn Sie diese Variable setzen, wird die verzögerte TCP-Bestätigung inaktiviert.

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

MQSNOAUT inaktiviert den Objektberechtigungsmanager (Object authority manager, OAM) und verhindert Sicherheitsprüfungen. Die Variable MQSNOAUT wird nur bei der Erstellung eines Warteschlangenmanagers aktiv.



Warnung: Um den OAM zu aktivieren, müssen Sie den Warteschlangenmanager löschen, die Umgebungsvariable löschen und anschließend 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 WebSphere 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 Schlüsselrepositors 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 stellt die Nummer der unverschlüsselten Bytes dar, die auf einem SSL-Kanal vor der Neuvereinbarung des geheimen SSL-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 WebSphere 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 amqsdq.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 Fehlermeldungsfunktionen. Setzen Sie die Umgebungsvariable ODQ_TRACE auf "YES" oder "yes", um die Traceerstellung zu aktivieren.

OMQ_PATH

Diese Umgebungsvariable gibt an, wo Sie den Symptombereicht zum ersten Auftreten eines Fehlers finden, wenn Ihre IBM WebSphere 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. Weitere Informationen finden Sie unter [Trace verwenden](#)

OMQ_TRACE_PATH

Eine der drei Umgebungsvariablen, mit denen die Traceerstellung gesteuert wird. Weitere Informationen finden Sie unter [Trace verwenden](#)

OMQ_TRACE_LEVEL

Eine der drei Umgebungsvariablen, mit denen die Traceerstellung gesteuert wird. Weitere Informationen finden Sie unter [Trace verwenden](#)

ONCONFIG

Der Name der Informix-Serverkonfigurationsdatei. Verwenden Sie zum Beispiel auf Systemen mit UNIX and Linux:

```
export ONCONFIG=onconfig.hostname_1
```

Verwenden Sie auf Windows-Systemen 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 WebSphere MQ installiert und konfiguriert wurde. Sie ist für lokale Warteschlangenmanager zugänglich. Weitere Informationen finden Sie unter [WCF client to a .NET service hosted by WebSphere MQ sample](#) und [WCF client to an Axis Java service hosted by WebSphere MQ sample](#)

Die Umgebungsvariable kann auch verwendet werden, wenn Sie WebSphere MQ-Webtransport für SOAP installieren. Weitere Informationen hierzu finden Sie unter [WebSphere MQ-Webtransport für SOAP installieren](#)

Beispiel einer Nachrichtenkanalplanung für verteilte Plattformen

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 wurde bereits von WebSphere MQ definiert. Sie können zwar eine andere Initialisierungswarteschlange verwenden, müssen diese jedoch selbst definieren und beim Start des Kanalinitiators den Namen der Warteschlange angeben.

Inhalt dieses Beispiels

Das Beispiel zeigt die WebSphere 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 beiden Methoden sehen identisch aus, aber um einen Befehl in der Befehlszeile auszugeben, müssen Sie zuerst `runmqsc` für den Standardwarteschlangenmanager oder `runmqsc qmname` eingeben, wobei `qmname` der Name des erforderlichen 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 in Windows-Systemen die Tastenkombination Strg+z, um die Eingabe in die Befehlszeile zu beenden. Verwenden Sie auf UNIX and Linux-Systemen den Befehl Strg+d. Alternativ können Sie den Befehl **end** eingeben.

Abbildung 4 auf Seite 148 zeigt das Beispielszenario.

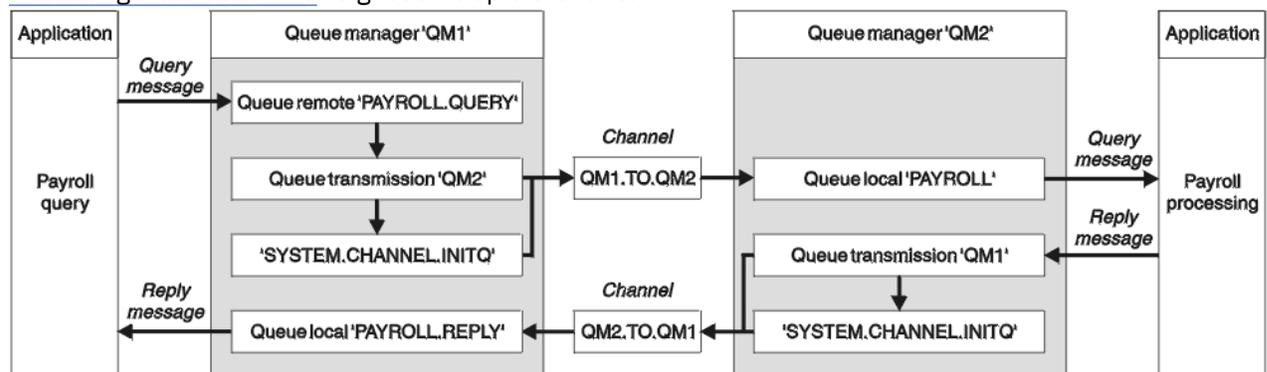


Abbildung 4. Beispiel für den Nachrichtenkanal für Windows-, UNIX and Linux-Systeme

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 Beispielfinitionen 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 4 auf Seite 148 zeigt ein entsprechendes Anordnungsdiagramm.

Beispiel Warteschlangenmanager QM1

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.

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) +
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) +
CONNAME('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 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.

Beispiel für Warteschlangenmanager QM2

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) +  
CONNNAME('192.0.2.0(1411)')
```

Empfängerkanaldefinition

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM1')
```

Beispiel 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 Listeners finden Sie unter [Kommunikation für Windows einrichten](#) und [Kommunikation auf UNIX and Linux -Systemen einrichten](#).

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 von Aliasnamen für Warteschlangenmanager und Aliasnamen für Empfangswarteschlangen für Antworten, um mehr darüber zu erfahren, wie diese in die Organisation Ihres Netzwerks aus Warteschlangenmanagern eingebunden werden können.

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 *  
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).

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 Farabbildungen.

Informationen zu Programmierschnittstellen

Die bereitgestellten Informationen zur Programmierschnittstelle sollen Sie bei der Erstellung von Anwendungssoftware für dieses Programm unterstützen.

Dieses Handbuch enthält Informationen zu geplanten Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zum Abrufen der Services von IBM WebSphere MQ zu schreiben.

Diese Informationen können jedoch auch Angaben über Diagnose, Bearbeitung und Optimierung enthalten. Die Informationen zu Diagnose, Bearbeitung und Optimierung sollten Ihnen bei der Fehlerbehebung für die Anwendungssoftware helfen.

Wichtig: Verwenden Sie diese Diagnose-, Änderungs- und Optimierungsinformationen nicht als Programmierschnittstelle, da sie Änderungen unterliegen.

Marken

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml. Weitere Produkt- und 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: