

9.4

IBM MQ Konfigurationsreferenz

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 279 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 4 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

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

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

Inhaltsverzeichnis

Konfigurationsreferenz.....	5
Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten.....	5
Verwendung der plattformübergreifenden Kommunikationsbeispiele.....	7
Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter AIX einrichten.....	9
Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter IBM i einrichten.....	16
Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Linux einrichten.....	32
Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Windows einrichten.....	39
Example: setting up IBM MQ cross-platform communication on z/OS.....	46
Example: setting up IBM MQ cross-platform communication on z/OS using QSGs.....	50
Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS.....	58
Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen.....	65
IBM MQ-Dateiberechtigungen in /opt/mqm mit setuid für mqm.....	70
IBM MQ-Dateisystemberechtigungen unter Windows.....	72
Benennungseinschränkungen für Warteschlangen.....	73
Benennungseinschränkungen für andere Objekte.....	75
Auflösung des Warteschlangennamens.....	76
Was ist die Auflösung von Warteschlangennamen?.....	78
Wie Zielobjektattribute für Aliasnamen, ferne Warteschlangen und Clusterwarteschlangen aufgelöst werden.....	79
System-und Standardobjekte.....	79
SYSTEM.BASE.TOPIC.....	85
Zeilengruppeninformationen für Konfigurationsdateien.....	86
Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen.....	88
Kanalattribute.....	90
Kanalattribute für MQSC-Schlüsselwörter (A-B).....	94
Kanalattribute für MQSC-Schlüsselwörter (C).....	98
Kanalattribute für MQSC-Schlüsselwörter (D-L).....	106
Kanalattribute für MQSC-Schlüsselwörter (M).....	113
Kanalattribute für MQSC-Schlüsselwörter (N-R).....	119
Kanalattribute für MQSC-Schlüsselwörter (S).....	123
Kanalattribute für MQSC-Schlüsselwörter (T-Z).....	128
IBM MQ -Clusterbefehle und -Attribute.....	131
In Kanaldefinitionsbefehlen verfügbare Clusterattribute.....	132
In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute.....	135
In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute.....	137
DISPLAY CLUSQMgr.....	139
REFRESH CLUSTER.....	141
RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen.....	142
Befehle SUSPEND QMgr und RESUME QMgr im Cluster.....	144
Lastausgleich in Clustern.....	145
Asynchronous behavior of CLUSTER commands on z/OS.....	155
Kanalprogramme.....	156
Jobs für übergreifende Kommunikation unter IBM i.....	156
Kanalzustände unter IBM i.....	156
Beispiel: Nachrichtenkanal unter AIX, Linux, and Windows planen.....	157
Nachrichtenkanalbeispiel für AIX, Linux, and Windows einrichten.....	159
Beispiel für AIX, Linux, and Windows ausführen und erweitern.....	160
Beispiel: Nachrichtenkanal unter IBM i planen.....	161
Nachrichtenkanalagenten unter IBM i einrichten.....	163
Beispiel für IBM i ausführen und erweitern.....	165
Example: planning a message channel on z/OS.....	166
Setting up the message channel agent on z/OS.....	168

Running and expanding the example for z/OS.....	170
Example: planning a message channel for z/OS using queue sharing groups.....	170
Setting up the queue sharing group definitions and a queue manager QM3 not in the queue sharing group.....	172
Running the queue sharing group example for z/OS.....	173
Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek.....	174
Managed File Transfer -Konfigurationsreferenz.....	174
Verwendung von Umgebungsvariablen in MFT-Eigenschaften.....	174
Die MFT-Datei 'installation.properties'.....	176
Die MFT agent.properties-Datei.....	180
Die MFT-Datei 'coordination.properties'.....	204
Die MFT-Datei 'command.properties'.....	209
Die MFT-Datei 'logger.properties'.....	213
Von der Funktion 'LogTransfer' erzeugte Ausgabe.....	223
Java-Systemeigenschaften für MFT.....	226
SHA-2-CipherSpecs und -CipherSuites für MFT.....	227
Konfigurationsdateien der MFT-Dateiprotokollfunktion.....	227
Die Bibliothek SCSQFCMD.....	235
Thema 'SYSTEM.FTE'.....	236
Einstellungen von MFT-Agentenwarteschlangen.....	238
MFT-Systemwarteschlangen und der Systemabschnitt.....	239
Konventionen zum Benennen von MFT-Objekten.....	241
Statusnachrichten von MFT-Agenten.....	242
IBM MQ Internet Pass-Thru -Konfigurationsreferenz.....	243
Zusammenfassung der MQIPT-Eigenschaften.....	244
Globale MQIPT-Eigenschaften.....	251
MQIPT-Routeneigenschaften.....	254
mqiptAdmin Eigenschaften.....	276
Bemerkungen.....	279
Informationen zu Programmierschnittstellen.....	280
Marken.....	281


Konfigurationsreferenz

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

Die Konfigurationsreferenzinformationen werden in folgenden Unterabschnitten bereitgestellt:

Zugehörige Tasks

konfigurieren


 z/OS konfigurieren

Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten

Dieses Beispiel zeigt, wie Sie ein funktionsfähiges IBM MQ -Netz einrichten, indem Sie IBM MQ -Sender- und -Empfängerkanäle konfigurieren, um den bidirektionalen Nachrichtenfluss zwischen den Plattformen über alle unterstützten Protokolle zu aktivieren.








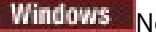

Vorbereitende Schritte

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

-  z/OS kommuniziert über einen 3745 -Netzcontroller (oder einen entsprechenden Controller)

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. In ähnlicher Weise wird bei TCP davon ausgegangen, dass die Namensserverfunktion verfügbar ist, indem entweder ein Domänennamensserver oder lokale Tabellen (z. B. eine Hostdatei) verwendet werden.

Die Beispielkonfigurationen decken die folgenden Netzsoftwareprodukte ab:

- SNA (Systems Network Architecture)
 -  IBM Personal Communications für Windows
 -  IBM Communications Server für AIX
 -  IBM i
 - OS/390
- TCP
 -  Microsoft Windows
 -  AIX
 -  IBM i
 -  TCP für z/OS
-  NetBIOS
-  SPX

Weitere Informationen zu unterstützten Kommunikationsprotokollen und Software finden Sie unter [Systemvoraussetzungen für IBM MQ](#).

Informationen zu diesem Vorgang

In diesem Beispiel werden Sender- und Empfängerkanäle verwendet. Um andere Kanaltypen als Sender- und Empfänger zu verwenden, [DEFINE CHANNEL](#) (neuen Kanal definieren).

Abbildung 1 auf Seite 6 ist eine konzeptionelle Darstellung eines einzelnen Kanals und der ihm zugeordneten IBM MQ-Objekte.

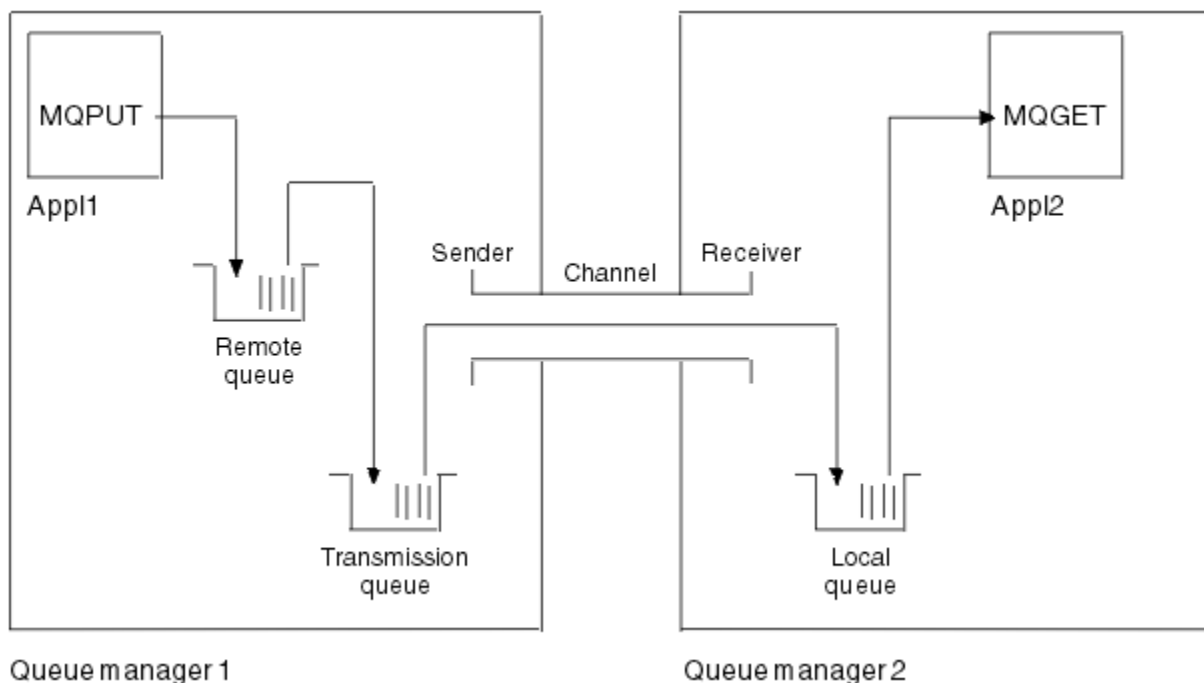


Abbildung 1. IBM MQ-Kanal, der in der Beispielkonfiguration einzurichten ist

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

Objekte in diesem Netz:

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

Vorgehensweise

1. Lesen Sie die Informationen unter [„Verwendung der plattformübergreifenden Kommunikationsbeispiele“](#) auf Seite 7.
2. Befolgen Sie die Anweisungen für die entsprechende (n) Plattform (en), um eine Netzverbindung herzustellen und die Kanäle zu definieren.
 - a)  Siehe [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter AIX einrichten“](#) auf Seite 9
 - b)  Siehe [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter IBM i einrichten“](#) auf Seite 16
 - c)  Siehe [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Linux einrichten“](#) auf Seite 32
 - d)  Siehe [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Windows einrichten“](#) auf Seite 39
 - e)  Siehe [„Example: setting up IBM MQ cross-platform communication on z/OS“](#) auf Seite 46
 - f)  Siehe [„Example: setting up IBM MQ cross-platform communication on z/OS using QSGs“](#) auf Seite 50
 - g)  Siehe [„Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS“](#) auf Seite 58

Zugehörige Tasks

[Verteilte Warteschlangensteuerung konfigurieren](#)

[Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten](#)

Verwendung der plattformübergreifenden Kommunikationsbeispiele

Die Beispielkonfigurationen zum Einrichten der plattformübergreifenden Kommunikation für IBM MQ beschreiben die Tasks, die auf einer einzelnen Plattform ausgeführt werden, um die Kommunikation mit einer anderen Plattform einzurichten. Die Beispiele beschreiben dann die Tasks zum Einrichten eines Arbeitskanals zu dieser Plattform.

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.

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

- **Linux** **AIX** Unter AIX and Linux:

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

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

- **Windows** Unter Windows:

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

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

Es gibt Beispiele, in denen Sie die in den Beispielkonfigurationen verwendeten Parameter finden. Zu jedem Parameter gibt es eine Kurzbeschreibung und Angaben dazu, wo die entsprechenden Werte in Ihrem System zu finden sind. Wenn Sie über eine eigene Gruppe von Werten verfügen, stellen Sie sicher, dass Sie diese Werte beim Durcharbeiten der Beispiele in diesem Abschnitt verwenden.

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:

- **AIX** [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter AIX einrichten“ auf Seite 9](#)
- **IBM i** [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter IBM i einrichten“ auf Seite 16](#)
- **Linux** [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Linux einrichten“ auf Seite 32](#)
- **Windows** [„Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Windows einrichten“ auf Seite 39](#)
- **z/OS** [„Example: setting up IBM MQ cross-platform communication on z/OS“ auf Seite 46](#)
- **z/OS** [„Example: setting up IBM MQ cross-platform communication on z/OS using QSGs“ auf Seite 50](#)
- **z/OS** [„Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS“ auf Seite 58](#)

IT-Zuständigkeiten

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

- Systemadministrator: Die Person (oder Gruppe von Personen), die die Software für eine bestimmte Plattform installiert und konfiguriert.

- Netzadministrator: Die Person, die die LAN-Konnektivität, die LAN-Adresszuordnungen, die Namenskonventionen im Netz und andere mit dem Netz verbundenen Tasks steuert. Diese Person kann Mitglied der Systemverwaltungsgruppe oder einer separaten Gruppe sein.

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

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

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

Zugehörige Tasks

„[Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten](#)“ auf Seite 5

Dieses Beispiel zeigt, wie Sie ein funktionsfähiges IBM MQ -Netz einrichten, indem Sie IBM MQ -Sender- und -Empfängerkanäle konfigurieren, um den bidirektionalen Nachrichtenfluss zwischen den Plattformen über alle unterstützten Protokolle zu aktivieren.

Zugehörige Verweise

[setmqenv](#)

Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter AIX einrichten

Dieses Beispiel zeigt, wie Sie Kommunikationsverbindungen von IBM MQ unter AIX zu IBM MQ auf einer anderen Plattform einrichten und einen funktionierenden Kanal zu dieser Plattform einrichten.

Vorbereitende Schritte

Hintergrundinformationen zu diesem Beispiel und seiner Verwendung finden Sie unter [„Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten“](#) auf Seite 5 und [„Verwendung der plattformübergreifenden Kommunikationsbeispiele“](#) auf Seite 7.

Informationen zu diesem Vorgang

In diesem Beispiel wird die plattformübergreifende Kommunikation zwischen IBM MQ unter AIX und den folgenden Plattformen beschrieben:

-  Windows
-  Linux
-  IBM i
-  z/OS
- VSE/ESA

Vorgehensweise

1. Richten Sie eine Netzverbindung mit einer der folgenden Optionen ein.
 - Stellen Sie eine LU 6.2 -Verbindung her. Weitere Informationen zur Konfiguration von SNA über TCP/IP finden Sie unter [Communications Server für AIX Library](#).
 - Stellen Sie eine TCP-Verbindung her.

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. Verwenden Sie den folgenden Befehl, um den IBM MQ for TCP-Listener zu starten:

```
runmqclsr -t tcp
```

a. Bearbeiten Sie die Datei `/etc/services`.

Anmerkung: Um die `/etc/services`-Datei zu bearbeiten, müssen Sie als Superuser oder Root angemeldet sein. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

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

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

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

c. 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".

2. Nachdem die Verbindung hergestellt wurde, definieren Sie einige Kanäle wie in „[Kanäle unter AIX konfigurieren](#)“ auf Seite 10 beschrieben.

Kanäle unter AIX konfigurieren

Um IBM MQ für die Beispielkonfiguration unter AIX zu konfigurieren, müssen Sie die grundlegenden Konfigurationsschritte für den Warteschlangenmanager ausführen und anschließend die Sender- und Empfängerkanäle konfigurieren.

Vorbereitende Schritte

Anmerkung:

1. Stellen Sie vor Beginn des Installationsprozesses sicher, dass Sie zuerst den Benutzer und die Gruppe `mqm` erstellt und das Kennwort festgelegt haben.
2. Wenn die Installation aufgrund von nicht ausreichendem Speicherplatz im Dateisystem fehlschlägt, können Sie die Größe wie folgt erhöhen, indem Sie den Befehl `smitty csn` verwenden. (Geben Sie `df` ein, um den Status des Dateisystems anzuzeigen. Daraufhin wird der volle logische Datenträger angezeigt).

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

3. Starten Sie einen beliebigen Kanal über den Befehl:

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

4. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert, wobei `MQ_INSTALLATION_PATH` das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
5. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
6. Unter AIX können Sie einen Trace der IBM MQ-Komponenten durch Eingabe der Standardtracebefehle für IBM MQ oder mithilfe des AIX-Systemtrace starten. Weitere Informationen zu IBM MQ Trace und AIX -Systemtrace finden Sie unter [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.

Vorgehensweise

1. Führen Sie die folgenden Schritte aus, um die Basiskonfiguration für den Warteschlangenmanager auszuführen:
 - a) Erstellen Sie den Warteschlangenmanager aus der AIX-Befehlszeile mithilfe des folgenden Befehls:

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

Dabei gilt:

aix

ist der Name des Warteschlangenmanagers

-q

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

-u *dlqname*

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

- b) Starten Sie den Warteschlangenmanager aus der AIX-Befehlszeile mithilfe des folgenden Befehls:

```
strmqm aix
```

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

- c) 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.

2. Konfigurieren Sie die Kanäle für die Beispielkonfiguration.

Weitere Informationen zu den in den folgenden Beispielen verwendeten Parametern finden Sie unter „Kanalkonfigurationsparameter für AIX“ auf Seite 13. In jedem Fall zeigt das Beispiel den MQSC-Befehl. Entweder starten Sie **runmqsc** in einer AIX-Befehlszeile und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Windows Diese Beispiele gelten für die Verbindung von IBM MQ unter AIX mit IBM MQ unter Windows. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie die entsprechenden Werte aus den Tabellen in „[Kanalkonfigurationsparameter für AIX](#)“ auf Seite 13 anstelle der Werte für Windows.

- a) Definieren Sie den Senderkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden:

```
def ql (WINNT) +
```

```
F
```

```

usage(xmitq) +
replace

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

def chl (AIX.WINNT.SNA) chltype(sdr) +
trptype(lu62) +
conname('WINNTCPIC') +
xmitq(WINNT) +
replace
G
17
F

```

- TCP verwenden:

```

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

```

b) Definieren Sie den Empfängerkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden:

```

def ql (AIX.LOCALQ) replace
B

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

```

- TCP verwenden:

```

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

```

Anmerkung: Es gibt verschiedene Möglichkeiten sicherzustellen, dass SNA-Empfängerkanäle ordnungsgemäß aktiviert werden, wenn ein Senderkanal einen Dialog 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* ist das übergeordnete Verzeichnis, in dem IBM MQ installiert ist. Nachdem diese Datei erstellt wurde, aktivieren Sie deren Ausführung durch den Befehl:

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

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

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

AIX Kanalkonfigurationsparameter für AIX

Die Parameter, die zum Konfigurieren der Kanäle für die Beispielkonfiguration unter AIX erforderlich sind.

Schritt „2“ auf Seite 11 von „Kanäle unter AIX konfigurieren“ auf Seite 10 beschreibt die Konfiguration, die auf dem AIX -Warteschlangenmanager ausgeführt werden muss, um den in „Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten“ auf Seite 5 beschriebenen Kanal zu implementieren. Die Beispiele in „Kanäle unter AIX konfigurieren“ auf Seite 10 beziehen sich auf die Verbindung von IBM MQ for IBM i und IBM MQ for Windows. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie die Werte aus der entsprechenden Tabelle anstelle der Werte für Windows.

Anmerkung: Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.

Definition für lokalen Knoten

Tabelle 1. Konfigurationsbeispiele für die Definition des lokalen Knotens			
ID	Parametername	Referenz	Verwendetes Beispiel
A	Name des Warteschlangenmanagers		AIX
B	Name der lokalen Warteschlange		AIX.LOCALQ

Verbindung zu IBM MQ unter Windows

Windows

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für Windows“ auf Seite 43 angegebenen entsprechen.

Tabelle 2. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Windows			
ID	Parametername	Referenz	Verwendetes Beispiel
C	Name des fernen Warteschlangenmanagers	A	WINNT
D	Name der fernen Warteschlange		WINNT.REMOTEQ

Tabelle 2. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Windows (Forts.)

ID	Parametername	Referenz	Verwendetes Beispiel
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 zu IBM MQ unter Linux

Linux

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für Linux“ auf Seite 36 angegebenen entsprechen.

Tabelle 3. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Linux

ID	Parametername	Referenz	Verwendetes Beispiel
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

Verbindung zu IBM MQ unter IBM i

IBM i

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für IBM i“ auf Seite 30 angegebenen entsprechen.

Tabelle 4. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter IBM i

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
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)		AIX.AS400.SNA
H	Name des Senderkanals (TCP)		AIX.AS400.TCP

Tabelle 4. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
I	Name des Empfängerkanals (SNA)	G	AS400.AIX.SNA
J	Name des Empfängerkanals (TCP)	H	AS400.AIXTCP-

Verbindung zu IBM MQ for z/OS



Die Werte in diesem Abschnitt der Tabelle müssen den in „Channel configuration parameters for z/OS“ auf Seite 48 angegebenen entsprechen.

Tabelle 5. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS

ID	Parametername	Referenz	Verwendetes Beispiel
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)		AIX.MVS.SNA
H	Name des Senderkanals (TCP)		AIX.MVS.TCP
I	Name des Empfängerkanals (SNA)	G	MVS.AIX.SNA
J	Name des Empfängerkanals (TCP)	H	MVS.AIX.TCP

Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange



Die Werte in diesem Abschnitt der Tabelle müssen den in „Shared channel configuration parameters“ auf Seite 57 angegebenen entsprechen.

Tabelle 6. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange

ID	Parametername	Referenz	Verwendetes Beispiel
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
J	Name des Empfängerkanals (TCP)	H	QSG.AIX.TCP

IBM i **Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter IBM i einrichten**

Dieses Beispiel zeigt, wie Sie Kommunikationsverbindungen von IBM MQ unter IBM i zu IBM MQ auf einer anderen Plattform einrichten und einen funktionierenden Kanal zu dieser Plattform einrichten.

Vorbereitende Schritte

Hintergrundinformationen zu diesem Beispiel und seiner Verwendung finden Sie unter [„Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten“](#) auf Seite 5 und [„Verwendung der plattformübergreifenden Kommunikationsbeispiele“](#) auf Seite 7.

Informationen zu diesem Vorgang

In diesem Beispiel wird die plattformübergreifende Kommunikation zwischen IBM MQ unter IBM i und den folgenden Plattformen beschrieben:

-  Windows
-  AIX
-  Linux
-  z/OS oder MVS
- VSE/ESA

Vorgehensweise

1. Richten Sie eine Netzverbindung mit einer der folgenden Optionen ein.
 - Stellen Sie eine LU 6.2 -Verbindung her, wie in [„LU 6.2 -Verbindung unter IBM i herstellen“](#) auf Seite 16 beschrieben.
 - Richten Sie eine TCP-Verbindung wie in [„TCP-Verbindung unter IBM i aufbauen“](#) auf Seite 25 beschrieben ein.
2. Nachdem die Verbindung hergestellt wurde, definieren Sie einige Kanäle wie in [„Kanäle unter IBM i konfigurieren“](#) auf Seite 26 beschrieben.

IBM i **LU 6.2 -Verbindung unter IBM i herstellen**

Zum Herstellen einer LU 6.2 -Verbindung unter IBM i müssen Sie den lokalen Knoten konfigurieren und ihn mit einem Partnerknoten verbinden.

Informationen zu diesem Vorgang

Weitere Informationen zu den Parametern, die zum Einrichten der Kommunikation zwischen dem IBM i -System und einer der anderen IBM MQ Plattformen erforderlich sind, finden Sie in den Tabellen unter [„Konfigurationsparameter für eine LU 6.2 -Verbindung unter IBM i“](#) auf Seite 20. Die Zahlen in Klammern () in den Taskschritten entsprechen den Werten in der Spalte *ID* dieser Tabellen.

Gehen Sie wie folgt vor, um den lokalen Knoten zu konfigurieren:

- Leitungsbeschreibung erstellen
- Einen Leitwegeintrag hinzufügen und anschließend das Subsystem starten

Gehen Sie wie folgt vor, um eine Verbindung zu einem Partnerknoten herzustellen:

- Steuereinheitenbeschreibung erstellen
- Einheitenbeschreibung erstellen
- CPI-C-Nebeninformationen erstellen

- DFV-Eintrag für APPC hinzufügen
- Konfigurationslisteneintrag hinzufügen

Vorgehensweise

1. Konfigurieren Sie den lokalen Knoten, indem Sie eine Leitungsbeschreibung erstellen und einen Leitwegeintrag hinzufügen.

a) Leitungsbeschreibung erstellen.

Wenn die Leitungsbeschreibung noch nicht erstellt wurde, können Sie mit dem Befehl **CRTLINTRN** Werte für **Leitungsbeschreibung** (6) und **Ressourcenname** (7) angeben, wie im folgenden Beispiel gezeigt:

```

Create Line Desc (token-ring) (CRTLINTRN)
Type choices, press Enter.

Line description . . . . . TOKENRINGL Name
Resource name . . . . . LIN041 Name, *NWID
NWI type . . . . . *FR *FR, *ATM
Online at IPL . . . . . *YES *YES, *NO
Vary on wait . . . . . *NOWAIT *NOWAIT, 15-180 (1 second)
Maximum controllers . . . . . 40 1-256
Attached NWI . . . . . *NONE Name, *NONE

Bottom
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
Parameter LIND required. +

```

b) Einen Leitwegeintrag hinzufügen.

Geben Sie den Befehl **ADDRTGE** ein und drücken Sie die Eingabetaste. Geben Sie dann Ihren eigenen Wert für **Subsystembeschreibung** (5) und die Werte an, die im folgenden Beispiel für **Folgennummer des Routing-Eintrags, Vergleichswert** (8), **Anfangsposition, aufzurufendes Programm** und die **Bibliothek** angezeigt werden, die das aufzurufende Programm enthält.

```

Add Routing Entry (ADDRTGE)
Type choices, press Enter.

Subsystem description . . . . . QCMN Name
Library . . . . . *LIBL Name, *LIBL, *CURLIB
Routing entry sequence number . 1 1-9999
Comparison data:
Compare value . . . . . 'MQSERIES'

Starting position . . . . . 37 1-80
Program to call . . . . . AMQCRC6B Name, *RTGDTA
Library . . . . . QMAS400 Name, * LI BL, *CURLIB
Class . . . . . *SBSD Name, *SBSD
Library . . . . . *LIBL Name, *LIBL, *CURLIB
Maximum active routing steps . . *NOMAX 0-1000, *NOMAX
Storage pool identifier . . . . . 1 1-10

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

```

Starten Sie das Subsystem, indem Sie den Befehl **STRSBS** *subsystem description* (5) eingeben und die Eingabetaste drücken.

- Erstellen Sie die Verbindung zum Partnerknoten, indem Sie eine Steuereinheitenbeschreibung, eine Einheitenbeschreibung und die CPI-C-Nebeninformationen erstellen und einen Kommunikationseintrag für APPC sowie einen Konfigurationslisteneintrag hinzufügen.

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

a) Steuereinheitenbeschreibung erstellen

Geben Sie in einer Befehlszeile CRTCTLAPPC ein und drücken Sie die Eingabetaste. Geben Sie dann Werte für **Controller description** (12) an, setzen Sie **Link type** auf *LAN und setzen Sie **Online bei IPL** auf *NO.

```

Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

Controller description . . . . . WINNTCP      Name
Link type . . . . . *LAN      *FAX, *FR, *IDLC,
*LAN...
Online at IPL . . . . . *NO      *YES, *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
Parameter CTLD required.          +

```

Drücken Sie zweimal die Eingabetaste, gefolgt von F10, geben Sie dann Werte für **Wählleitungsliste** (6), **Ferne Netz-ID** (9), **Ferner Kontrollpunkt** (10) und **Adresse des fernen LAN-Adapters** (16) an und drücken Sie die Eingabetaste.

```

Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

Controller description . . . . . > WINNTCP      Name
Link type . . . . . > *LAN      *FAX, *FR, *IDLC, *LAN...
Online at IPL . . . . . > *NO      *YES, *NO
APPN-capable . . . . . *YES      *YES, *NO
Switched line list . . . . . TOKENRINGL Name
+ for more values
Maximum frame size . . . . . *LINKTYPE  265-16393, 256, 265, 512...
Remote network identifier . . . NETID      Name, *NETATR, *NONE, *ANY
Remote control point . . . . . WINNTCP      Name, *ANY
Exchange identifier . . . . . 00000000-FFFFFFFF
Initial connection . . . . . *DIAL      *DIAL, *ANS
Dial initiation . . . . . *LINKTYPE  *LINKTYPE, *IMMED, *DELAY
LAN remote adapter address . . . 10005AFC5D83 000000000001-FFFFFFFFFFFF
APPN CP session support . . . . *YES      *YES, *NO
APPN node type . . . . . *ENDNODE  *ENDNODE, *LENNODE...
APPN transmission group number  1      1-20, *CALC
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

b) Erstellen Sie eine Einheitenbeschreibung.

Geben Sie den Befehl CRTDEVAPPC ein und drücken Sie die Eingabetaste. Geben Sie dann Werte für **Device description** (13), **Remote location** (11), **Local location** (3), **Remote network identifier** (9) und **Attached controller** (12) an.

```

Create Device Desc (APPC) (CRTDEVAPPC)

Type choices, press Enter.

Device description . . . . . WINNTLU      Name
Remote location . . . . . WINNTLU      Name
Online at IPL . . . . . *YES          *YES, *NO
Local location . . . . . AS400LU      Name, *NETATR
Remote network identifier . . . . . NETID   Name, *NETATR, *NONE
Attached controller . . . . . WINNTCP   Name
Mode . . . . . *NETATR              Name, *NETATR
+ for more values
Message queue . . . . . QSYSOPR      Name, QSYSOPR
Library . . . . . *LIBL             Name, *LIBL, *CURLIB
APPN-capable . . . . . *YES          *YES, *NO
Single session:
Single session capable . . . . . *NO      *NO, *YES
Number of conversations . . . . . 1-512

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
Parameter DEVD required.

```

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

c) Erstellen Sie die CPI-C-Nebeninformationen.

Geben Sie CRTCSI ein und drücken Sie die Taste F10. Geben Sie anschließend Werte für **Nebeninformationen** (14), **Ferner Standort** (11), **Transaktionsprogramm** (15), **Lokaler Standort** (3), **Modus** und **Ferne Netz-ID** (9) an und drücken die Eingabetaste.

```

Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . NTCPIC      Name
Library . . . . . *CURLIB          Name, *CURLIB
Remote location . . . . . WINNTLU      Name
Transaction program . . . . . MQSERIES

Text 'description' . . . . . *BLANK

Additional Parameters

Device . . . . . *LOC              Name, *LOC
Local location . . . . . AS400LU      Name, *LOC, *NETATR
Mode . . . . . #INTER              Name, *NETATR
Remote network identifier . . . . . NETID   Name, *LOC, *NETATR, *NONE
Authority . . . . . *LIBCRTAUT      Name, *LIBCRTAUT, *CHANGE...

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

```

d) Einen DFV-Eintrag für APPC hinzufügen.

Geben Sie in einer Befehlszeile ADDCMNE ein und drücken Sie die Eingabetaste. Geben Sie dann Werte für **Subsystembeschreibung** (5) und **Einheit** (13) an und drücken Sie erneut die Eingabetaste.

```

Add Communications Entry (ADDCMNE)

Type choices, press Enter.

Subsystem description . . . . . QCMN      Name
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Device . . . . . WINNTLU     Name, generic*, *ALL...
Remote location . . . . . Name
Job description . . . . . *USRPRF  Name, *USRPRF, *SBSD
Library . . . . . Name, *LIBL, *CURLIB
Default user profile . . . . . *NONE   Name, *NONE, *SYS
Mode . . . . . *ANY      Name, *ANY
Maximum active jobs . . . . . *NOMAX  0-1000, *NOMAX

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

```

e) Fügen Sie einen Konfigurationslisteneintrag hinzu.

Geben Sie ADDCFGLE *APPNRMT ein und drücken Sie F4. Geben Sie anschließend Werte für **Name des fernen Standorts (11)**, **Ferne Netz-ID (9)**, **Name des lokalen Standorts (3)**, **Ferner Kontrollpunkt (10)** und **Netz-ID des Kontrollpunkts (9)** an und drücken Sie die Eingabetaste.

```

Add Configuration List Entries (ADDCFGLE)

Type choices, press Enter.

Configuration list type . . . . > *APPNRMT  *APPNLCL, *APPNRMT...
APPN remote location entry:
Remote location name . . . . . WINNTLU     Name, generic*, *ANY
Remote network identifier . . . NETID      Name, *NETATR, *NONE
Local location name . . . . . AS400LU    Name, *NETATR
Remote control point . . . . . WINNTCP    Name, *NONE
Control point net ID . . . . . NETID     Name, *NETATR, *NONE
Location password . . . . . *NONE
Secure location . . . . . *NO           *YES, *NO
Single session . . . . . *NO           *YES, *NO
Locally controlled session . . *NO           *YES, *NO
Pre-established session . . . *NO           *YES, *NO
Entry 'description' . . . . . *BLANK
Number of conversations . . . 10         1-512
+ for more values

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

```

Nächste Schritte

Die LU 6.2-Verbindung ist nun eingerichtet. Sie können die Konfiguration wie in „[Kanäle unter IBM i konfigurieren](#)“ auf Seite 26 beschrieben abschließen.

Zugehörige Tasks

„[TCP-Verbindung unter IBM i aufbauen](#)“ auf Seite 25

Wenn TCP bereits konfiguriert ist, gibt es keine zusätzlichen Konfigurationstasks. Ist TCP/IP nicht konfiguriert, müssen Sie eine TCP/IP-Schnittstelle hinzufügen, eine TCP/IP-Loopback-Schnittstelle hinzufügen und einen Standardleitweg hinzufügen.

Konfigurationsparameter für eine LU 6.2 -Verbindung unter IBM i

Die Parameter, die zum Einrichten der Kommunikation von IBM MQ auf einem IBM i -System mit einer der anderen IBM MQ -Plattformen über eine LU 6.2 -Verbindung erforderlich sind.

Verwenden Sie diese Tabellen mit den Tabellen für die Plattform, zu der Sie eine Verbindung herstellen.

Wenn Zahlen in der Spalte *Referenz* angezeigt werden, geben sie an, dass der Wert mit dem Wert in der entsprechenden Tabelle in diesem Abschnitt übereinstimmen muss. Die Taskschritte in „LU 6.2 -Verbindung unter IBM i herstellen“ auf Seite 16 beziehen sich auf die Werte in der Spalte *ID* dieser Tabelle.

Die Einträge in der Spalte *Parametername* werden im Abschnitt „Erläuterung der in den Tabellen verwendeten Begriffe“ auf Seite 23 erläutert.

Definition für den lokalen Knoten

Tabelle 7. Konfigurationsbeispiele für die Definition des lokalen Knotens

ID	Parametername	Referenz	Verwendetes Beispiel
1	ID des lokalen Netzes		NETID
2	Name des lokalen Steuerpunkts		AS400PU
3	LU-Name		AS400LU
4	LAN-Zieladresse		10005A5962EF
5	Subsystembeschreibung		QCMN
6	Leitungsbeschreibung		TOKENRINGL
7	Ressourcenname		LIN041
8	Name des lokalen Transaktionsprogramms		MQSERIES

Verbindung zu IBM MQ on Windows

Windows

Windows

Tabelle 8. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Windows

ID	Parametername	Referenz	Verwendetes Beispiel
9	Netz-ID	2	NETID
10	Steuerpunktname	3	WINNTCP
11	LU-Name	5	WINNTLU
12	Controllerbeschreibung		WINNTCP
13	Einheit		WINNTLU
14	Nebeninformationen		NTCPIC
15	Transaktionsprogramm	7	MQSERIES
16	LAN-Adapteradresse	9	08005AA5FAB9
17	Modus	17	#INTER

Verbindung zu IBM MQ unter AIX

AIX

Tabelle 9. Konfigurationsbeispiele für die Verbindung zu IBM MQ auf AIX -Systemen

ID	Parametername	Referenz	Verwendetes Beispiel
9	Netz-ID	1	NETID
10	Steuerpunktname	2	AIXPU

Tabelle 9. Konfigurationsbeispiele für die Verbindung zu IBM MQ auf AIX -Systemen (Forts.)

ID	Parametername	Referenz	Verwendetes Beispiel
11	LU-Name	4	AIX-LU
12	Controllerbeschreibung		AIXPU
13	Einheit		AIX-LU
14	Nebeninformationen		AIXCPIC
15	Transaktionsprogramm	6	MQSERIES
16	LAN-Adapteradresse	8	123456789012
17	Modus	14	#INTER

Verbindung zu IBM MQ auf Linux (Plattformx86)

Linux

Tabelle 10. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Linux (x86 -Plattform)

ID	Parametername	Referenz	Verwendetes Beispiel
9	Netz-ID	4	NETID
10	Steuerpunktname	2	LINUXPU
11	LU-Name	5	LINUXLU
12	Controllerbeschreibung		LINUXPU
13	Einheit		LINUXLU
14	Nebeninformationen		LXCPIC
15	Transaktionsprogramm	7	MQSERIES
16	LAN-Adapteradresse	8	08005AC6DF33
17	Modus	6	#INTER

Verbindung zu IBM MQ for z/OS

z/OS

Tabelle 11. Konfigurationsbeispiele für Verbindung zu IBM MQ for z/OS

ID	Parametername	Referenz	Verwendetes Beispiel
9	Netz-ID	2	NETID
10	Steuerpunktname	3	MVSPU
11	LU-Name	4	MVSLU
12	Controllerbeschreibung		MVSPU
13	Einheit		MVSLU
14	Nebeninformationen		MVSPIC
15	Transaktionsprogramm	7	MQSERIES
16	LAN-Adapteradresse	8	400074511092
17	Modus	6	#INTER

Verbindung zu einem VSE/ESA -System

Tabelle 12. Konfigurationsbeispiele für Verbindung zu einem VSE/ESA -System			
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
9	Netz-ID	1	NETID
10	Steuerpunktname	2	VSEPU
11	LU-Name	3	VSELU
12	Controllerbeschreibung		VSEPU
13	Einheit		VSELU
14	Nebeninformationen		VSECPIC
15	Transaktionsprogramm	4	MQ01
16	LAN-Adapteradresse	5	400074511092
17	Modus		#INTER

Erläuterung der in den Tabellen verwendeten Begriffe

1 2 3

Informationen zum Suchen der konfigurierten Werte finden Sie unter „[Netzattribute finden](#)“ auf Seite [24](#).

4 LAN-Zieladresse

Die Hardwareadresse des Token-Ring-Adapters des IBM i-Systems. Sie finden den Wert mit *DSPLIND Line description* (6).

5 Subsystembeschreibung

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

6 Leitungsbeschreibung

Wenn dieser Parameter angegeben wurde, wird er im Beschreibungsfeld des Ressourcennamens angezeigt. Weitere Informationen finden Sie unter „[Den Wert des Ressourcennamens ermitteln](#)“ auf Seite [24](#). Wird der Wert nicht angegeben, muss eine Leitungsbeschreibung erstellt werden.

7 Ressourcename

Informationen zum Suchen des konfigurierten Werts finden Sie unter „[Den Wert des Ressourcennamens ermitteln](#)“ auf Seite [24](#).

8 Name des lokalen Transaktionsprogramms

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

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

12 Controllerbeschreibung

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

13 Einheit

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

14 Nebeninformationen

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

Netzattribute finden

Der lokale Knoten wurde teilweise als Bestandteil der IBM i-Installation konfiguriert. Geben Sie den Befehl **DSPNETA** ein, um die aktuellen Netzattribute anzuzeigen.

Wenn Sie diese Werte ändern müssen, verwenden Sie den Befehl **CHGNETA**. Zum Anwenden der Änderungen ist unter Umständen ein IPL erforderlich.

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

More...
Press Enter to continue.

F3=Exit F12=Cancel
```

Überprüfen Sie, ob die Werte für **Local network ID** (1), **Local control point name** (2) und **Default local location** (3) den Werten in der Tabelle oder Ihren eigenen Werten entsprechen, wenn Sie sie geändert haben.

Den Wert des Ressourcennamens ermitteln

Um den Wert des Ressourcennamens zu suchen, geben Sie **WRKHDWRSC TYPE(*CMN)** ein und drücken Sie die Eingabetaste.

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


```

Work with Communication Resources
System: AS400PU
Type options, press Enter.
2=Edit 4=Remove 5=Work with configuration description
7=Add configuration description ...

```

```

Configuration
Opt Resource      Description Type Description
CC02              2636 Comm Processor
LIN04             2636 LAN Adapter
LIN041  TOKEN-RING 2636 Token-ring Port

```

```

Bottom
F3=Exit  F5=Refresh  F6=Print  F11=Display resource addresses/statuses
F12=Cancel  F23=More options

```

IBM i TCP-Verbindung unter IBM i aufbauen

Wenn TCP bereits konfiguriert ist, gibt es keine zusätzlichen Konfigurationstasks. Ist TCP/IP nicht konfiguriert, müssen Sie eine TCP/IP-Schnittstelle hinzufügen, eine TCP/IP-Loopback-Schnittstelle hinzufügen und einen Standardleitweg hinzufügen.

Vorgehensweise

1. Eine TCP/IP-Schnittstelle hinzufügen.

Geben Sie in einer Befehlszeile ADDTCPIFC ein und drücken Sie die Eingabetaste. Geben Sie dann die **IP-Adresse** und die **Leitungsbeschreibung** sowie eine **Teilnetzmaske** der Maschine an und drücken Sie erneut die Eingabetaste.

```

Add TCP/IP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . 19.22.11.55
Line description . . . . . TOKENRINGL Name, *LOOPBACK
Subnet mask . . . . . 255.255.0.0
Type of service . . . . . *NORMAL *MINDELAY, *MAXTHRPUT..
Maximum transmission unit . . . *LIND 576-16388, *LIND
Autostart . . . . . *YES *YES, *NO
PVC logical channel identifier 001-FFF
+ for more values
X.25 idle circuit timeout . . . 60 1-600
X.25 maximum virtual circuits . 64 0-64
X.25 DDN interface . . . . . *NO *YES, *NO
TRLAN bit sequencing . . . . . *MSB *MSB, *LSB

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

```

2. Eine TCP/IP-Loopback-Schnittstelle hinzufügen.

Geben Sie in der Befehlszeile ADDTCPIFC ein und drücken Sie die Eingabetaste. Geben Sie dann die Werte für **IP-Adresse**, **Leitungsbeschreibung** und **Teilnetzmaske** an.

```

Add TCP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . 127.0.0.1
Line description . . . . . *LOOPBACK      Name, *LOOPBACK
Subnet mask . . . . . 255.0.0.0
Type of service . . . . . *NORMAL        *MINDELAY, *MAXTHRPUT..
Maximum transmission unit . . . *LIND      576-16388, *LIND
Autostart . . . . . *YES             *YES, *NO
PVC logical channel identifier . . . . . 001-FFF
+ for more values
X.25 idle circuit timeout . . . 60        1-600
X.25 maximum virtual circuits . 64        0-64
X.25 DDN interface . . . . . *NO        *YES, *NO
TRLAN bit sequencing . . . . . *MSB       *MSB, *LSB

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

```

3. Fügen Sie eine Standardroute hinzu.

Geben Sie in der Befehlszeile ADDTCP RTE ein und drücken Sie die Eingabetaste. Geben Sie dann die entsprechenden Werte für Ihr Netzwerk an und drücken Sie die Eingabetaste, um einen Standardleit-
wegeintrag zu erstellen.

```

Add TCP Route (ADDTCP RTE)

Type choices, press Enter.

Route destination . . . . . *DFTRROUTE
Subnet mask . . . . . *NONE
Type of service . . . . . *NORMAL        *MINDELAY, *MAXTHRPUT.
Next hop . . . . . 19.2.3.4
Maximum transmission unit . . . 576        576-16388, *IFC

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Command prompting ended when user pressed F12.

```

Nächste Schritte

Die TCP-Verbindung ist nun eingerichtet. Sie können die Konfiguration wie in „[Kanäle unter IBM i konfigurieren](#)“ auf Seite 26 beschrieben abschließen.

Zugehörige Tasks

„[LU 6.2 -Verbindung unter IBM i herstellen](#)“ auf Seite 16

Zum Herstellen einer LU 6.2 -Verbindung unter IBM i müssen Sie den lokalen Knoten konfigurieren und ihn mit einem Partnerknoten verbinden.

Kanäle unter IBM i konfigurieren

Um IBM MQ für die Beispielkonfiguration unter IBM i zu konfigurieren, müssen Sie die grundlegenden Konfigurationsschritte für den Warteschlangenmanager ausführen und anschließend die Sender- und Empfängerkanäle konfigurieren.

Informationen zu diesem Vorgang

Mit dem Befehl **WRKMQMQ** können Sie das Konfigurationsmenü von IBM MQ anzeigen.

Starten Sie den TCP-Kanallistener mit dem Befehl **STRMQMLSR**.

Starten Sie jeden Senderkanal mit dem Befehl STRMQMCHL CHLNAME(*channel_name*).

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

Vorgehensweise

1. Mit diesem Befehl wird ein Warteschlangenmanager erstellt.

a) Geben Sie CRTMQM ein und drücken Sie die Eingabetaste.

```
                Create Message Queue Manager (CRTMQM)

Type choices, press Enter.
Message Queue Manager name . . .
Text 'description' . . . . . *BLANK
Trigger interval . . . . . 999999999      0-999999999
Undelivered message queue . . . *NONE
Default transmission queue . . . *NONE
Maximum handle limit . . . . . 256          1-999999999
Maximum uncommitted messages . . 1000      1-10000
Default Queue manager . . . . . *NO        *YES, *NO

                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

b) Geben Sie im Feld **Name des Nachrichtenwarteschlangenmanagers** AS400 und im Feld **Warteschlange für nicht zugestellte Nachrichten** DEAD.LETTER.QUEUE ein und drücken Sie die Eingabetaste.

c) Starten Sie den Warteschlangenmanager, indem Sie STRMQM MQMNAME(AS400) eingeben.

d) Erstellen Sie die Warteschlange für nicht zugestellte Nachrichten mit den folgenden Parametern:

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

Weitere Informationen und ein Beispiel zum Definieren einer Warteschlange finden Sie in Schritt „2“ auf Seite 27.

2. Definieren Sie eine Warteschlange.

a) Geben Sie in der Befehlszeile den Befehl CRTMQMQ ein.

```

Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . .
Queue type . . . . .          *ALS, *LCL, *RMT

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

```

- b) Füllen Sie die beiden Felder in dieser Anzeige aus und drücken Sie die Eingabetaste.
 Anschließend wird eine weitere Anzeige mit Eingabefeldern für die anderen Parameter aufgerufen.
 Für alle anderen Warteschlangenattribute können Sie die Standardeinstellungen übernehmen.
3. Definieren Sie einen Kanal.
- a) Geben Sie CRTMQMCHL in die Befehlszeile ein, um die Anzeige **Create MQM-Kanal** aufzurufen.

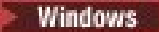
```

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

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

```

- b) Füllen Sie die beiden Felder in dieser Anzeige aus und drücken Sie die Eingabetaste.
 Anschließend erscheint eine weitere Anzeige, in der Sie die Werte für die anderen zuvor angegebenen Parameter angeben können. Für alle anderen Kanalattribute können Sie die Standardeinstellungen übernehmen.
4. Konfigurieren Sie die Kanäle für die Beispielkonfiguration.
 Weitere Informationen zu den in den folgenden Beispielen verwendeten Parametern finden Sie unter „Kanal Konfigurationsparameter für IBM i“ auf Seite 30.

 Diese Beispiele gelten für die Verbindung von IBM MQ unter IBM i mit IBM MQ unter Windows. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie

die entsprechenden Werte aus den Tabellen in „Kanalkonfigurationsparameter für IBM i“ auf Seite 30 anstelle der Werte für Windows.

a) Definieren Sie den Senderkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden:

```

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

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

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

```

- TCP verwenden:

```

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

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

Sender Channel
  Channel Name :      AS400.WINNT.TCP    H
  Channel Type  :      *SDR
  Transport type :      *TCP
  Connection name :      WINNT.tcpip.hostname
  Transmission queue :      WINNT        F

```

b) Definieren Sie den Empfängerkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden:

```

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

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

```

- TCP verwenden:

```

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

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

```

Die Parameter, die zum Konfigurieren der Kanäle für die Beispielkonfiguration unter IBM i erforderlich sind.

Schritt „4“ auf Seite 28 von „Kanäle unter IBM i konfigurieren“ auf Seite 26 beschreibt die Konfiguration, die auf dem IBM i -Warteschlangenmanager ausgeführt werden muss, um den in „Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten“ auf Seite 5 beschriebenen Kanal zu implementieren. Die Beispiele in „Kanäle unter IBM i konfigurieren“ auf Seite 26 beziehen sich auf die Verbindung von IBM MQ for IBM i und IBM MQ for Windows. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie die Werte aus der entsprechenden Tabelle anstelle der Werte für Windows.

Anmerkung:

1. Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.
2. Der Pingbefehl für IBM MQ -Kanäle (**PNGMQMCHL**) wird interaktiv ausgeführt, während das Starten eines Kanals bewirkt, dass ein Batch-Job übergeben wird. Wenn der Pingbefehl eines Kanals erfolgreich ausgeführt wird, der Kanal jedoch nicht startet, sind möglicherweise das Netz und die IBM MQ-Definitionen korrekt, nicht jedoch die IBM i-Umgebung für den Batch-Job. Stellen Sie beispielsweise sicher, dass QSYS2 nicht nur in Ihre persönliche Bibliotheksliste, sondern auch in den Systemteil der Bibliotheksliste aufgenommen wird.

Weitere Informationen und Beispiele zum Erstellen der in den Tabellen aufgelisteten Objekte finden Sie unter „Kanäle unter IBM i konfigurieren“ auf Seite 26.

Definition für lokalen Knoten

Tabelle 13. Konfigurationsbeispiele für die Definition des lokalen Knotens

ID	Parametername	Referenz	Verwendetes Beispiel
A	Name des Warteschlangenmanagers		AS400
B	Name der lokalen Warteschlange		AS400.LOCALQ

Verbindung zu IBM MQ unter Windows**Windows**

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für Windows“ auf Seite 43 verwendeten Werten wie angegeben entsprechen.

Tabelle 14. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Windows

ID	Parametername	Referenz	Verwendetes Beispiel
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)		AS400.WINNT.SNA
H	Name des Senderkanals (TCP/IP)		AS400.WINNT.TCP
I	Name des Empfängerkanals (SNA)	G	WINNT.AS400.SNA
J	Name des Empfängerkanals (TCP/IP)	H	WINNT.AS400.TCP

Verbindung zu IBM MQ unter AIX

AIX

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für AIX“ auf Seite 13 verwendeten Werten wie angegeben entsprechen.

Tabelle 15. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter AIX

ID	Parametername	Referenz	Verwendetes Beispiel
C	Name des fernen Warteschlangenmanagers	A	AIX
D	Name der fernen Warteschlange		AIXREMOTEQ
E	Name der Warteschlange auf fernem System	B	AIX.LOCALQ
F	Name der Übertragungswarteschlange		AIX
G	Name des Senderkanals (SNA)		AS400.AIX.SNA
H	Name des Senderkanals (TCP/IP)		AS400.AIXTCP-
I	Name des Empfängerkanals (SNA)	G	AIX.AS400.SNA
J	Name des Empfängerkanals (TCP)	H	AIX.AS400.TCP

Verbindung zu IBM MQ unter Linux

Linux

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für Linux“ auf Seite 36 verwendeten Werten wie angegeben entsprechen.

Tabelle 16. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Linux

ID	Parametername	Referenz	Verwendetes Beispiel
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)		AS400.LINUX.SNA
H	Name des Senderkanals (TCP/IP)		AS400.LINUX.TCP
I	Name des Empfängerkanals (SNA)	G	LINUX.AS400.SNA
J	Name des Empfängerkanals (TCP/IP)	H	LINUX.AS400.TCP

Verbindung zu IBM MQ for z/OS

z/OS

Die Werte in diesem Abschnitt der Tabelle müssen den in „Channel configuration parameters for z/OS“ auf Seite 48 verwendeten Werten wie angegeben entsprechen.

Tabelle 17. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS

ID	Parametername	Referenz	Verwendetes Beispiel
C	Name des fernen Warteschlangenmanagers	A	MVS
D	Name der fernen Warteschlange		MVS.REMOTEQ

Tabelle 17. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS (Forts.)

ID	Parametername	Referenz	Verwendetes Beispiel
E	Name der Warteschlange auf fernem System	B	MVS.LOCALQ
F	Name der Übertragungswarteschlange		MVS
G	Name des Senderkanals (SNA)		AS400.MVS.SNA
H	Name des Senderkanals (TCP)		AS400.MVS.TCP
I	Name des Empfängerkanals (SNA)	G	MVS.AS400.SNA
J	Name des Empfängerkanals (TCP)	H	MVS.AS400.TCP

Verbindung zu einem VSE/ESA -System

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

Tabelle 18. Konfigurationsbeispiele für die Verbindung zu einem VSE/ESA -System

ID	Parametername	Referenz	Verwendetes Beispiel
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		AS400.VSE.SNA
I	Name des Empfängerkanals	G	VSE.AS400.SNA

Linux **Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Linux einrichten**


Dieses Beispiel zeigt, wie Sie Kommunikationsverbindungen von IBM MQ unter Linux zu IBM MQ auf einer anderen Plattform einrichten und einen funktionierenden Kanal zu dieser Plattform einrichten.

Vorbereitende Schritte

Hintergrundinformationen zu diesem Beispiel und seiner Verwendung finden Sie unter „[Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten](#)“ auf Seite 5 und „[Verwendung der plattformübergreifenden Kommunikationsbeispiele](#)“ auf Seite 7.

Informationen zu diesem Vorgang

In diesem Beispiel wird die plattformübergreifende Kommunikation zwischen IBM MQ unter Linux und den folgenden Plattformen beschrieben:

-  Windows
-  AIX
-  IBM i
-  z/OS

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

Stellen Sie eine Netzverbindung mit LU 6.2 oder TCP her.

Anmerkung: Für TCP verwenden einige Linux -Distributionen jetzt den erweiterten inet-Dämon (XINETT) anstelle des inet-Dämons (INETD). Die folgenden Anweisungen dienen zum Aufbauen einer TCP-Verbindung mit dem inet-Dämon oder dem erweiterten inet-Dämon.

Vorgehensweise

1. Netzverbindung mit LU6.2 herstellen

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

Die neuesten Informationen zum Konfigurieren von SNA über TCP/IP finden Sie im Administratorhandbuch für Ihre Version von Linux in der folgenden Dokumentation: [Communications Server for Data Center Deployment on Linux library](#).

2. TCP-Verbindung mit dem inet-Dämon (INETD) herstellen

a) 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.

b) Bearbeiten Sie die Datei `/etc/inetd.conf`.

Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

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

c) Ermitteln Sie die Prozess-ID von InetD durch Eingabe folgenden Befehls:

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

d) Führen Sie folgenden Befehl aus:

```
kill -1 inetd processid
```

Wenn Sie mehrere Warteschlangenmanager auf Ihrem System haben und daher mehrere Services benötigen, müssen Sie für jeden zusätzlichen Warteschlangenmanager eine Zeile zu `/etc/services` und `inetd.conf` hinzufügen.

For example:

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

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

So lässt sich vermeiden, dass Fehlermeldungen ausgegeben werden, wenn eine Beschränkung für die Anzahl der ausstehenden Verbindungsanforderungen gilt, die in der Warteschlange eines einzelnen TCP-Ports stehen können. Informationen zur Anzahl der ausstehenden Verbindungsanforderungen finden Sie im Abschnitt [Rückstandsoption des TCP-Empfangsprogramms verwenden](#).

Der InetD-Prozess kann unter Linux die Rate der an einem TCP-Anschluss eingehenden Verbindungen einschränken. Die Standardrate sind 40 Verbindungen mit einem Intervall von 60 Sekunden. Wenn Sie eine höhere Rate benötigen, geben Sie eine neue Beschränkung für die Anzahl der

eingehenden Verbindungen in einem 60-Sekunden-Intervall an, indem Sie einen Punkt (.) gefolgt von dem neuen Grenzwert an den Parameter 'nowait' des betreffenden Service in 'inetd.conf' anhängen. Geben Sie beispielsweise für einen Grenzwert von 500 Verbindungen in einem 60-Sekunden-Intervall Folgendes ein:

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

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

3. TCP-Verbindung mit dem erweiterten inet-Dämon (XINETD) herstellen

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.

a) 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
```

Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

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

b) Erstellen Sie eine Datei namens IBM MQ im XINETD-Konfigurationsverzeichnis `/etc/xinetd.d`, indem Sie der Datei die folgende Zeilengruppe hinzufügen:

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

c) Starten Sie den erweiterten inet-Dämon erneut, indem Sie den folgenden Befehl eingeben:

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

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

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

```
cps = 500 60
```

4. Schließen Sie die Konfiguration ab, nachdem die TCP/IP-Verbindung hergestellt wurde.

Wechseln Sie zu „Kanäle unter Linux konfigurieren“ auf Seite 35.

Um IBM MQ für die Beispielkonfiguration unter Linux zu konfigurieren, müssen Sie die grundlegenden Konfigurationsschritte für den Warteschlangenmanager ausführen und anschließend die Sender- und Empfängerkanäle konfigurieren.

Vorbereitende Schritte

Stellen Sie vor Beginn des Prozesses sicher, dass Sie zuerst die mqm-Benutzer-ID und die mqm-Gruppe erstellt haben, und legen Sie das Kennwort fest.

Starten Sie einen beliebigen Kanal über den Befehl:

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

Informationen zu diesem Vorgang

Anmerkungen:

1. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert, wobei `MQ_INSTALLATION_PATH` das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
2. Fehlerprotokolle werden in `/var/mqm/qmgrs/qmgrname` /-Fehlern 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.

Vorgehensweise

1. Richten Sie die Basiskonfiguration ein:

- a) Erstellen Sie den WS-Manager und eine Gruppe von Standardobjekten in der UNIX -Eingabeaufforderung mit dem Befehl:

```
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.

- b) Starten Sie den Warteschlangenmanager über die UNIX -Eingabeaufforderung mit dem Befehl:

```
strmqm linux
```

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

2. Konfigurieren Sie die Kanäle für die Beispielkonfiguration.

Weitere Informationen zu den in den folgenden Beispielen verwendeten Parametern finden Sie unter „Kanalkonfigurationsparameter für Linux“ auf Seite 36. In jedem Fall zeigt das Beispiel den MQSC-Befehl. Entweder starten Sie **runmqsc** in einer Linux-Befehlszeile und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Windows

Diese Beispiele gelten für die Verbindung von IBM MQ unter Linux mit IBM MQ unter Windows. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie

die entsprechenden Werte aus den Tabellen in „[Kanalkonfigurationsparameter für Linux](#)“ auf Seite 36 anstelle der Werte für Windows.

a) Definieren Sie den Senderkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden

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

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

def chl (LINUX.WINNT.SNA) chltype(sdr) +      G
  trptype(lu62) +
  conname('WINNTCPIC') +                      14
  xmitq(WINNT) +                               F
  replace
```

- TCP verwenden

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

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

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

b) Definieren Sie den Empfängerkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden:

```
def q1 (LINUX.LOCALQ) replace                  B

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

- TCP verwenden:

```
def q1 (LINUX.LOCALQ) replace                  B

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

Linux Kanalkonfigurationsparameter für Linux

Die Parameter, die zum Konfigurieren der Kanäle für die Beispielkonfiguration unter Linux erforderlich sind.

Schritt „2“ auf Seite 35 von „[Kanäle unter Linux konfigurieren](#)“ auf Seite 35 beschreibt die Konfiguration, die auf dem Linux -Warteschlangenmanager ausgeführt werden muss, um den in „[Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten](#)“ auf Seite 5 beschriebenen Kanal zu implementieren. Die Beispiele in „[Kanäle unter Linux konfigurieren](#)“ auf Seite 35 beziehen sich auf die Verbindung von IBM MQ for IBM i und IBM MQ for Windows. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie die Werte aus der entsprechenden Tabelle anstelle der Werte für Windows.

Anmerkung: Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.

Definition für lokalen Knoten

Tabelle 19. Konfigurationsbeispiele für die Definition des lokalen Knotens

ID	Parametername	Referenz	Verwendetes Beispiel
A	Name des Warteschlangenmanagers		LINUX
B	Name der lokalen Warteschlange		LINUX.LOCALQ

Verbindung zu IBM MQ unter Windows

Windows

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfigurationsparameter für Windows“ auf Seite 43 angegebenen entsprechen.

Tabelle 20. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter Windows

ID	Parametername	Referenz	Verwendetes Beispiel
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 IBM MQ unter AIX

AIX

Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanäle unter AIX konfigurieren“ auf Seite 10 angegebenen entsprechen.

Tabelle 21. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter AIX

ID	Parametername	Referenz	Verwendetes Beispiel
C	Name des fernen Warteschlangenmanagers	A	AIX
D	Name der fernen Warteschlange		AIX.REMOTEQ
E	Name der Warteschlange auf fernem System	B	AIX.LOCALQ
F	Name der Übertragungswarteschlange		AIX
G	Name des Senderkanals (SNA)		.LINUX.AIX.SNA

Tabelle 21. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter AIX (Forts.)

ID	Parametername	Referenz	Verwendetes Beispiel
H	Name des Senderkanals (TCP/IP)		LINUX.AIX.TCP
I	Name des Empfängerkanals (SNA)	G	AIX.LINUX.SNA
J	Name des Empfängerkanals (TCP/IP)	H	AIX.LINUX.TCP

Verbindung zu IBM MQ for IBM i

IBM i

Die Werte in diesem Abschnitt der Tabelle müssen den in „[Kanalkonfigurationsparameter für IBM i](#)“ auf Seite 30 angegebenen entsprechen.

Tabelle 22. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter IBM i

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
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)		LINUX.AS400.SNA
H	Name des Senderkanals (TCP)		LINUX.AS400.TCP
I	Name des Empfängerkanals (SNA)	G	AS400.LINUX.SNA
J	Name des Empfängerkanals (TCP)	H	AS400.LINUX.TCP

Verbindung zu IBM MQ for z/OS

z/OS

Die Werte in diesem Abschnitt der Tabelle müssen den in „[Channel configuration parameters for z/OS](#)“ auf Seite 48 angegebenen entsprechen.

Tabelle 23. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS

ID	Parametername	Referenz	Verwendetes Beispiel
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
J	Name des Empfängerkanals (TCP)	H	MVS.LINUX.TCP

Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange



Die Werte in diesem Abschnitt der Tabelle müssen den in „[Shared channel configuration parameters](#)“ auf Seite 57 angegebenen entsprechen.

Tabelle 24. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange

ID	Parametername	Referenz	Verwendetes Beispiel
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)		LINUX.QSG.SNA
H	Name des Senderkanals (TCP)		LINUX.QSG.TCP
I	Name des Empfängerkanals (SNA)	G	QSG.LINUX.SNA
J	Name des Empfängerkanals (TCP)	H	QSG.LINUX.TCP

Windows **Beispiel: Plattformübergreifende IBM MQ -Kommunikation unter Windows einrichten**

Dieses Beispiel zeigt, wie Sie Kommunikationsverbindungen von IBM MQ unter Windows zu IBM MQ auf einer anderen Plattform einrichten und einen funktionierenden Kanal zu dieser Plattform einrichten.

Vorbereitende Schritte

Hintergrundinformationen zu diesem Beispiel und seiner Verwendung finden Sie unter „[Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten](#)“ auf Seite 5 und „[Verwendung der plattformübergreifenden Kommunikationsbeispiele](#)“ auf Seite 7.

Informationen zu diesem Vorgang

In diesem Beispiel wird die plattformübergreifende Kommunikation zwischen IBM MQ unter Windows und den folgenden Plattformen beschrieben:

- AIX
- IBM i
- Linux
- z/OS

Vorgehensweise

1. Stellen Sie mithilfe von LU6.2 eine Netzverbindung her.

Informationen zur Konfiguration von AnyNet SNA über TCP/IP finden Sie unter [AnyNet SNA über TCP/IP und Communications Server für Windows](#).

2. Richten Sie eine Netzverbindung mit TCP ein.

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

Sie müssen den Listener explizit starten, bevor Kanäle gestartet werden. Es ermöglicht empfangenden Kanälen das automatische Starten nach dem Erhalt einer Anforderung von einem Senderkanal für eingehende Nachrichten.

Verwenden Sie den folgenden Befehl, um das TCP-Empfangsprogramm von IBM MQ zu starten:

```
runmqclsr -t tcp
```

3. Richten Sie mit NetBIOS eine Netzverbindung ein.

- a) Geben Sie an jedem Ende des Kanals den lokalen NetBIOS -Namen an, der von den IBM MQ -Kanalprozessen in der Konfigurationsdatei qm.inides Warteschlangenmanagers verwendet wird. Die NETBIOS-Zeilengruppe in Windows kann an der Sendeseite wie folgt aussehen:

```
NETBIOS:  
LocalName=WNTNETB1
```

und am empfangenden Ende wie folgt aussehen:

```
NETBIOS:  
LocalName=WNTNETB2
```

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

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

Geben Sie die richtige LAN-Adapternummer in der NETBIOS-Zeilengruppe der Windows-Registrierungsdatenbank an. For example:

```
NETBIOS:  
AdapterNum=1
```

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

```
SET MQNAME=WNTNETB1I
```

Anmerkung: Dieser Name muss eindeutig sein.

- d) Definieren Sie an der Sendeseite einen Kanal mit dem von der anderen Kanalseite verwendeten NetBIOS-Namen.

For example:

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

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

- e) Definieren Sie an der Empfangsseite den zugehörigen Empfängerkanal.

For example:

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

f) Starten Sie den Kanalinitiator.

Jeder neue Kanal wird als Thread und nicht als neuer Prozess gestartet:

```
runmqchi
```

g) Starten Sie auf der Empfangsseite das Empfangsprogramm von IBM MQ:

```
runmqldr -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](#).

4. Schließen Sie die Konfiguration ab, nachdem die Netzverbindung hergestellt wurde. Weitere Informationen finden Sie unter [„Kanäle unter Windows konfigurieren“](#) auf Seite 41.

Kanäle unter Windows konfigurieren

Um IBM MQ für die Beispielkonfiguration unter Windows zu konfigurieren, müssen Sie die grundlegenden Konfigurationsschritte für den Warteschlangenmanager ausführen und anschließend die Sender- und Empfängerkanäle konfigurieren.

Informationen zu diesem Vorgang

Anmerkungen:

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

```
AMQSBCG q_name qmgr_name
```

zeigt die Inhalte der Warteschlange *q_name* an, die im Warteschlangenmanager *qmgr_name* definiert ist.

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

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 IBM 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.

Vorgehensweise

1. Führen Sie die folgenden Schritte aus, um die Basiskonfiguration über die Eingabeaufforderung einzurichten:

- a) Erstellen Sie den Warteschlangenmanager und eine Gruppe von Standardobjekten mit dem folgenden Befehl:

```
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.

- b) Starten Sie den Warteschlangenmanager über den folgenden Befehl:

```
strmqm winnt
```

Dabei ist *winnt* der Name, der dem Warteschlangenmanager bei seiner Erstellung zugeordnet wurde.

2. Konfigurieren Sie die Kanäle für die Beispielkonfiguration.

Weitere Informationen zu den in den folgenden Beispielen verwendeten Parametern finden Sie unter „Kanalkonfigurationsparameter für Windows“ auf Seite 43. In jedem Fall zeigt das Beispiel den MQSC-Befehl. Entweder starten Sie **runmqsc** in einer Linux-Befehlszeile und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen. Diese Beispiele beziehen sich auf die Verbindung zwischen IBM MQ for Windows und IBM MQ for AIX. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie die entsprechenden Werte aus den Tabellen in „Kanalkonfigurationsparameter für Windows“ auf Seite 43 anstelle der Werte für IBM MQ for AIX.

- a) Definieren Sie den Senderkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden

```
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
  trtype(lu62) +
  conname(AIXCPIC) +                         18
  xmitq(AIX) +                               F
  replace
```

- TCP verwenden

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

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

def chl (WINNT.AIX.TCP) chltype(sdr) +      H
  trtype(tcp) +
  conname(remote_tcpip_hostname) +
```

```
xmitq(AIX) + F
replace
```

b) Definieren Sie den Empfängerkanal wie in den folgenden Beispielen gezeigt:

- SNA verwenden:

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

- TCP verwenden:

```
def q1 (WINNT.LOCALQ) replace B
def chl (AIX.WINNT.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

Nächste Schritte

Automatischer Start

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

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

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

Kanäle als Prozesse oder Threads ausführen

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

Windows Kanalkonfigurationsparameter für Windows

Die Parameter, die zum Konfigurieren der Kanäle für die Beispielkonfiguration unter Windows erforderlich sind.

Schritt „2“ auf Seite 42 von „Kanäle unter Windows konfigurieren“ auf Seite 41 beschreibt die Konfiguration, die auf dem Linux -Warteschlangenmanager ausgeführt werden muss, um den in „Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten“ auf Seite 5 beschriebenen Kanal zu implementieren. Die Beispiele in „Kanäle unter Windows konfigurieren“ auf Seite 41 beziehen sich auf die Verbindung von IBM MQ for Windows und IBM MQ for AIX. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, verwenden Sie die Werte aus der entsprechenden Tabelle anstelle der Werte für Windows.

Anmerkung: Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.

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

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

Definition für lokalen Knoten

Tabelle 25. Konfigurationsbeispiele für die Definition des lokalen Knotens

ID	Parametername	Referenz	Verwendetes Beispiel
A	Name des Warteschlangenmanagers		WINNT
B	Name der lokalen Warteschlange		WINNT.LOCALQ

Verbindung zu IBM MQ unter AIX



Die Werte in diesem Abschnitt der Tabelle müssen den in „[Kanalkonfigurationsparameter für AIX](#)“ auf Seite 13 angegebenen entsprechen.

Tabelle 26. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter AIX

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
C	Name des fernen Warteschlangenmanagers	A	AIX
D	Name der fernen Warteschlange		AIXREMOTEQ
E	Name der Warteschlange auf fernem System	B	AIX.LOCALQ
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
J	Name des Empfängerkanals (TCP)	H	AIX.WINNT.TCP

Verbindung zu IBM MQ unter IBM i



Die Werte in diesem Abschnitt der Tabelle müssen den in „[Kanalkonfigurationsparameter für IBM i](#)“ auf Seite 30 angegebenen entsprechen.

Tabelle 27. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter IBM i

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
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

Tabelle 27. Konfigurationsbeispiele für die Verbindung zu IBM MQ unter IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
H	Name des Senderkanals (TCP)		WINNT.AS400.TCP
I	Name des Empfängerkanals (SNA)	G	AS400.WINNT.SNA
J	Name des Empfängerkanals (TCP)	H	AS400.WINNT.TCP

Verbindung zu IBM MQ for z/OS



Die Werte in diesem Abschnitt der Tabelle müssen den in „Channel configuration parameters for z/OS“ auf Seite 48 angegebenen entsprechen.

Tabelle 28. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS

ID	Parametername	Referenz	Verwendetes Beispiel
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
J	Name des Empfängerkanals (TCP)	H	MVS.WINNT.TCP

Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange



Die Werte in diesem Abschnitt der Tabelle müssen den in „Shared channel configuration parameters“ auf Seite 57 angegebenen entsprechen.

Tabelle 29. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange

ID	Parametername	Referenz	Verwendetes Beispiel
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)		WINNT.QSG.SNA
H	Name des Senderkanals (TCP)		WINNT.QSG.TCP
I	Name des Empfängerkanals (SNA)	G	QSG.WINNT.SNA

Tabelle 29. Konfigurationsbeispiele für die Verbindung zu IBM MQ for z/OS mithilfe von Gruppen mit gemeinsamer Warteschlange (Forts.)

ID	Parametername	Referenz	Verwendetes Beispiel
J	Name des Empfängerkanals (TCP)	H	QSG.WINNT.TCP

Example: setting up IBM MQ cross-platform communication on z/OS

This example shows how to set up communication links from IBM MQ on z/OS to IBM MQ on another platform and establish a working channel to that platform.

Before you begin

For background information about this example and how to use it, see [“Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten”](#) on page 5 and [“Verwendung der plattformübergreifenden Kommunikationsbeispiele”](#) on page 7.

About this task

This example covers setting up cross platform communication from IBM MQ on z/OS to the following platforms:

-  Windows
-  AIX
-  Linux
-  IBM i
- VSE/ESA

You can also connect any of the following:

- z/OS to z/OS
- z/OS to MVS
- MVS to MVS

Procedure

1. Establish a network connection.

- Establishing an LU 6.2 connection

For the latest information about configuring SNA over TCP/IP, refer to the following online IBM documentation: [Communications Server for z/OS](#).

- Establishing a TCP connection

Alter the queue manager object to use the correct distributed queuing parameters using the following command. You must add the name of the TCP address space to the TCPNAME queue manager attribute.

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

The TCP connection is now established. You are ready to complete the configuration.

2. Configure the channels.

See “Configuring the channels on IBM MQ for z/OS ” on page 47 for details on how you configure the channels.

z/OS Configuring the channels on IBM MQ for z/OS

To configure IBM MQ for the example configuration on z/OS, start and configure the channels and listeners.

Procedure

1. Start the channel initiator using the command:

```
/cpf START CHINIT 1
```

2. Start an LU 6.2 listener using the command:

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

The LUNAME of M1 refers to the symbolic name you gave your LU (5). You must specify TRPTYPE(LU62), otherwise the listener assumes that you want TCP.

3. Start a TCP listener using the command:

```
/cpf START LSTR
```

If you want to use a port other than 1414 (the default IBM MQ port), use the command:

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

IBM MQ channels do not initialize successfully if the channel negotiation detects that the message sequence number is different at each end. You might need to reset these channels manually.

4. Configure the channels for the example configuration.

For more information about the parameters used in the following examples, see “Channel configuration parameters for z/OS” on page 48. These examples are for connecting IBM MQ for z/OS and IBM MQ for Windows. To connect to IBM MQ on another platform use the values from the appropriate table in “Channel configuration parameters for z/OS” on page 48 instead of the values for Windows.

- a) Define the sender channel as shown in the following example:s

For LU 6.2:

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

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

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

For TCP:

```
Local Queue
```

```

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

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

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

```

b) Define the receiver channel as shown in the following examples:

For LU 6.2:

```

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

Receiver Channel
Channel name : WINNT.MVS.SNA I

```

For TCP:

```

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

Receiver Channel
Channel name : WINNT.MVS.TCP J

```

Channel configuration parameters for z/OS

The parameters needed to configure the channels for the example configuration on z/OS.

Step “4” on page 47 of “Configuring the channels on IBM MQ for z/OS ” on page 47 describes the configuration to be performed on the z/OS queue manager to implement the channel described in “[Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten](#)” on page 5. The examples in “[Configuring the channels on IBM MQ for z/OS](#) ” on page 47 are for connecting IBM MQ for z/OS and IBM MQ for Windows. To connect to IBM MQ on another platform use the values from the appropriate table in place of the values for Windows.

Note: The words in **bold** are suggested values and reflect the names of IBM MQ objects used throughout these examples. You can change them in your product installation but, if you do, make sure that you use your own values when working through the examples in this section

Definition for local node

Table 30. Configuration examples for the definition for the local node			
ID	Parameter Name	Reference	Example Used
A	Queue Manager Name		MVS
B	Local queue name		MVS.LOCALQ

Connection to IBM MQ on Windows



The values in this section of the table must match the values used in [“Kanalkonfigurationsparameter für Windows”](#) on page 43, as indicated.

Table 31. Configuration examples for connecting to IBM MQ on Windows

ID	Parameter Name	Reference	Example Used
C	Remote queue manager name	A	WINNT
D	Remote queue name		WINNT.REMOTEQ
E	Queue name at remote system	B	WINNT.LOCALQ
F	Transmission queue name		WINNT
G	Sender (LU 6.2) channel name		MVS.WINNT.SNA
H	Sender (TCP) channel name		MVS.WINNT.TCP
I	Receiver (LU 6.2) channel name	G	WINNT.MVS.SNA
J	Receiver (TCP/IP) channel name	H	WINNT.MVS.TCP

Connection to IBM MQ on AIX



The values in this section of the table must match the values used in [“Kanalkonfigurationsparameter für AIX”](#) on page 13, as indicated.

Table 32. Configuration examples for connecting to IBM MQ on AIX

ID	Parameter Name	Reference	Example Used
Connection to IBM MQ for AIX			
C	Remote queue manager name	A	AIX
D	Remote queue name		AIX.REMOTEQ
E	Queue name at remote system	B	AIX.LOCALQ
F	Transmission queue name		AIX
G	Sender (LU 6.2) channel name		MVS.AIX.SNA
H	Sender (TCP/IP) channel name		MVS.AIX.TCP
I	Receiver (LU 6.2) channel name	G	AIX.MVS.SNA
J	Receiver (TCP/IP) channel name	H	AIX.MVS.TCP

Connection to IBM MQ on Linux



The values in this section of the table must match the values used in [“Kanalkonfigurationsparameter für Linux”](#) on page 36, as indicated.

Table 33. Configuration examples for connecting to IBM MQ on Linux

ID	Parameter Name	Reference	Example Used
C	Remote queue manager name	A	LINUX
D	Remote queue name		LINUX.REMOTEQ
E	Queue name at remote system	B	LINUX.LOCALQ

Table 33. Configuration examples for connecting to IBM MQ on Linux (continued)

ID	Parameter Name	Reference	Example Used
F	Transmission queue name		LINUX
G	Sender (LU 6.2) channel name		MVS.LINUX.SNA
H	Sender (TCP) channel name		MVS.LINUX.TCP
I	Receiver (LU 6.2) channel name	G	LINUX.MVS.SNA
J	Receiver (TCP/IP) channel name	H	LINUX.MVS.TCP

Connection to IBM MQ on IBM i

IBM i

The values in this section of the table must match the values used in “[Kanalkonfigurationsparameter für IBM i](#)” on page 30, as indicated.

Table 34. Configuration examples for connecting to IBM MQ on IBM i

ID	Parameter name	Reference	Example used
C	Remote queue manager name	A	AS400
D	Remote queue name		AS400.REMOTEQ
E	Queue name at remote system	B	AS400.LOCALQ
F	Transmission queue name		AS400
G	Sender (LU 6.2) channel name		MVS.AS400.SNA
H	Sender (TCP/IP) channel name		MVS.AS400.TCP
I	Receiver (LU 6.2) channel name	G	AS400.MVS.SNA
J	Receiver (TCP/IP) channel name	H	AS400.MVS.TCP

z/OS Example: setting up IBM MQ cross-platform communication on z/OS using QSGs

This example shows how to set up communication links to a queue sharing group (QSG) from IBM MQ on Windows and AIX. You can also connect from z/OS to z/OS.

Before you begin

Setting up communication links from a queue sharing group to a platform other than z/OS is the same as described in “[Example: setting up IBM MQ cross-platform communication on z/OS](#)” on page 46.

For background information about this example and how to use it, see “[Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten](#)” on page 5 and “[Verwendung der plattformübergreifenden Kommunikationsbeispiele](#)” on page 7.

Procedure

1. Establish a network connection using one of the following options.
 - Establish an LU 6.2 connection as described in “[Establishing an LU 6.2 connection into a queue sharing group](#)” on page 51.
 - Establish a TCP connection using Sysplex Distributor as described in “[Establishing a TCP connection using Sysplex Distributor](#)” on page 55.

2. Define some channels to complete the configuration after the connection is established.
See [“Configuring shared channels on IBM MQ for z/OS”](#) on page 55 for details of this process.

Establishing an LU 6.2 connection into a queue sharing group

There are two steps to establish an LU 6.2 connection. Defining yourself to the network and defining a connection to the partner.

About this task

Note: This example is for a connection to a Windows system but the task is the same for other platforms.

Procedure

1. Use VTAM Generic Resources to have one connection name to connect to the queue sharing group.
 - a) SYS1.PARMLIB(APPCPMxx) contains the start-up parameters for APPC. You must add a line to this file to tell APPC where to locate the sideinfo.

This line must be of the form:

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

- b) Add another line to SYS1.PARMLIB(APPCPMxx) to define the local LU name you intend to use for the IBM MQ LU 6.2 group listener.

The line you add must take the form

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

Specify values for ACBNAME (9), TPDATA and GRNAME (10).

The NOSCHED parameter tells APPC that our new LU is not using the LU 6.2 scheduler (ASCH), but has one of its own. TPDATA refers to the Transaction Program data set in which LU 6.2 stores information about transaction programs. Again, IBM MQ does not use this parameter, but it is required by the syntax of the LUADD command.

- c) Start the APPC subsystem with the command:

```
START APPC, SUB=MSTR, APPC=xx
```

where *xx* is the suffix of the PARMLIB member in which you added the LU in step 1.

Note: If APPC is already running, it can be refreshed with the command:

```
SET APPC=xx
```

The effect of this is cumulative, that is, APPC does not lose its knowledge of objects already defined to it in this member or another PARMLIB member.

- d) Add the new LU to a suitable VTAM major node definition. These are typically in SYS1.VTAMLST. The APPL definition will look like the sample shown.

```
        MVSLU APPL  ACBNAME=MVSLU1,      9
                   APPXC=YES,
                   AUTOSES=0,
                   DDRAINL=NALLOW,
                   DLOGMOD=#INTER,      6
                   DMINWML=10,
```

```

DMINWNR=10,
DRESPL=NALLOW,
DSESLIM=60,
LMDENT=19,
MODETAB=MTCICS,
PARSESS=YES,
VERIFY=NONE,
SECACPT=ALREADYV,
SRBEXIT=YES

```

e) Activate the major node.

You can do this activation with the command:

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

f) Add entries defining your LU and generic resource name to the CPI-C side information data set, using the APPC utility program ATBSDLFNU to do so.

Sample JCL is in *thlqual.SCSQPROC(CSQ4SIDE)* (where *thlqual* is the target library high-level qualifier for IBM MQ data sets in your installation.)

The entries you add will look like this example:

```

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

```

g) Alter the queue manager object to use the correct distributed queuing parameters using the following command.

You must specify the local LU (9) assigned to your queue manager in the LUGROUP attribute of the queue manager.

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

2. Define a connection to a partner by adding an entry to the CPI-C side information data set.

a) Add an entry to the CPI-C side information data set to define the connection.

Sample JCL to do this definition is in *thlqual.SCSQPROC(CSQ4SIDE)*.

The entry you add looks like this:

```

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

```

What to do next

The connection is now established. You are ready to complete the configuration.

Go to [“Configuring shared channels on IBM MQ for z/OS”](#) on page 55.

Configuration parameters for an LU 6.2 connection

The following table lists all the parameters required to set up communication from a z/OS system to IBM MQ on another platform.

The steps required to set up an LU 6.2 connection are described in “Establishing an LU 6.2 connection into a queue sharing group” on page 51, with numbered cross-references to the parameters in the example.

Numbers in the Reference column indicate that the value must match that in the appropriate example elsewhere in this section. The examples that follow in this section refer to the values in the ID column. The entries in the Parameter Name column are explained in “Explanation of terms” on page 54.

Definition for local node using generic resources

Table 35. Configuration examples for the definition for the local node using generic resources

ID	Parameter name	Reference	Example used
1	Command prefix		/cpf
2	Network ID		NETID
3	Node name		MVSPU
6	Modename		#INTER
7	Local Transaction Program name		MQSERIES
8	LAN destination address		400074511092
9	Local LU name		MVSLU1
10	Generic resource name		MVSGR
11	Symbolic destination		G1
12	Symbolic destination for generic resource name		G2

Connection to IBM MQ on Windows

Windows

Table 36. Configuration examples for connecting to IBM MQ on Windows using LU 6.2

ID	Parameter name	Reference	Example used
13	Symbolic destination		M3
14	Modename	21	#INTER
15	Remote Transaction Program name	7	MQSERIES
16	Partner LU name	5	WINNTLU
21	Remote node ID	4	05D 30F65

Connection to IBM MQ on AIX

AIX

Table 37. Configuration examples for connecting to IBM MQ on AIX using LU 6.2

ID	Parameter name	Reference	Example used
13	Symbolic Destination		M4
14	Modename	18	#INTER
15	Remote Transaction Program name	6	MQSERIES
16	Partner LU name	4	AIXLU

Explanation of terms

1 Command prefix

This term is the unique command prefix of your IBM MQ for z/OS queue manager subsystem. The z/OS system programmer defines this value at installation time, in SYS1.PARMLIB(IEFSSNss), and can tell you the value.

2 Network ID

The VTAM startup procedure in your installation is partly customized by the ATCSTRxx member of the data set referenced by the DDNAME VTAMLST. The Network ID is the value specified for the NETID parameter in this member. For Network ID, you must specify the name of the NETID that owns the IBM MQ communications subsystem. Your network administrator can tell you the value.

3 Node name

VTAM, being a low-entry network node, does not have a Control Point name for Advanced Peer-to-Peer Networking (APPN) use. It does however have a system services control point name (SSCPNAME). For node name, you must specify the name of the SSCP that owns the IBM MQ communications subsystem. This value is defined in the same ATCSTRxx member as the Network ID. Your network administrator can tell you the value.

9 Local LU name

A logical unit (LU) is software that serves as an interface or translator between a transaction program and the network. It manages the exchange of data between transaction programs. The local LU name is the unique VTAM APPLID of this IBM MQ subsystem. Your network administrator can tell you this value.

11 12 13 Symbolic destination

This term is the name you give to the CPI-C side information profile. You need a side information entry for each LU 6.2 listener.

6 14 Modename

This term is the name given to the set of parameters that control the LU 6.2 conversation. An entry with this name and similar attributes must be defined at each end of the session. In VTAM, this corresponds to a mode table entry. Your network administrator can assign this table entry to you.

7 15 Transaction Program name

IBM MQ applications trying to converse with this queue manager specify a symbolic name for the program to be run at the receiving end. This has been specified in the TPNAME attribute on the channel definition at the sender. For simplicity, wherever possible use a transaction program name of MQSERIES, or in the case of a connection to VSE/ESA, where the length is limited to 4 bytes, use MQTP.

See [Defining an LU6.2 connection for z/OS using APPC/MVS](#) for more information.

8 LAN destination address

This term is the LAN destination address that your partner nodes use to communicate with this host. When you are using a 3745 network controller, it is the value specified in the LOCADD parameter for the line definition to which your partner is physically connected. If your partner nodes use other devices such as 317X or 6611 devices, the address is set during the customization of those devices. Your network administrator can tell you this value.

10 Generic resource name

A generic resource name is a unique name assigned to a group of LU names used by the channel initiators in a queue sharing group.

16 Partner LU name

This term is the LU name of the IBM MQ queue manager on the system with which you are setting up communication. This value is specified in the side information entry for the remote partner.

21 Remote node ID

For a connection to Windows, this ID is the ID of the local node on the Windows system with which you are setting up communication.

Establishing a TCP connection using Sysplex Distributor

You can set up Sysplex distributor to use one connection name to connect to the queue sharing group.

Procedure

1. Define a Distributed DVIPA address as follows:
 - a) Add a DYNAMICXCF statement to the IPCONFIG. This statement is used for inter-image connectivity using dynamically created XCF TCP/IP links.
 - b) Use the VIPADYNAMIC block on each image in the Sysplex.

On the owning image, code a VIPADEFINE statement to create the DVIPA. Then code a VIPADISTRIBUTE statement to distribute it to all other or selected images.

On the backup image, code a VIPABACKUP statement for the DVIPA address.

2. Add the SHAREPORT option for the port to be shared in the PORT reservation list in the PROFILE data set if more than one channel initiator is to be started on any LPAR in the sysplex.

See [PORT statement](#) in the *z/OS Communications Server: IP Configuration Reference* for more information.

When you have completed these steps, the TCP connection is established. You are ready to complete the configuration.

What to do next

Go to [“Configuring shared channels on IBM MQ for z/OS”](#) on page 55.

Configuring shared channels on IBM MQ for z/OS

Configure the shared channel by starting the channel initiator and issuing appropriate commands for your configuration.

About this task

There can be only one instance of the shared channel running at a time. If you try to start a second instance of the channel it fails (the error message varies depending on other factors). The shared synchronization queue tracks the channel status.

Important: IBM MQ channels do not initialize successfully if the channel negotiation detects that the message sequence number is different at each end. You might need to reset this manually.

Procedure

1. Start the channel initiator using the command:

```
/cpf START CHINIT
```

2. Start an LU6.2 group listener using the command:

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

The LUNAME of G1 refers to the symbolic name you gave your LU (11).

3. Use the following command if you are using Virtual IP Addressing using Sysplex Distributor and want to listen on a specific address:

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

4. Configure the channels for the example configuration.

For more information about the parameters used in the following examples, see [“Shared channel configuration parameters”](#) on page 57. These examples are for connecting IBM MQ for z/OS and Windows. To connect to IBM MQ on another platform, use the appropriate values from the tables in [“Shared channel configuration parameters”](#) on page 57 instead of the values for Windows.

a) Define the shared sender channel as shown in the following examples.

Using LU 6.2:

```

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

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

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

```

Using TCP

```

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

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

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

```

b) Define the shared receiver channel as shown in the following examples.

Using LU 6.2:

```

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

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

```

Using TCP:

```

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

```



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

Shared channel configuration parameters

The parameters needed to configure a shared channel for the example configuration on z/OS.

Step “4” on page 55 of [“Configuring shared channels on IBM MQ for z/OS”](#) on page 55 describes the configuration to be performed on the z/OS queue manager to implement the channel described in [“Beispiel: Plattformübergreifende Kommunikation für IBM MQ einrichten”](#) on page 5. The examples in [“Configuring shared channels on IBM MQ for z/OS”](#) on page 55 are for connecting IBM MQ for z/OS and Windows. To connect to IBM MQ on another platform, use the values from the appropriate table in place of the values for Windows.

Note: The words in **bold** are suggested values and reflect the names of IBM MQ objects used throughout these examples. You can change them in your product installation but, if you do, make sure that you use your own values when working through the examples in this section.

Definition for local node

Table 38. Configuration examples for the definition for the local node

ID	Parameter Name	Reference	Example Used
A	Queue Manager Name		QSG
B	Local queue name		QSG.SHAREDQ

Connection to IBM MQ on Windows

The values in this section of the table must match the values used in [“Kanalkonfigurationsparameter für Windows”](#) on page 43, as indicated.

Table 39. Configuration examples for connecting to to IBM MQ on Windows

ID	Parameter name	Reference	Example used
C	Remote queue manager name	A	WINNT
D	Remote queue name		WINNT.REMOTEQ
E	Queue name at remote system	B	WINNT.LOCALQ
F	Transmission queue name		WINNT
G	Sender (LU 6.2) channel name		QSG.WINNT.SNA
H	Sender (TCP) channel name		QSG.WINNT.TCP
I	Receiver (LU 6.2) channel name	G	WINNT.QSG.SNA
J	Receiver (TCP/IP) channel name	H	WINNT.QSG.TCP

Connection to IBM MQ on AIX

The values in this section of the table must match the values used in [“Kanalkonfigurationsparameter für AIX”](#) on page 13, as indicated.

Table 40. Configuration examples for connecting to IBM MQ on AIX

ID	Parameter name	Reference	Example used
C	Remote queue manager name		AIX
D	Remote queue name		AIX.REMOTEQ
E	Queue name at remote system	B	AIX.LOCALQ
F	Transmission queue name		AIX
G	Sender (LU 6.2) channel name		QSG.AIX.SNA
H	Sender (TCP/IP) channel name		QSG.AIX.TCP
I	Receiver (LU 6.2) channel name	G	AIX.QSG.SNA
J	Receiver (TCP/IP) channel name	H	AIX.QSG.TCP

Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS

This example shows how a typical payroll query application that currently uses distributed queuing to transfer small messages between queue managers could be migrated to use queue sharing groups and shared queues.

About this task

Three configurations are described to illustrate the use of distributed queuing, intra-group queuing with shared queues, and shared queues. The associated diagrams show only the flow of data in one direction, that is, from queue manager QMG1 to queue manager QMG3.

Procedure

1. Set up and run Configuration 1.
For more information, see [“Setting up and running configuration 1”](#) on page 59.
2. Set up and run Configuration 2.
For more information, see [“Setting up and running configuration 2”](#) on page 61.
3. Set up and run Configuration 3.
For more information, see [“Setting up and running configuration 3”](#) on page 63.

What to do next

You can expand the example in a number of ways by:

- Using channel triggering as well as application (PAYROLL and PAYROLL.REPLY queue) triggering.
- Configuring for communication using LU6.2.
- Configuring more queue managers to the queue sharing group. Then the server application can be cloned to run on other queue manager instances to provide multiple servers for the PAYROLL query queue.
- Increasing the number of instances of the payroll query requesting application to demonstrate the processing of requests from multiple clients.
- Using security (IGQAUT and IGQUSER).

z/OS Setting up and running configuration 1

Configuration 1 describes how distributed queuing is currently used to transfer messages between queue managers QMG1 and QMG3.

About this task

Configuration 1 shows a distributed queuing system that is used to transfer messages received by queue manager QMG1 from the payroll query to queue manager QMG2 and then finally on to queue manager QMG3, to be sent to the payroll server.

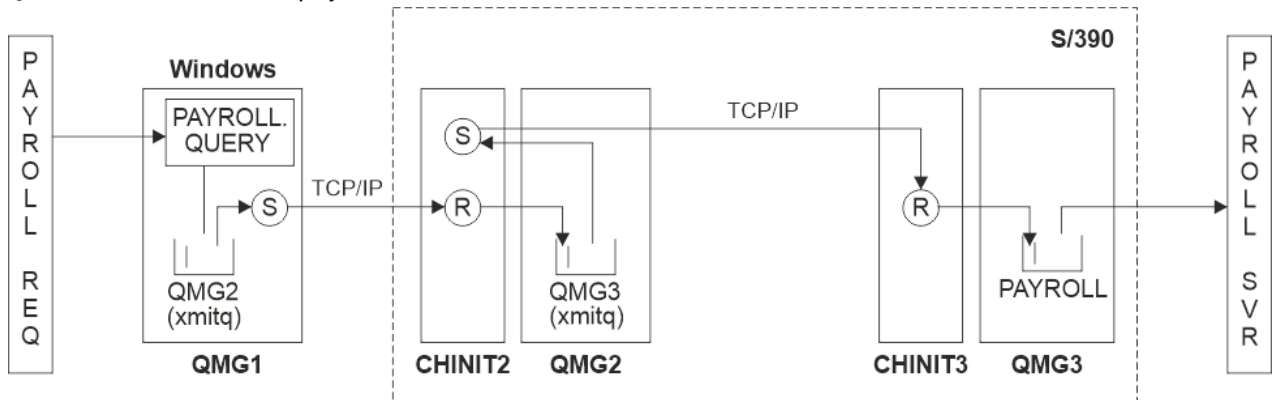


Figure 2. Configuration 1: z/OS using intra-group queuing

The flow of operations is as follows:

1. A query is entered using the payroll request application connected to queue manager QMG1.
2. The payroll request application puts the query on to remote queue PAYROLL.QUERY. As queue PAYROLL.QUERY resolves to transmission queue QMG2, the query is put on to transmission queue QMG2.
3. Sender channel (S) on queue manager QMG1 delivers the query to the partner receiver channel (R) on queue manager QMG2.
4. Receiver channel (R) on queue manager QMG2 puts the query on to queue PAYROLL on queue manager QMG3. As queue PAYROLL on QMG3 resolves to transmission queue QMG3, the query is put on to transmission queue QMG3.
5. Sender channel (S) on queue manager QMG2 delivers the query to the partner receiver channel (R) on queue manager QMG3.
6. Receiver channel (R) on queue manager QMG3 puts the query on to local queue PAYROLL.
7. The payroll server application connected to queue manager QMG3 retrieves the query from local queue PAYROLL, processes it, and generates a suitable reply.

The definitions required for Configuration 1 are as follows (note that the definitions do not take into account triggering, and that only channel definitions for communication using TCP/IP are provided).

Procedure

1. Procedure on QMG1:

- a) Setup the remote queue definition:

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

- b) Setup the transmission queue definition:

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

- c) Setup the sender channel definition using TCP/IP:

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

Note: Replace MVSQMG2(1415) with your queue manager connection name and port.

- d) Setup the receiver channel definition using TCP/IP:

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

- e) Setup the reply-to queue definition:

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

2. Procedure on QMG2:

- a) Setup the transmission queue definition:

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

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

- b) Setup the sender channel definitions using TCP/IP:

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

Note: Replace WINTQMG1(1414) with your queue manager connection name and port.

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

Note: Replace MVSQMG3(1416) with your queue manager connection name and port.

- c) Setup the receiver channel definitions using TCP/IP:

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

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

3. Procedure on QMG3:

- a) Setup the local queue definition:

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

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

- b) Setup the sender channel definition using TCP/IP:

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

Note: Replace MVSQMG2(1415) with your queue manager connection name and port.

c) Setup the receiver channel definition using TCP/IP:

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

4. Use the following procedure to run configuration 1:

- a) Start queue managers QMG1, QMG2, and QMG3.
- b) Start channel initiators for QMG2 and QMG3.
- c) Start the listeners on QMG1 to listen to port 1414, QMG2 to listen on port 1415, and QMG3 to listen on port 1416.
- d) Start sender channels on QMG1, QMG2, and QMG3.
- e) Start the payroll query requesting application connected to QMG1.
- f) Start the payroll server application connected to QMG3.
- g) Submit a payroll query request to QMG3 and wait for the payroll reply.

z/OS Setting up and running configuration 2

Configuration 2 describes how queue sharing groups and intra-group queuing can be used, with no effect on the back-end payroll server application, to transfer messages between queue managers QMG1 and QMG3.

About this task

Configuration 2 shows a distributed queuing system that uses queue sharing groups and intra-group queuing to transfer messages from the payroll request application to the payroll server. This configuration removes the need for channel definitions between queue managers QMG2 and QMG3 because intra-group queuing is used to transfer messages between these two queue managers.

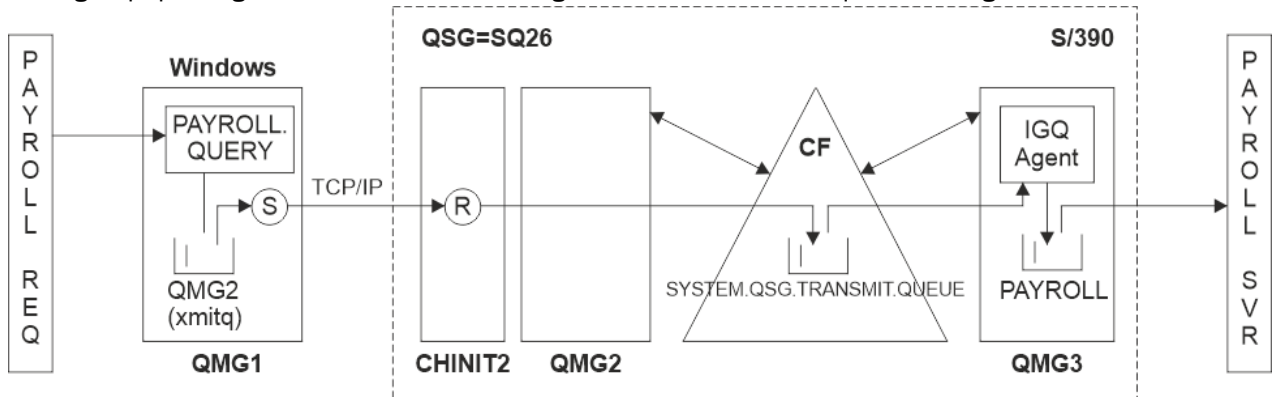


Figure 3. Configuration 2

The flow of operations is as follows:

1. A query is entered using the payroll request application connected to queue manager QMG1.
2. The payroll request application puts the query on to remote queue PAYROLL.QUERY. As queue PAYROLL.QUERY resolves to transmission queue QMG2, the query is put on to transmission queue QMG2.
3. Sender channel (S) on queue manager QMG1 delivers the query to the partner receiver channel (R) on queue manager QMG2.
4. Receiver channel (R) on queue manager QMG2 puts the query on to queue PAYROLL on queue manager QMG3. As queue PAYROLL on QMG3 resolves to shared transmission queue SYSTEM.QSG.TRANSMIT.QUEUE, the query is put on to shared transmission queue SYSTEM.QSG.TRANSMIT.QUEUE.
5. IGQ agent on queue manager QMG3 retrieves the query from shared transmission queue SYSTEM.QSG.TRANSMIT.QUEUE, and puts it on to local queue PAYROLL on queue manager QMG3.

6. The payroll server application connected to queue manager QMG3 retrieves the query from local queue PAYROLL, processes it, and generates a suitable reply.

Notes:

- The payroll query example transfers small messages only. If you need to transfer both persistent and non-persistent messages, you can establish a combination of Configuration 1 and Configuration 2, so that large messages can be transferred using the distributed queuing route, while small messages can be transferred using the potentially faster intra-group queuing route.
- The definitions do not take into account triggering, and that only channel definitions for communication using TCP/IP are provided.
- The example assumes that you have already configured queue managers QMG2 and QMG3 to be members of the same queue sharing group.

Procedure

1. Procedure on QMG1:

- a) Setup the remote queue definition:

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

- b) Setup the transmission queue definition:

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

- c) Setup the sender channel definition for TCP/IP:

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

Note: Replace MVSQMG2(1415) with your queue manager connection name and port.

- d) Setup the receiver channel definition for TCP/IP:

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

- e) Setup the reply-to queue definition:

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

2. Procedure on QMG2:

- a) Setup the transmission queue definition:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)  
  
DEFINE QLOCAL(SYSTEM.QSG.TRANSMIT.QUEUE) QSGDISP(SHARED) +  
DESCR('IGQ Transmission queue') REPLACE PUT(ENABLED) USAGE(XMITQ) +  
GET(ENABLED) INDXTYPE(CORRELID) CFSTRUCT('APPLICATION1') +  
DEFSOPT(SHARED) DEFPSIST(NO)
```

Note: Replace APPLICATION1 with your defined CF structure name. Also, this queue being a shared queue, need only be defined on one of the queue managers in the queue sharing group.

- b) Setup the sender channel definitions for TCP/IP:

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

Note: Replace WINTQMG1(1414) with your queue manager connection name and port.

c) Setup the receiver channel definition for TCP/IP:

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

d) Setup the queue manager definition:

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

3. Procedure on QMG3:

a) Setup the local queue definition:

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

b) Setup the queue manager definition:

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

4. Use the following procedure to run configuration 2:

- a) Start queue managers QMG1, QMG2, and QMG3.
- b) Start the channel initiator for QMG2.
- c) Start the listeners on QMG1 to listen on port 1414, and QMG2 to listen on port 1415.
- d) Start the sender channel on QMG1 and QMG2.
- e) Start the payroll query requesting application connected to QMG1.
- f) Start the payroll server application connected to QMG3.
- g) Submit a payroll query request to QMG3 and wait for the payroll reply.

Setting up and running configuration 3

Configuration 3 describes how queue sharing groups and shared queues can be used, with no effect on the back-end payroll server application, to transfer messages between queue managers QMG1 and QMG3.

About this task

Configuration 3 shows a distributed queuing system that uses queue sharing groups and shared queues to transfer messages between queue manager QMG1 and queue manager QMG3.

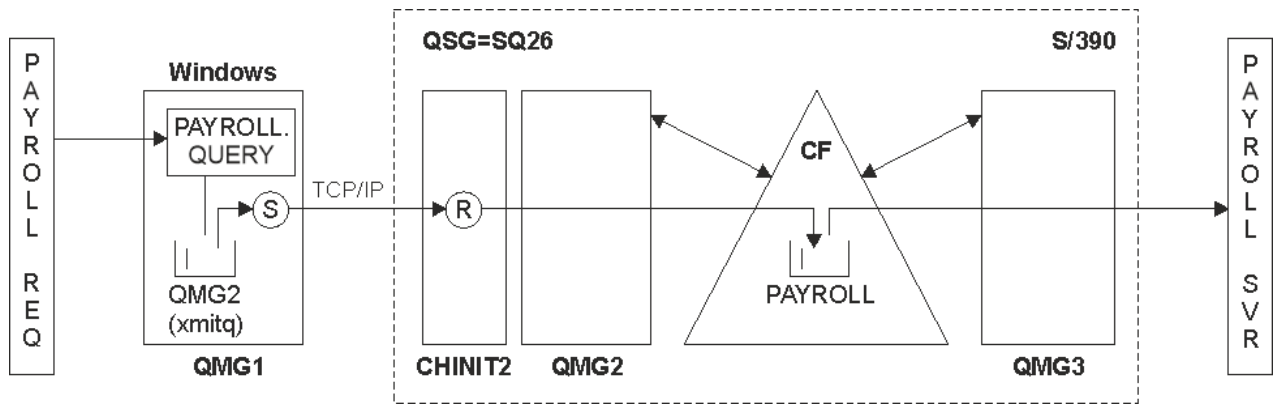


Figure 4. Configuration 3

The flow of operations is:

1. A query is entered using the payroll request application connected to queue manager QMG1.
2. The payroll request application puts the query on to remote queue PAYROLL.QUERY. As queue PAYROLL.QUERY resolves to transmission queue QMG2, the query is put on to transmission queue QMG2.
3. Sender channel (S) on queue manager QMG1 delivers the query to the partner receiver channel (R) on queue manager QMG2.
4. Receiver channel (R) on queue manager QMG2 puts the query on to shared queue PAYROLL.
5. The payroll server application connected to queue manager QMG3 retrieves the query from shared queue PAYROLL, processes it, and generates a suitable reply.

This configuration is certainly the simplest to configure. However, you would need to configure distributed queuing or intra-group queuing to transfer replies (generated by the payroll server application connected to queue manager QMG3) from queue manager QMG3 to queue manager QMG2, and then on to queue manager QMG1.

For the configuration used to transfer replies back to the payroll request application, see [“Example: planning a message channel for z/OS using queue sharing groups”](#) on page 170.

Notes:

- Only channel definitions for communication using TCP/IP are provided.
- The example assumes that you have already configured queue managers QMG2 and QMG3 to be members of the same queue sharing group.
- No definitions are required on QMG3.

Procedure

1. Procedure on QMG1:

- a) Setup the remote queue definition:

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

- b) Setup the transmission queue definition:

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

- c) Setup the sender channel definition:

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


Note: Replace MVSQMG2(1415) with your queue manager connection name and port.

d) Setup the transmission channel definition:

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

e) Setup the reply-to queue definition:

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

2. Procedure on QMG2:

a) Setup the transmission queue definition:

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

b) Setup the sender channel definitions for TCP/IP:

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

Note: Replace WINTQMG1(1414) with your queue manager connection name and port.

c) Setup the receiver channel definitions for TCP/IP:

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

d) Setup the local queue definition:

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

Note: Replace APPLICATION1 with your defined CF structure name. Also this queue being a shared queue, need only be defined on one of the queue managers in the queue sharing group.

3. Use the following procedure to run configuration 3:

- a) Start queue managers QMG1, QMG2, and QMG3.
- b) Start the channel initiator for QMG2.
- c) Start the listeners on QMG1 to listen on port 1414, and QMG2 to listen on port 1415.
- d) Start sender channels on QMG1 and QMG2.
- e) Start the payroll query requesting application connected to QMG1.
- f) Start the payroll server application connected to QMG3.
- g) Submit a payroll query request to QMG3 and wait for the payroll reply.

Linux > IBM i > AIX **Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen**

Nachfolgend ist die Sicherheit beschrieben, die auf die Dateien und Verzeichnisse unter /var/mqm/ angewendet wird; außerdem wird erläutert, weshalb die Dateisystemberechtigungen auf die jeweiligen Werte gesetzt sind. Um sicherzustellen, dass IBM MQ ordnungsgemäß funktioniert, sollten Sie die von IBM MQ festgelegten Dateisystemberechtigungen nicht ändern.

crtmqdir-Befehl

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

IBM MQ-Dateisystemsicherheit unter AIX, Linux und IBM i

In den Dateien unter dem IBM MQ-Datenverzeichnis (`/var/mqm`) wird Folgendes gespeichert:

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

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

Die Erteilung des Zugriffs erfolgt in den folgenden drei Kategorien:

Nur Gruppe 'mqm'

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

Die Dateiberechtigungen für diese Dateien und Verzeichnisse lauten wie folgt:

```
-rwxrwx---   mqm:mqm      (UNIX and Linux)
-rwxrwx---   QMQMADM:QMOM (IBM i)
```

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

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

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

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

Die Dateiberechtigungen für diese Dateien und Verzeichnisse lauten wie folgt:

```
-rwxrwxr-x   mqm:mqm      (UNIX and Linux)
-rwxrwxr-x   QMQMADM:QMOM (IBM i)
```

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

```
/var/mqm/mqs.ini
/var/mqm/exits/
/var/mqm/qmgrs/
/var/mqm/qmgrs/QMGR/
```

```
/var/mqm/qmgrs/QMGR/@app/  
/var/mqm/qmgrs/QMGR/@ipcc/
```



Achtung: Sie sollten Ausführungsberechtigungen nur für ausführbare Dateien und Scripts festlegen. Beispiel: Unter Linux werden bei der Ausführung des Befehls **crtmqm** die folgenden Dateiberechtigungen festgelegt:

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

IBM MQ 8.0:

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

Lese- und Schreibzugriff für alle Benutzer

Dateien mit Lese- und Schreibzugriff für alle Benutzer

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

Diese speziellen Dateien bedeuten keine Sicherheitslücke. Obwohl die Berechtigungen als 777 angezeigt werden, handelt es sich nicht um *Regulär*-Dateien und Sie können nicht direkt in sie schreiben.

Es handelt sich hierbei um die folgenden speziellen Dateien:

Symbolische Links

Symbolische Links werden durch das Zeichen `l` am Anfang ihrer Berechtigungen identifiziert. Die Berechtigungen für die symbolische Verbindung haben keinen Einfluss darauf, wer auf die Zieldatei zugreifen kann. Der Grund hierfür ist, dass der Zugriff auf den Befehl durch die Berechtigungen auf dem Ziel der symbolischen Verbindung gesteuert wird.

Auf den meisten AIX and Linux-Systemen ist es nicht möglich, die Berechtigungen für symbolische Verbindungen zu ändern, sodass sie immer als `lrwxrwxrwx` angezeigt werden.

Socketdateien

Socketdateien sind spezielle Dateien, die vom Betriebssystem als Folge eines Prozesses erstellt werden, der ein UNIX-Domänensocket erstellt. Diese Dateien können durchs am Anfang der Dateiberechtigungen identifiziert werden, d. h. `srwxrwxrwx`.

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

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

Die folgenden Verzeichnisse enthalten Socketdateien mit Lese-/Schreibberechtigungen für alle Benutzer (`srwxrwxrwx`).

IBM MQ 8.0:

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

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

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

Verzeichnisse mit Lese- und Schreibzugriff für alle Benutzer

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

Mit Ausnahme der Fehlerprotokolldateien, in die von jedem Mitglied der Gruppe mqm geschrieben werden kann, werden alle in diesen Verzeichnissen erstellten Dateien mit eingeschränkten Berechtigungen erstellt, die nur den Schreibzugriff des Dateierstellers zulassen. Dadurch kann der Systemadministrator die Benutzer-ID aller Daten verfolgen, die in Dateien dieser Verzeichnisse geschrieben werden.

/var/mqm/errors/

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

Das `SetGroupId`-Bit zeigt an, dass alle Dateien, die in diesem Verzeichnis erstellt werden, das Gruppeneigentum mqm haben.

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

Anmerkung:  Diese Funktion ist unter IBM i nicht verfügbar.

AMQERRO*.LOG

Diese Fehlerprotokolldateien können nur direkt von Mitgliedern der `group` geschrieben werden, aber jeder Benutzer kann die in diese Dateien geschriebenen Nachrichten lesen (Berechtigung: `-rw-rw-r--`).

AMQnnnnn*.FDC

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

/var/mqm/trace/

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

Die Berechtigungen für dieses Verzeichnis sind `drwxrwsrwt`, d. h., alle Benutzer auf dem System können Dateien in diesem Verzeichnis erstellen.

Das `SetGroupId`-Bit zeigt an, dass alle Dateien, die in diesem Verzeichnis erstellt werden, das Gruppeneigentumsrecht von 'mqm' haben.

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

Anmerkung:  Diese Funktion ist unter IBM i nicht verfügbar.

AMQnnnnn*.TRC

Diese Dateien enthalten die Tracedaten, die von jedem Prozess geschrieben werden, für den ein Trace erstellt wird, und die mit den Berechtigungen `-rw-r-----` erstellt werden.

Die Berechtigungen für dieses Verzeichnis sind `drwxrwsrwt` und die Berechtigungen der in diesem Verzeichnis erstellten Socketdateien sind `srwx-----`.

IBM MQ 8.0:

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

Dieses Verzeichnis wird von Anwendungen verwendet, die über *isolierte* Bindungen eine Verbindung zum IBM MQ -Warteschlangenmanager herstellen. Während der Verbindungsverarbeitung wird von der verbindenden Anwendung in diesem Verzeichnis eine Socketdatei erstellt. Nachdem die Verbindung zu dem Warteschlangenmanager hergestellt wurde, wird die Socketdatei entfernt.

Die Berechtigungen für dieses Verzeichnis sind `drwxrwsrwt` und die Berechtigungen der in diesem Verzeichnis erstellten Socketdateien sind `srxw-----`.

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

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

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

AIX Für Prozesse, die über *gemeinsam genutzte* Bindungen eine Verbindung zu IBM MQ herstellen, können UNIX-Domänensockets für die Synchronisation zwischen der Anwendung und dem Warteschlangenmanager verwendet werden. Wenn UNIX-Domänensockets verwendet werden, wird die zugehörige Socketdatei in diesen Verzeichnissen erstellt.

Die Berechtigungen für diese Verzeichnisse sind `drwxrwsrwt` und die Berechtigungen der in diesen Verzeichnissen erstellten Socketdateien sind `srxwrxwrxw`.

Das `SetGroupId`-Bit für diese Verzeichnisse stellt sicher, dass alle in diesen Verzeichnissen erstellten Dateien das Gruppeneigentumsrecht `mqm` haben.

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

HOME

Bei Verwendung einer nicht registrierten oder nicht installierten Version von IBM MQ wie dem wieder-verteilbaren Client wird das Verzeichnis `/${HOME}/.mqm` erstellt.

Das Verzeichnis wird erstellt, damit IBM MQ zuverlässig auf seine Socketdateien zugreifen kann, indem ein Pfad verwendet wird, der innerhalb der `sun_path`-Länge liegt. Wenn IBM MQ keine Daten in das Verzeichnis HOME schreiben kann, erhalten Sie eine Fehlermeldung.

Verwendung von System V-IPC-Ressourcen durch IBM MQ

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

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

- Prüfen Sie die Eigentümerschaft.

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

- Verwenden Sie ab IBM MQ 8.0 das Dienstprogramm `amqspdbg`.

Das Dienstprogramm `amqspdbg`, das mit IBM MQ geliefert wird, kann zum Anzeigen des gemeinsam genutzten Speichers und der Semaphor-IDs für einen bestimmten Warteschlangenmanager verwendet werden.

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

```
# amqspdbg -z -I
```

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

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

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

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

Linux AIX IBM MQ-Dateiberechtigungen in /opt/mqm mit setuid für mqm

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

Problemursache

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

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

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

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

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

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

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

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

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

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

Lösung des Problems

Eine der Bedenken in Bezug auf Setuid-Programme auf UNIX-Systemen war, dass die Systemsicherheit durch die Manipulation von Umgebungsvariablen wie LD* (LD_LIBRARY_PATH, LIBPATH auf AIX und so weiter) gefährdet werden könnte. Dies ist kein Problem mehr, weil verschiedene UNIX-Betriebssysteme diese LD*-Umgebungsvariablen jetzt beim Laden von setuid-Programmen ignorieren.

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

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

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

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

Dank der Möglichkeit, unterschiedliche Authentifizierungsebenen für Benutzer einzurichten, und der Tatsache, dass **setuid**- und **setgid**-Programme LD*-Variablen ignorieren, wird die Sicherheit des Systems in keinerlei Weise durch die Binär- und Bibliotheksdateien von IBM MQ beeinträchtigt.

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

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

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

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

Weitere Informationen finden Sie unter [„Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen“](#) auf Seite 65.

Zugehörige Konzepte

[Dateisystem](#)

In den folgenden Informationen wird die Sicherheit beschrieben, die auf die Dateien und Verzeichnisse unter Windows angewendet wird. Um sicherzustellen, dass IBM MQ ordnungsgemäß funktioniert, sollten Sie die von IBM MQ festgelegten Dateisystemberechtigungen nicht ändern.

Datenverzeichnis

Anmerkung: Die Berechtigungen, die für das Stammelement dieses Verzeichnisses festgelegt sind, werden in der gesamten Verzeichnisstruktur nach unten übernommen.

Für die Verzeichnisse unter dem Datenverzeichnis (DATADIR) sind die folgenden Berechtigungen festgelegt, abgesehen von den Ausnahmen, die im folgenden Text ausführlich beschrieben werden.

Administratoren

Uneingeschränkter Zugriff

Gruppe 'mqm'

Uneingeschränkter Zugriff

SYSTEM

Uneingeschränkter Zugriff

Alle

Lese- und Ausführungszugriff

Die Ausnahmen sind:

DATADIR\errors

Alle uneingeschränkten Zugriff

DATADIR\trace

Alle uneingeschränkten Zugriff

DATADIR\log

Administratoren

Uneingeschränkter Zugriff

Gruppe 'mqm'

Uneingeschränkter Zugriff

SYSTEM

Uneingeschränkter Zugriff

Alle

Lesen

DATADIR\log\\active

Administratoren

Uneingeschränkter Zugriff

Gruppe 'mqm'

Uneingeschränkter Zugriff

SYSTEM

Uneingeschränkter Zugriff

Kein Zugriff erteilt für 'Alle'.

Die Fehlerprotokolldateien AMQERR01.LOG usw. übernehmen ihre Sicherheitseinstellungen nicht aus ihrem Verzeichnis, sondern werden stattdessen auf Everyone: Full Control (Alle: uneingeschränkter Zugriff) gesetzt.

Frühere Releases des Produkts

In Releases des Produkts vor IBM MQ 8.0 befanden sich das Standardprogramm und die Standarddatenverzeichnisse an einer gemeinsamen Position.

In jeder Installation, die ursprünglich vor IBM MQ 8.0 installiert wurde. Die Daten- und Programmverzeichnisse, die an den Standardpositionen installiert und anschließend von dort aktualisiert wurden, bleiben an derselben Position (in C:\Program Files\IBM\WebSphere MQ).

Falls sich die Daten- und Programmverzeichnisse an einer gemeinsamen Position befinden, gelten die obigen Informationen nur für die Verzeichnisse, die zum Datenverzeichnis gehören, und nicht für die Verzeichnisse, die Teil des Programmverzeichnisses sind.

Benennungseinschränkungen für Warteschlangen

Es gibt Einschränkungen bei der Länge von Warteschlangennamen. Einige Warteschlangennamen sind für Warteschlangen reserviert, die vom Warteschlangenmanager definiert werden.

Einschränkungen bei der Länge von Namen

Die Namen von Warteschlangen können bis zu 48 Zeichen lang sein.

Reservierte Warteschlangennamen

Namen, die mit "SYSTEM." beginnen, sind für Warteschlangen reserviert, die vom Warteschlangenmanager definiert werden. Mit dem Befehl **ALTER** oder **DEFINE REPLACE** können Sie diese Warteschlangendefinitionen an Ihre Installation anpassen. Folgende Namen sind für IBM MQ definiert:

<i>Tabelle 41. Namen und Beschreibungen für reservierte Warteschlangen</i>	
Name der Warteschlange	Beschreibung
SYSTEM.ADMIN.ACTIVITY.QUEUE	Warteschlange für Aktivitätsberichte
SYSTEM.ADMIN.CHANNEL.EVENT	Warteschlange für Kanalereignisse
SYSTEM.ADMIN.COMMAND.EVENT	Warteschlange für Befehlsereignisse
SYSTEM.ADMIN.COMMAND.QUEUE	Warteschlange, an die PCF-Befehlsnachrichten gesendet werden
SYSTEM.ADMIN.CONFIG.EVENT	Warteschlange für Konfigurationsereignisse
SYSTEM.ADMIN.PERFM.EVENT	Warteschlange für Leistungsereignisse
SYSTEM.ADMIN.PUBSUB.EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM.ADMIN.QMGR.EVENT	Warteschlange für Warteschlangenmanagerereignisse
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Warteschlange für Traceroute-Antwortnachrichten
SYSTEM.AUTH.DATA.QUEUE	Warteschlange mit den Zugriffskontrolllisten für den Warteschlangenmanager (nicht für z/OS)
SYSTEM.CHANNEL.INITQ	Initialisierungswarteschlange für Kanäle
SYSTEM.CHANNEL.SYNCQ	Warteschlange mit den Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	Warteschlange zum Auslösen (nicht unter z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange zur Kommunikation von Repository-Änderungen zwischen Warteschlangenmanagern
SYSTEM.CLUSTER.HISTORY.QUEUE	Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange mit Informationen zum Repository

Tabelle 41. Namen und Beschreibungen für reservierte Warteschlangen (Forts.)

Name der Warteschlange	Beschreibung
SYSTEM.CLUSTER.TRANSMIT.MODEL.DEL.QUEUE	Die Warteschlange zum Erstellen der einzelnen Übertragungswarteschlangen für den jeweiligen Clustersenderkanal
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle von der Clusterunterstützung verwalteten Ziele
SYSTEM.COMMAND.INPUT	Warteschlange, an die Befehlsnachrichten unter z/OS gesendet werden
SYSTEM.COMMAND.REPLY.MODEL	Definition der Modellwarteschlange für Antworten auf Befehle (unter z/OS)
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten (nicht unter z/OS)
SYSTEM.DEFAULT.ALIAS.QUEUE	Standarddefinition der Aliaswarteschlange
SYSTEM.DEFAULT.INITIATION.QUEUE	Warteschlange für die Auslösung eines angegebenen Prozesses (nicht unter z/OS)
SYSTEM.DEFAULT.LOCAL.QUEUE	Standarddefinition der lokalen Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standarddefinition der Modellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standarddefinition der fernen Warteschlange
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	Lokale Warteschlange mit einer persistenten Kopie der permanenten Subskriptionen des Warteschlangenmanagers
SYSTEM.HIERARCHY.STATE	Warteschlange mit Informationen zum Status der Beziehungen zwischen Warteschlangenmanagern in einer Publish/Subscribe-Hierarchie
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen
SYSTEM.INTERNAL.REPLY.QUEUE	IBM MQ-interne Antwortwarteschlange (nicht unter z/OS)
SYSTEM.INTER.QMGR.CONTROL	Warteschlange in einer Publish/Subscribe-Hierarchie für den Empfang von Anforderungen von einem fernen Warteschlangenmanager zur Erstellung einer Proxy-Subskription
SYSTEM.INTER.QMGR.PUBS	Warteschlange in einer Publish/Subscribe-Hierarchie für den Empfang von Veröffentlichungen von einem fernen Warteschlangenmanager
SYSTEM.INTER.QMGR.FANREQ	Warteschlange in einer Publish/Subscribe-Hierarchie für die Verarbeitung von Anforderungen zur Erstellung einer Proxy-Subskription auf einem fernen Warteschlangenmanager
SYSTEM.MQEXPLORER.REPLY.MODEL	Definition der Modellwarteschlange für Antworten auf IBM MQ Explorer
SYSTEM.MQSC.REPLY.QUEUE	Modellwarteschlangendefinition für Antworten auf WebSphere MQ-Scriptbefehle (nicht unter z/OS)
SYSTEM.QSG.CHANNEL.SYNCQ	Gemeinsam genutzte lokale Warteschlange zum Speichern von Nachrichten mit den Synchronisationsdaten für gemeinsame Kanäle (nur unter z/OS)

<i>Tabelle 41. Namen und Beschreibungen für reservierte Warteschlangen (Forts.)</i>	
Name der Warteschlange	Beschreibung
SYSTEM.QSG.TRANSMIT.QUEUE	Gemeinsam genutzte lokale Warteschlange, die vom Agenten zur gruppeninternen Warteschlangensteuerung für die Übertragung von Nachrichten zwischen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange verwendet wird (nur unter z/OS)
SYSTEM.RETAINED.PUB.QUEUE	Lokale Warteschlange mit Kopien aller ständigen Veröffentlichungen des Warteschlangenmanagers
SYSTEM.SELECTION.EVALUATION.QUEUE	IBM MQ-interne Warteschlange für Auswahlbewertungen (nicht unter z/OS)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM MQ-interne Warteschlange für Auswahlprüfungen (nicht unter z/OS)

Benennungseinschränkungen für andere Objekte

Es gibt Einschränkungen bei der Länge von Objektnamen. Einige Objektnamen sind für Objekte reserviert, die vom Warteschlangenmanager definiert werden.

Einschränkungen bei der Länge von Namen

Die Namen von Prozessen, Namenslisten, Clustern, Themen, Services und Authentifizierungsdatenobjekten können eine Länge von maximal 48 Zeichen haben.

Die Namen von Kanälen können eine Länge von maximal 20 Zeichen haben.

Die Namen von Speicherklassen können eine Länge von maximal 8 Zeichen haben.



Die Namen von CF-Strukturen können eine Länge von maximal 12 Zeichen haben.

Reservierte Objektnamen

Namen, die mit SYSTEM beginnen, sind für vom Warteschlangenmanager definierte Objekte reserviert. Mit den Befehlen **ALTER** oder **DEFINE REPLACE** können Sie diese Objektdefinitionen an Ihre Installation anpassen. Folgende Namen sind für IBM MQ definiert:

<i>Tabelle 42. Namen und Beschreibungen für reservierte Objekte</i>	
Objektname	Beschreibung
SYSTEM.ADMIN.SVRCONN	Serververbindungskanal für die ferne Verwaltung eines Warteschlangenmanagers
SYSTEM.AUTO.RECEIVER	Standardempfängerkanal für automatische Definition (nur unter AIX, Linux, and Windows)
SYSTEM.AUTO.SVRCONN	Standardserververbindungskanal für automatische Definition (nur Multiplatforms)
SYSTEM.BASE.TOPIC	Basisartikel zur ASPARENT-Auflösung. Wenn ein besonderes administratives Artikelobjekt keine übergeordneten administrativen Artikelobjekte hat, werden alle ASPARENT-Attribute von diesem Objekt übernommen.
SYSTEM.DEF.CLNTCONN	Standarddefinition für Clientverbindungskanal
SYSTEM.DEF.CLUSRCVR	Standarddefinition für Clusterempfängerkanal
SYSTEM.DEF.CLUSSDR	Standarddefinition für Clustersenderkanal

Tabelle 42. Namen und Beschreibungen für reservierte Objekte (Forts.)

Objektname	Beschreibung
SYSTEM.DEF.RECEIVER	Standarddefinition für Empfängerkanal
SYSTEM.DEF.REQUESTER	Standarddefinition für Requesterkanal
SYSTEM.DEF.SENDER	Standarddefinition für Senderkanal
SYSTEM.DEF.SERVER	Standarddefinition für Serverkanal
SYSTEM.DEF.SVRCONN	Standarddefinition für Serververbindungskanal
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Definition eines Standardauthentifizierungsdatenobjekts zur Definition von Authentifizierungsdatenobjekten vom Typ CRLLDAP
SYSTEM.DEFAULT.AUTHINFO.OCSP	Definition eines Standardauthentifizierungsdatenobjekts zur Definition von Authentifizierungsdatenobjekten vom Typ OCSP
SYSTEM.DEFAULT.LISTENER.LU62	Standardmäßiges SNA-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.NETBIOS	Standardmäßiges NetBIOS-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.SPX	Standardmäßiges SPX-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.TCP	Standardmäßiges TCP/IP-Empfangsprogramm (nur Multiplatforms)
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.DEFAULT.PROCESS	Standarddefinition für Prozesse
SYSTEM.DEFAULT.SERVICE	Standardservice (nur Multiplatforms)
SYSTEM.DEFAULT.TOPIC	Standarddefinition für Prozesse
SYSTEM.QPUBSUB.QUEUE.NAMELIST	Eine Liste von Warteschlangen für das zu überwachende Queued Publish/Subscribe Interface
  SYSTEMST	Standardmäßige Speicherklassendefinition (nur unter z/OS)

Auflösung des Warteschlangennamens

In größeren Netzen bietet die Verwendung von Warteschlangenmanagern gegenüber anderen Kommunikationsformen einige Vorteile. Diese Vorteile ergeben sich aus der Namensauflösungsfunktion im verteilten Warteschlangenmanagement, die sicherstellt, dass Warteschlangennamensauflösungen von Warteschlangenmanagern auf der Sende- und der Empfangsseite eines Kanals durchgeführt werden.

Die Hauptvorteile dieses Ansatzes sind:

- Anwendungen müssen keine Routing-Entscheidungen treffen.
- Anwendungen müssen die Netzstruktur nicht kennen.
- Netzverknüpfungen werden von Systemadministratoren erstellt.
- Die Netzstruktur wird von Netzplanern gesteuert.
- Zur Partitionierung des Datenverkehrs können mehrere Kanäle verwendet werden.

In der folgenden Abbildung wird ein Beispiel für die Auflösung von Warteschlangennamen gezeigt. Zu sehen sind zwei Maschinen in einem Netz; auf einer Maschine ist eine PUT-Anwendung aktiv, auf der anderen Maschine eine GET-Anwendung. Die Anwendungen kommunizieren miteinander über den IBM MQ-Kanal, der von den MCAs gesteuert wird.

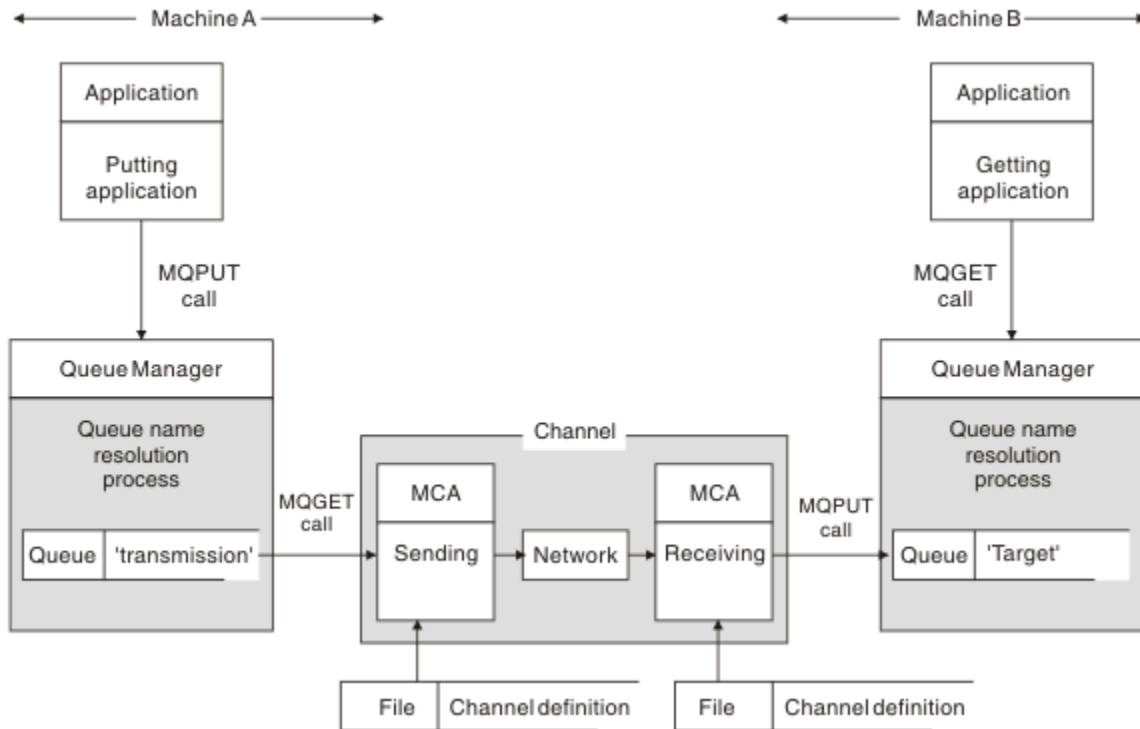


Abbildung 5. Namensauflösung

Wie in Abbildung 5 auf Seite 77 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.

Zugehörige Tasks

Einreihen von Nachrichten in ferne Warteschlangen

Was ist die Auflösung von Warteschlangennamen?

Die Warteschlangennamensauflösung ist notwendig für verteiltes Warteschlangenmanagement. 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, gibt es eine Zwischenstufe zwischen dem Namen, der von der Anwendung für die Zielwarteschlange verwendet wird, und der Benennung des Kanals, 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 Warteschlangenname, 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.

Wie Zielobjektattribute für Aliasnamen, ferne Warteschlangen und Clusterwarteschlangen aufgelöst werden

Wenn die Namensauflösung für einen Anwendungs-API-Aufruf durchgeführt wird, werden Attribute, die sich auf die Verwendung des Objekts auswirken, aus einer Kombination aus ursprünglich benanntem Objekt, dem "Pfad" (siehe „Auflösung des Warteschlangennamens“ auf Seite 76), und aufgelöstem Zielobjekt aufgelöst. In einem Warteschlangenmanagercluster ist das fragliche "benannte Objekt" die Definition des Objekts im Cluster (Warteschlange oder Thema). Dies ist eine Untergruppe der Objektattribute, die von Warteschlangenmanagern gemeinsam genutzt werden und über sichtbar sind. Beispiel: **DISPLAY QCLUSTER**.

Wo ein Attribut für das benannte Objekt, das von der Anwendung geöffnet wurde, definiert werden kann, hat es Vorrang. Beispielsweise können alle DEF****-Attribute (Standardpersistenz, Priorität und asynchrone PUT-Antwort) für Definitionen von Alias- und fernen Warteschlangen konfiguriert werden. Sie werden wirksam, wenn die Alias- oder ferne Warteschlange von einer Anwendung geöffnet wird, statt einer aufgelösten Zielwarteschlange oder Übertragungswarteschlange.

Attribute, die die Anwendungsinteraktion mit einem Zielobjekt einschränken oder begrenzen sollen, können in der Regel nicht für das benannte Objekt (Definition der fernen Warteschlange oder Aliasname) definiert werden. Zum Beispiel können **MAXMSGL** und **MAXDEPTH** nicht für eine Definition einer fernen Warteschlange oder einen Aliasnamen festgelegt werden und sie werden nicht zwischen den Mitgliedern eines Warteschlangenmanagerclusters übergeben. Diese Attribute werden daher von der aufgelösten Warteschlange übernommen (z. B. der lokalen Warteschlange, der entsprechenden Übertragungswarteschlange oder SYSTEM.CLUSTER.TRANSMIT.QUEUE). Bei der Ankunft auf einem fernen Warteschlangenmanager kann für die Zustellung an die Zielwarteschlange eine zweite Einschränkung wirksam werden, die dazu führen kann, dass eine Nachricht in eine Warteschlange für nicht zustellbare Nachrichten gestellt wird oder der Kanal zwangsweise gestoppt wird.

Es ist zu beachten, dass die **PUT**- und **GET**-Aktivierung einen Sonderfall von Attributauflösung darstellen. Für beide Attribute gilt, dass eine Instanz von **DISABLED** im Warteschlangenpfad bewirkt, dass es ein allgemeines aufgelöstes Attribut **DISABLED** gibt.

System-und Standardobjekte

Hier sind die über den Befehl **crtmqm** erstellten System- und Standardobjekte aufgeführt.

Wenn Sie mit dem Steuerbefehl **crtmqm** einen Warteschlangenmanager erstellen, werden die System- und Standardobjekte automatisch erstellt.

- Systemobjekte sind die für den Betrieb eines Warteschlangenmanagers oder Kanals erforderlichen IBM MQ-Objekte.
- In den Standardobjekten sind alle Attribute eines Objekts definiert. Wenn Sie ein Objekt erstellen (z. B. eine lokale Warteschlange), werden alle Attribute, die Sie nicht explizit angeben, aus dem Standardobjekt übernommen.

In den folgenden Tabellen sind die System- und Standardobjekte aufgeführt, die von **crtmqm** erstellt werden.

Anmerkung: Es gibt zwei weitere Standardobjekte, die nicht in den Tabellen enthalten sind: das Warteschlangenmanagerobjekt und der Objektkatalog. Diese Objekte werden protokolliert und sind wiederherstellbar.

- [System- und Standardobjekte: Warteschlangen](#)
- [System- und Standardobjekte: Themen](#)
- [System- und Standardobjekte: Serverkanäle](#)
- [System- und Standardobjekte: Clientkanäle](#)
- [System- und Standardobjekte: Authentifizierungsdaten](#)
- [System- und Standardobjekte: Kommunikationsdaten](#)
- [System- und Standardobjekte: Empfangsprogramme](#)
- [System- und Standardobjekte: Namenslisten](#)
- [System- und Standardobjekte: Prozesse](#)
- [System- und Standardobjekte: Services](#)

<i>Tabelle 43. System- und Standardobjekte: Warteschlangen</i>	
Objektname	Beschreibung
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Warteschlange für Abrechnungsnachrichtendaten, die bei der Trennung der Verbindung zwischen einer Anwendung und dem Warteschlangenmanager generiert werden.
SYSTEM.ADMIN.ACTIVITY.QUEUE	Die Warteschlange für zurückgegebene Aktivitätsbereichsnachrichten.
SYSTEM.ADMIN.CHANNEL.EVENT	Ereigniswarteschlange für Kanäle
SYSTEM.ADMIN.COMMAND.EVENT	Ereigniswarteschlange für Befehlsereignisse.
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.LOGGER.EVENT	Ereigniswarteschlange für Protokollierungsnachrichten (Journalempfänger).
SYSTEM.ADMIN.PERFM.EVENT	Ereigniswarteschlange für Leistungsereignisse
SYSTEM.ADMIN.PUBSUB.EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM.ADMIN.QMGR.EVENT	Ereigniswarteschlange für Warteschlangenmanagereignisse.
SYSTEM.ADMIN.STATISTICS.QUEUE	Die Warteschlange für MQI-, Warteschlangen- und Kanalstatistiknachrichtendaten.
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.AMQP.COMMAND.QUEUE	IBM MQ-Verwaltungsbefehlswarteschlange für AMQP
SYSTEM.AUTH.DATA.QUEUE	Warteschlange mit den Zugriffskontrolllisten für den Warteschlangenmanager Wird vom Objektberechtigungsmanager verwendet

Tabelle 43. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.BROKER.ADMIN.STREAM	Verwaltungsdatenstrom für Schnittstelle für eingereihtes Publish/Subscribe
SYSTEM.BROKER.CONTROL.QUEUE	Steuerwarteschlange für Publish/Subscribe-Schnittstelle
SYSTEM.BROKER.DEFAULT.STREAM	Standarddatenstrom für Schnittstelle für eingereihtes Publish/Subscribe
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	Warteschlange für Kommunikation zwischen Brokern
SYSTEM.CHANNEL.INITQ	Kanalinitialisierungwarteschlange
SYSTEM.CHANNEL.SYNCQ	Warteschlange für die Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	CICS-Standardinitialisierungwarteschlange.
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange für die Übertragung von Nachrichten an den Repository-Warteschlangenmanager
SYSTEM.CLUSTER.HISTORY.QUEUE	Die Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange für die Speicherung aller Repositoryinformationen
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Die Warteschlange zum Erstellen der einzelnen Übertragungwarteschlangen für den jeweiligen Cluster-senderkanal
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungwarteschlange für alle Nachrichten an alle Cluster
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten.
SYSTEM.DEFAULT.ALIAS.QUEUE	Standardaliaswarteschlange
SYSTEM.DEFAULT.INITIATION.QUEUE	Standardinitialisierungwarteschlange
SYSTEM.DEFAULT.LOCAL.QUEUE	Standardmäßige lokale Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standardmodellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standardmäßige ferne Warteschlange
SYSTEM.DOTNET.XARECOVERY.QUEUE	IBM MQ .NET-XA-Wiederherstellungwarteschlange
SYSTEM.DURABLE.MODEL.QUEUE	Die Warteschlange, die als Modell für verwaltete, permanente Subskriptionen verwendet wird
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	Warteschlange mit einer persistenten Kopie der permanenten Subskriptionen des Warteschlangenmanagers
SYSTEM.HIERARCHY.STATE	Hierarchiebeziehungsstatus für verteiltes Publish/Subscribe von IBM MQ.

Tabelle 43. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.INTER.QMGR.CONTROL	Steuerwarteschlange für verteiltes Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.FANREQ	Eingabewarteschlange für Ausgabefächerungsprozesse der internen Proxy-Subskription des verteilten Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.PUBS	Veröffentlichungen des verteilten Publish/Subscribe von IBM MQ.
SYSTEM.INTERNAL.REPLY.QUEUE	
SYSTEM.INTERNAL.REQUEST.QUEUE	
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen
SYSTEM.MQEXPLORER.REPLY.MODEL	Die IBM MQ Explorer-Warteschlange für zu beantwortende Nachrichten. Hierbei handelt es sich um eine Modellwarteschlange, mit der eine temporäre dynamische Warteschlange für Antworten an den IBM MQ Explorer erstellt wird.
SYSTEM.MQSC.REPLY.QUEUE	Empfangswarteschlange für Antworten für MQSC-Befehle. Dies ist eine Modellwarteschlange, aus der eine temporäre dynamische Warteschlange für Antworten an ferne MQSC-Befehle erstellt wird.
SYSTEM.NDURABLE.MODEL.QUEUE	Warteschlange, die als Modell für verwaltete, nicht permanente Subskriptionen verwendet wird
SYSTEM.PENDING.DATA.QUEUE	Unterstützt verzögerte Nachrichten in JMS.
SYSTEM.PROTECTION.ERROR.QUEUE	IBM MQ Fehlerwarteschlange für Nachrichtenschutz.
SYSTEM.PROTECTION.POLICY.QUEUE	IBM MQ Warteschlange für Nachrichtenschutzrichtlinien.
SYSTEM.REST.REPLY.QUEUE	
SYSTEM.RETAINED.PUB.QUEUE	Warteschlange für eine Kopie jeder ständigen Veröffentlichung im Warteschlangenmanager
SYSTEM.SELECTION.EVALUATION.QUEUE	
SYSTEM.SELECTION.VALIDATION.QUEUE	

Tabelle 44. System- und Standardobjekte: Themen

Objektname	Beschreibung
SYSTEM.ADMIN.TOPIC	Verwaltungsthema.
<u>SYSTEM.BASE.TOPIC</u>	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.BROKER.ADMIN.STREAM	Verwaltungsdatenstrom, der von der Schnittstelle für eingereichtes Publish/Subscribe verwendet wird

Tabelle 44. System- und Standardobjekte: Themen (Forts.)

Objektname	Beschreibung
SYSTEM.BROKER.DEFAULT.STREAM	Standarddatenstrom, der von der Schnittstelle für eingereihtes Publish/Subscribe verwendet wird
SYSTEM.BROKER.DEFAULT.SUBPOINT	Der Standardunterpunkt, der von der Schnittstelle für eingereihtes Publish/Subscribe verwendet wird
SYSTEM.DEFAULT.TOPIC	Standardthemendefinition.

Tabelle 45. System- und Standardobjekte: Serverkanäle

Objektname	Beschreibung
SYSTEM.AUTO.RECEIVER	Dynamischer Empfängerkanal
SYSTEM.AUTO.SVRCONN	Dynamischer Serververbindungskanal
SYSTEM.DEF.AMQP	AMQP-Standardkanal. Beachten Sie, dass das Objekt definiert ist, aber der AMQP-Service nicht unterstützt wird.
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.DEFAULT.AUTHINFO.IDPWLDAP	
SYSTEM.DEFAULT.AUTHINFO.IDPWOS	

Tabelle 46. System- und Standardobjekte: Clientkanäle

Objektname	Beschreibung
SYSTEM.DEF.CLNTCONN	Standardclientverbindungskanal.

Tabelle 47. System- und Standardobjekte: Authentifizierungsdaten

Objektname	Beschreibung
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Standardauthentifizierungsdatenobjekt zur Definition von Authentifizierungsdatenobjekten vom Typ CRLLDAP.

Tabelle 47. System- und Standardobjekte: Authentifizierungsdaten (Forts.)

Objektname	Beschreibung
SYSTEM.DEFAULT.AUTHINFO.OCSP	Standardauthentifizierungsdatenobjekt zur Definition von Authentifizierungsdatenobjekten vom Typ OCSP.

Tabelle 48. System- und Standardobjekte: Kommunikationsdaten

Objektname	Beschreibung
SYSTEM.DEFAULT.COMMINFO.MULTICAST	Standardkommunikationsinformationsobjekt für Multicasting.

Tabelle 49. System- und Standardobjekte: Empfangsprogramme




Objektname	Beschreibung
SYSTEM.DEFAULT.LISTENER.TCP	Standardempfangsprogramm für TCP-Transport
 SYSTEM.DEFAULT.LISTENER.LU62	LU62-Standardempfangsprogramm.
 SYSTEM.DEFAULT.LISTENER.NETBIOS	NETBIOS-Standardempfangsprogramm.
 SYSTEM.DEFAULT.LISTENER.SPX	SPX-Standardempfangsprogramm.

Tabelle 50. System- und Standardobjekte: Namenslisten

Objektname	Beschreibung
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.QPUBSUB.QUEUE.NAMELIST	Liste mit Warteschlangennamen, die von der Schnittstelle für eingereichtes Publish/Subscribe überwacht werden
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	Liste mit Themenobjekten, die von der Schnittstelle für eingereichtes Publish/Subscribe zum Abgleichen von Themenobjekten mit Subskriptionspunkten verwendet werden

Tabelle 51. System- und Standardobjekte: Prozesse

Objektname	Beschreibung
SYSTEM.DEFAULT.PROCESS	Standardprozessdefinition.

Tabelle 52. System- und Standardobjekte: Services

Objektname	Beschreibung
SYSTEM.AMQP.SERVICE	MQ LightAPI-Service. Beachten Sie, dass das Objekt definiert ist, aber der Service nicht unterstützt wird.
SYSTEM.DEFAULT.SERVICE	Standardservice.

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.

Tabelle 53. Standardwerte von SYSTEM.BASE.TOPIC	
Parameter	Wert
TOPICSTR	"
V9.4.0 CAPEXPY	NOLIMIT
CLROUTE	DIRECT
Cluster	Der Standardwert ist eine leere Zeichenfolge.
COMMINFO	SYSTEM.DEFAULT.COMMINFO.MULTICAST
DEFPRESP	SYNC
DEFPRTY	0
DEFPSIST	NO
DESCR	'Base topic for resolving attributes'
DURSUB	YES
MCAST	DISABLED
MDURMDL	SYSTEM.DURABLE.MODEL.QUEUE
MNDURMDL	SYSTEM.NDURABLE.MODEL.QUEUE
NPMSGDLV	ALLAVAIL
PMSGDLV	ALLDUR
PROXYSUB	FIRSTUSE
PUB	ENABLED
PUBSCOPE	ALL
z/OS z/OS QSGDISP (nur z/OS-Plattform)	QMGR
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

Falls dieses Objekt nicht vorhanden ist, werden seine Standardwerte von IBM MQ dennoch für ASPARENT-Attribute verwendet, die nicht über übergeordnete Themen in der Themenstruktur aufgelöst werden.

Durch Einstellen des Attributs PUB oder SUB von SYSTEM.BASE.TOPIC auf DISABLED wird verhindert, dass Anwendungen Themen in der Themenstruktur publizieren oder subscribieren, mit zwei Ausnahmen:

1. Themenobjekte in der Themenstruktur, deren Attribut PUB oder SUB explizit auf ENABLE eingestellt ist. Anwendungen können diese Themen und die ihnen untergeordneten Elemente publizieren und 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.




Weitere Informationen finden Sie unter [Besondere Handhabung des Parameters PUB](#).



Zeilengruppeninformationen für Konfigurationsdateien

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:

- Zeilengruppen und Attribute der Datei [mqs.ini](#) unterstützen Sie bei der Konfiguration der folgenden Elemente:
 - Zeilengruppe *AllQueueManagers*
 - Zeilengruppe *DefaultQueueManager*
 - Zeilengruppe *ExitProperties*
 - Zeilengruppe *LogDefaults*
 - Zeilengruppe *Security* in der Datei `qm.ini`
- Zeilengruppen und Attribute der Datei [qm.ini](#) unterstützen Sie bei der Konfiguration der folgenden Komponenten:
 -  Zeilengruppe *AccessMode* (nur unter Windows)
 - Zeilengruppe *Service* - für installierbare Services
 - Zeilengruppe *Log*
 -   Zeilengruppe *RestrictedMode* (nur auf Systemen mit AIX 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*und enthält Links dazu, wie sie für verschiedene Dienste auf AIX, Linux, and Windows Plattformen verwendet werden.
- [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*
- Die IBM MQ MQI client -Konfigurationsdatei `mqclient.ini` unterstützt Sie bei der Konfiguration der folgenden Komponenten:

- Zeilengruppe *CHANNELS*
- Zeilengruppe *ClientExitPath*
-  Zeilengruppen *LU62*, *NETBIOS* und *SPX* (nur unter Windows)
- Zeilengruppe *MessageBuffer*
- Zeilengruppe *SSL*
- Zeilengruppe *TCP*
-  Zeilengruppe *Trace* (nur für IBM MQ .NET und XMS .NET verwendet)
- „Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen“ auf Seite 88 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.

- Mithilfe der automatischen Konfiguration können Sie Folgendes konfigurieren:
 - Zeilengruppe 'AutoConfig'
 - Zeilengruppe 'AutoCluster'
 - Zeilengruppe 'Variables'

Konfigurationsdateien

Unter

- Datei [mqs.ini](#)
- Datei [qm.ini](#)
- Datei [mqclient.ini](#)

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

mqs.ini-Datei

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

Eine Datei *mqs.ini* kann die folgenden Zeilengruppen enthalten:

- [AllQueue](#)
- [DefaultQueueManager](#)
- [ExitProperties](#)

- [LogDefaults](#)


Außerdem gibt es für jeden Warteschlangenmanager eine [QueueManager](#)-Zeilegruppe.

Datei 'qm.ini'

Ein Beispiel für eine Datei `qm.ini` finden Sie im Abschnitt [Beispiel einer Konfigurationsdatei für den Warteschlangenmanager für IBM MQ for AIX or Linux-Systeme](#).

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

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

 Verwenden Sie zum Konfigurieren von [InstallableServices](#) die Zeilegruppen `Service` und `ServiceComponent`.



- `Connection` für [DefaultBindType](#)



Achtung: Falls Sie eine `Connection`-Zeilegruppe benötigen, müssen Sie diese erstellen.

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

Zusätzlich können Sie Folgendes ändern:

-  `AccessMode` (nur unter Windows)
-  `RestrictedMode` (nur auf Systemen mit AIX and Linux)

Hierfür verwenden Sie den Befehl `crtmqm`.

Datei 'mqclient.ini'

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

- [KANÄLE](#)
- [ClientExitPfad](#)
- [LU62, NETBIOS und SPX](#)
- [MessageBuffer](#)
- [SSL](#)
- [TCP](#)

Darüber hinaus wird möglicherweise eine [Zeilegruppe PreConnect](#) zur Konfiguration eines PreConnect-Exits benötigt.

Zeilegruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen

Beschreibung der Zeilegruppen der Warteschlangenmanager-Konfigurationsdatei `qm.ini` für die verteilte Steuerung von Warteschlangen

Dieser Abschnitt enthält die Zeilegruppen in der Warteschlangenmanager-Konfigurationsdatei für die verteilte Steuerung von Warteschlangen. Sie gelten für die Warteschlangenmanager-Konfigurationsdatei für IBM MQ for Multiplatforms. Die Datei heißt auf allen Plattformen `qm.ini`.

Zeilengruppen für die verteilte Steuerung von Warteschlangen sind:

- Kanäle
- TCP
- LU62
- NETBIOS
- EXITPATH

In [Abbildung 6 auf Seite 89](#) 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=type       ; Whether the binding for applications is to be
                          ; "fastpath" or "standard".
                          ; The default is "standard".
  PIPELINELENGTH=n       ; The maximum number of concurrent threads a channel will use.
                          ; The default is 1. Any value greater than 1 is treated as 2.
  ADOPTNEWMCA=chlname    ; 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.
  ADOPTNEWMCACHECK=     ; Specifies the type checking required.
  typecheck              ; The default is "NAME","ADDRESS", and "QM".
  CHLAUTHEARLYADOPT=Y/N ; The order in which connection authentication and channel authentica
tion rules are          ; processed. If not present in the qm.ini file the default is "N".
From MQ9.0.4 all        ; queue managers are created with a default of "Y"
  PASSWORDPROTECTION=   ; From MQ8.0, set protected passwords in the MQCSP structure, rather
than using TLS.         ;
  options               ; The options are "compatible", "always", "optional" and "warn"
                          ; The default is "compatible".
  IGNORESEQNUMBERMISMATCH ; How the queue manager handles a sequence number mismatch during chan
nel startup.           ;
  =Y/N                 ; The options are "Y" and "N" with the default being "N".
  CHLAUTHIGNOREUSERCASE ; Enables a queue manager to make username matching within CHLAUTH
rules case-insensitive. ;
  =Y/N                 ; The options are "Y" and "N" with the default being "N".
  CHLAUTHISSUEWARN=Y   ; If you want message AMQ9787 to be generated when you set the WARN=YES
attribute              ; on the SET CHLAUTH command.
TCP:                   ; TCP entries
  PORT=n               ; Port number, the default is 1414
  KEEPALIVE=Yes        ; Switch TCP/IP KeepAlive on
LU62:
  LIBRARY2=DLLName2    ; Used if code is in two libraries
  EXITPATH:1           ; Location of user exits
  EXITPATHS=           ; String of directory paths.
```

Abbildung 6. Zeilengruppen der Datei qm.ini für die verteilte Steuerung von Warteschlangen

Anmerkungen:

1. EXITPATH gilt nur auf den folgenden Plattformen:

-  AIX
-  Windows

Zugehörige Tasks
[konfigurieren](#)

Kanalattribute

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

Sie wählen die Attribute eines Kanals so aus, dass sie unter bestimmten Bedingungen für jeden Kanal optimal sind. Wenn der Kanal dann ausgeführt wird, wurden die tatsächlichen Werte jedoch möglicherweise im Rahmen von Startvereinbarungen geändert. Weitere Informationen finden Sie im Abschnitt [Kanäle vorbereiten](#).

Für viele Attribute gibt es Standardwerte, die für die meisten Kanäle verwendet werden können. Für Bedingungen, unter denen die Standardwerte nicht optimal sind, finden Sie in diesem Abschnitt eine Anleitung zur Auswahl der richtigen Werte.

Anmerkung: In IBM MQ for IBM i können die meisten Attribute als *SYSDFTCHL angegeben werden, d. h., es wird der Wert des Standardkanals Ihres Systems übernommen.

Die Kanaltypen für IBM MQ-Kanalattribute werden in der folgenden Tabelle in der Reihenfolge der MQSC-Befehlsparameter aufgelistet.

Anmerkung: Wenn bei Clusterkanälen (die Tabellenspalten CLUSSDR und CLUSRCVR) ein Attribut für beide Kanäle gesetzt werden kann, definieren Sie das Attribut für beide Kanäle und achten Sie darauf, dass für beide Kanäle dieselbe Attributeinstellung verwendet wird. Bei unterschiedlichen Einstellungen werden in der Regel die Einstellungen verwendet, die Sie für den Kanal CLUSRCVR angegeben haben. Dies wird im Abschnitt [Clusterkanäle](#) erläutert.

Tabelle 54. Kanalattribute für die Kanaltypen


Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AM QP
Verbindungsaffinität	AFFINITY					Ja				
Datum ändern	ALTDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Zeit ändern	ALTTIME	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
 AMQP-Keepalive	AMQPKA									Ja
Intervall der Überwachungssignale für Stapel	BATCHHB	Ja	Ja					Ja	Ja	
Stapelintervall	BATCHINT	Ja	Ja					Ja	Ja	
Batch limit (Stapeldatengrenzwert)	BATCHLIM	Ja	Ja					Ja	Ja	
Stapelgröße	BATCHSZ	Ja	Ja	Ja	Ja			Ja	Ja	
Zertifikatsbezeichnung	CERTLABL	Ja	Ja	Ja	Ja	Ja	Ja	Yes (Ja) „1“ auf Seite 94	Ja	Ja
Channel Name	CHANNEL	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja

Tabelle 54. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS - SDR	CLUS-RCVR	AM QP
<u>Channel Type</u>	CHLTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>Clientkanalge- wichtung</u>	CLNTWGH T					Ja				
<u>Clusternamenslis- te</u>	CLUSNL							Ja	Ja	
<u>Cluster</u>	CLUSTER							Ja	Ja	
<u>Clusterauslas- tungspriorität</u>	CLWLPRT Y							Ja	Ja	
<u>Clusterauslas- tungsrangordnung</u>	CLWLRAN K							Ja	Ja	
<u>Clusterauslas- tungsgewichtung</u>	CLWLWGH T							Ja	Ja	
<u>Header-Kompri- mierung</u>	COMPHDR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Datenkomprimie- rung</u>	COMPMSG	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Verbindungsname</u>	CONNNAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Nachricht umwan- deln</u>	CONVERT	Ja	Ja					Ja	Ja	
<u>Standardmäßige Verbindungswie- derholung</u>	DEFRE- CON					Ja				
<u>Beschreibung</u>	DESCR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>Unterbrechungs- intervall</u>	DISCINT	Ja	Ja				z/OS Ja Multi Nein	Ja	Ja	
<u>Intervall der Über- wachungssignale</u>	HBINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Keepalive-Inter- vall</u>	KAINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Lokale Adresse</u>	LOC- LADDR	Ja	Ja		Ja	Ja		Ja	Ja	Ja
<u>Zähler für Wieder- holungsversuche nach langem In- tervall</u>	LONGRTY	Ja	Ja					Ja	Ja	
<u>Langes Wiederho- lungsintervall</u>	LONGTMR	Ja	Ja					Ja	Ja	
<u>Maximale Instan- zen</u>	MAXINST						Ja			Ja

Tabelle 54. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AM QP
Maximale Instanzen pro Client	MAXINSTC						Ja			
Maximale Nachrichtenlänge	MAXMSGLE	Ja	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	Ja
Gibt den LU 6.2-Modusnamen an.	MODENAME	Ja	Ja		Ja	Ja		Ja	Ja	
Monitoring	MONCHL	Ja	Ja	Ja	Ja		Ja	Ja	Ja	
Benutzerdaten des Exits für Nachrichtenwiederholungen	MRDATA			Ja	Ja				Ja	
Name des Exits für Nachrichtenwiederholung	MREXIT			Ja	Ja				Ja	
Zähler für Nachrichtenwiederholungen	MRRTY			Ja	Ja				Ja	
Intervall für Nachrichtenwiederholungen	MRTMR			Ja	Ja				Ja	
Benutzerdaten des Nachrichtenexits	MSGDATA	Ja	Ja	Ja	Ja			Ja	Ja	
Name des Nachrichtenexits	MSGEXIT	Ja	Ja	Ja	Ja			Ja	Ja	
Netzverbindungspriorität	NETPRTY								Ja	
Gibt die Geschwindigkeit nicht permanenter Nachrichten an.	NPMSPEED	Ja	Ja	Ja	Ja			Ja	Ja	
Kennwort	PASSWORD	Ja	Ja		Ja	Ja		Ja		
Port Number	PORT									Ja

Tabelle 54. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Be- fehlspara- meter	SDR	SVR	RCV R	RQST R	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR	AM QP
<u>PROPCTL</u> -Kanal- optionen für MQGMO	PROPCTL	Ja	Ja					Ja	Ja	
<u>PUT-Berechtigung</u>	PUTAUT			Ja	Ja		z/OS Ja Multi Nein		Ja	
<u>Warteschlangen- managername</u>	QMNAME					Ja				
z/OS Dis- position	QSGDISP	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Empfangs- exits</u>	RCVDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzername für Empfangsexit</u>	RCVEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Sicherheits- exits</u>	SCYDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Name des Sicher- heitsexits</u>	SCYEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Sendeexits</u>	SENDDA- TA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Name des Sende- exits</u>	SENDEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Folgenummernse- rie</u>	SEQWRAP	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Gemeinsam ge- nutzte Verbindun- gen</u>	SHA- RECNV					Ja	Ja			
<u>Zähler für Wieder- holungsversuche nach kurzem In- tervall</u>	SHORT- RTY	Ja	Ja					Ja	Ja	
<u>Kurzes Wiederho- lungsintervall</u>	SHORTTM R	Ja	Ja					Ja	Ja	
z/OS Si- cherheitsrichtlini- enschutz	SPLPROT	Ja	Ja	Ja	Ja					
<u>SSL-Clientauthen- tifizierung</u>	SSLCAUTH		Ja	Ja	Ja		Ja		Ja	Ja

Tabelle 54. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AM QP
<u>SSL-Verschlüsselungsspezifikation</u>	SSLCIPH	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>SSL-Peer</u>	SSLPEER	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>Kanalstatistik</u>	STATCHL	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Gibt das LU 6.2-Transaktionsprogramm an.</u>	TPNAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Themen-Root</u>	TPROOT									Ja
<u>Transport Type</u>	TRPTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Client-ID verwenden</u>	USECLTID									Ja
<u>Warteschlange für nicht zustellbare Nachrichten verwenden</u>	USEDLQ	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Benutzer-ID</u>	USERID	Ja	Ja		Ja	Ja		Ja		
<u>Name der Übertragungswarteschlange</u>	XMITQ	Ja	Ja							

Anmerkungen:

1. Dieses Attribut kann durch keine Verwaltungsschnittstelle abgefragt oder für CLUSSDR-Kanäle festgelegt werden. Sie erhalten eine MQRCCF_WRONG_CHANNEL_TYPE-Nachricht. Das Attribut ist allerdings in CLUSSDR-Kanalobjekten vorhanden (einschließlich MQCD-Strukturen) und kann bei Bedarf durch einen CHAD-Exit programmgesteuert festgelegt werden.

IBM MQ implementiert für bestimmte Plattformen unter Umständen nicht alle in diesem Abschnitt aufgeführten Attribute. Auf Ausnahmen und Plattformunterschiede wird ggf. in den einzelnen Attributbeschreibungen hingewiesen.

Der Name jedes Attributs wird in Klammern angezeigt.

Die Attribute sind in alphabetischer Reihenfolge in Gruppen angeordnet.

Zugehörige Verweise

MQSC-Befehle

ALTER CHANNEL

CHANNEL DEFINE CHANNEL

Kanalattribute für MQSC-Schlüsselwörter (A-B)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter, beginnend mit dem Buchstaben A oder B.

AFFINITY (Verbindungsaffinität)

Dieses Attribut gibt an, ob Clientanwendungen, bei denen mehrfach Verbindungen mit dem gleichen Warteschlangenmanagernamen hergestellt werden, denselben Clientkanal verwenden.

Verwenden Sie dieses Attribut (MQIACH_CONNECTION_AFFINITY), wenn mehrere gültige Kanaldefinitionen verfügbar sind.

Folgende Werte sind möglich:

PREFERRED

Die erste Verbindung eines Prozesses, der eine Definitionstabelle für Clientkanäle (CCDT) liest, erstellt basierend auf der Clientkanalgewichtung eine Liste gültiger Definitionen, in der die Definitionen mit der Gewichtung 0 jeweils in alphabetischer Reihenfolge zuerst aufgeführt sind. Bei jeder Verbindung des Prozesses wird versucht, die Verbindung über die erste Definition der Liste herzustellen. Wenn eine Verbindung nicht erfolgreich ist, wird die nächste Definition verwendet. Nicht erfolgreiche Definitionen mit Clientkanalgewichtungswerten ungleich null werden an das Ende der Liste verschoben. Definitionen mit einer Clientkanalgewichtung von 0 verbleiben am Anfang der Liste und werden für jede Verbindung zuerst ausgewählt.

Jeder Clientprozess mit demselben Hostnamen erstellt immer dieselbe Liste.

Bei Clientanwendungen, die in C, C++ oder im .NET-Programmierframework geschrieben wurden (einschließlich vollständig verwalteter .NET-Clientanwendungen) und für Anwendungen, die IBM MQ classes for Java und IBM MQ classes for JMS verwenden, wird die Liste aktualisiert, wenn die Definitionstabelle für Clientkanäle (CCDT) seit der Erstellung der Liste geändert wurde.

Dieser Wert ist der Standardwert und hat den Wert 1.

KEINE

Die erste Verbindung eines Prozesses, die eine CCDT liest, erstellt eine Liste gültiger Definitionen. Alle Verbindungen eines Prozesses wählen eine gültige Definition basierend auf der Clientkanalgewichtung aus, wobei Definitionen mit der Gewichtung 0 in alphabetischer Reihenfolge zuerst ausgewählt werden.

Bei Clientanwendungen, die in C, C++ oder im .NET-Programmierframework geschrieben wurden (einschließlich vollständig verwalteter .NET-Clientanwendungen) und für Anwendungen, die IBM MQ classes for Java und IBM MQ classes for JMS verwenden, wird die Liste aktualisiert, wenn die Definitionstabelle für Clientkanäle (CCDT) seit der Erstellung der Liste geändert wurde.

Dieses Attribut ist nur für den Kanaltyp Clientverbindungskanal gültig.

ALTDATE (Änderungsdatum)

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

ALTTIME (Änderungszeit)

Dieses Attribut gibt den Zeitpunkt der letzten Änderung der Definition im Format hh.mm.ss an und ist für alle Kanaltypen gültig.

AMQPKA (AMQP-Keepalive)



Mit dem Attribut **AMQPKA** können Sie eine Keepalive-Zeit für die AMQP-Clientverbindung angeben. Wenn der AMQP-Client innerhalb des Keepalive-Intervalls keine Frames gesendet hat, wird die Verbindung geschlossen.

Das Attribut **AMQPKA** bestimmt den Wert des Attributs für das Inaktivitätszeitlimit, das von IBM MQ an einen AMQP-Client gesendet wird. Das Attribut gibt einen Zeitraum in Millisekunden an.

Wenn **AMQPKA** auf einen Wert > 0 gesetzt ist, fließt IBM MQ die Hälfte dieses Werts als Attribut für das Inaktivitätszeitlimit. Der Wert 10000 bewirkt beispielsweise, dass der Warteschlangenmanager einen Wert von 5000 für das Inaktivitätszeitlimit sendet. Der Client muss sicherstellen, dass mindestens alle 10.000 Millisekunden Daten an IBM MQ gesendet werden. Wenn IBM MQ in dieser Zeit keine Daten empfängt, geht IBM MQ davon aus, dass der Client seine Verbindung verloren hat, und erzwingt das Schließen der Verbindung mit einer `amqp:resource-limit-exceeded`-Fehlerbedingung.

Der Wert AUTO oder 0 gibt an, dass IBM MQ kein Attribut für das Inaktivitätszeitlimit an den AMQP-Client übergibt.

Ein AMQP-Client kann jedoch einen eigenen Wert für das Inaktivitätszeitlimit übergeben. In diesem Fall übergibt IBM MQ Daten (oder einen leeren AMQP-Frame) mit mindestens dieser Häufigkeit, um den Client darüber zu informieren, dass er verfügbar ist.

BATCHHB (Intervall des Stapelüberwachungssignals)

Mit diesem Attribut kann ein sendender Kanal überprüfen, ob der empfangende Kanal noch aktiv ist, bevor ein Nachrichtenstapel festgeschrieben wird.

Damit ermöglicht das Stapelintervall der Überwachungssignale, dass der Stapel zurückgesetzt wird und nicht unbestätigt bleibt, wenn der empfangende Kanal inaktiv ist. Durch das Zurücksetzen des Stapels bleiben die Nachrichten für die Verarbeitung verfügbar und können zum Beispiel an einen anderen Kanal umgeleitet werden.

Wenn der sendende Kanal vom empfangenden Kanal innerhalb des Überwachungsintervalls für den Stapelbetrieb ein Meldung erhalten hat, wird davon ausgegangen, dass der empfangende Kanal immer noch aktiv ist. Andernfalls wird zur Überprüfung ein Überwachungssignal an den empfangenden Kanal gesendet. Der sendende Kanal wartet die durch das Kanalattribut 'Intervall der Überwachungssignale' (HBINT) angegebene Anzahl an Sekunden auf eine Antwort vom empfangenden Kanal.

Der Wert wird in Millisekunden angegeben und muss im Bereich 0 bis 999999 liegen. Beim Wert 0 werden keine Überwachungssignale für Stapel verwendet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

BATCHINT (Stapelintervall)

Dieses Attribut gibt an, wie lange (in Millisekunden) der Kanal einen Stapel geöffnet hält, auch wenn sich keine Nachrichten in der Übertragungswarteschlange befinden.

Sie können einen beliebigen Wert von 0 bis 999.999.999 angeben. Der Standardwert ist null.

Wenn Sie kein Stapelintervall angeben, wird der Stapel bei Eintreten einer der folgenden Bedingungen geschlossen:

- Die in BATCHSZ angegebene Anzahl von Nachrichten wurde gesendet.
- Die in BATCHLIM angegebene Anzahl an Byte wurde gesendet.
- Die Übertragungswarteschlange ist leer.

Bei Kanälen mit geringer Auslastung und häufig leerer Übertragungswarteschlange kann die effektive Stapelgröße deutlich unter BATCHSZ liegen.

Mit dem Attribut BATCHINT können Sie die Effizienz der Kanäle steigern, indem Sie die Anzahl kurzer Stapel verringern. Beachten Sie jedoch, dass sich die Reaktionszeit verlängert, weil Stapel länger dauern und Nachrichten länger nicht festgeschrieben werden.

Wenn Sie BATCHINT angeben, werden Stapel nur bei einer der folgenden Bedingungen geschlossen:

- Die in BATCHSZ angegebene Anzahl von Nachrichten wurde gesendet.
- Die in BATCHLIM angegebene Anzahl an Byte wurde gesendet.
- Es gibt keine weiteren Nachrichten in der Übertragungswarteschlange und das Zeitintervall von BATCHINT ist abgelaufen, während auf Nachrichten gewartet wurde (seit Abruf der ersten Nachricht des Stapels).

Anmerkung: BATCHINT gibt die gesamte Zeit an, in der auf Nachrichten gewartet wird. Darin ist nicht die Zeit für den Abruf von Nachrichten enthalten, die bereits in der Übertragungswarteschlange verfügbar sind, und nicht die Zeit, die für das Übertragen von Nachrichten benötigt wird.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

BATCHLIM (Stapelgrenzwert)

Dieses Attribut gibt (in Kilobyte) das Datenvolumen an, das vor dem nächsten Synchronisationspunkt maximal über einen Kanal gesendet werden kann.

Ein Synchronisationspunkt wird erreicht, nachdem die Nachricht, mit der dieser Grenzwert erreicht wurde, vollständig über den Kanal übertragen wurde.

Der Wert muss zwischen 0 und 999999 liegen. Der Standardwert ist 5000.

Der Wert null für dieses Attribut bedeutet, dass es für Stapel auf diesem Kanal keinen Datengrenzwert gibt.

Der Stapel wird beendet, wenn eine der folgenden Bedingungen zutrifft:

- BATCHSZ-Nachrichten wurden gesendet.
- BATCHLIM-Bytes wurden gesendet.
- Die Übertragungswarteschlange ist leer und BATCHINT wurde überschritten.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

BATCHSZ (Stapelgröße)

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.
- 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

Kanalattribute für MQSC-Schlüsselwörter (C)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter, beginnend mit dem Buchstaben C.

CERTLABL (Zertifikatsbezeichnung)

Dieses Attribut gibt die Zertifikatsbezeichnung der Kanaldefinition an.

Die Bezeichnung ermittelt, welches persönliche Zertifikat im Schlüsselrepository an den fernen Peer gesendet wird. Das Zertifikat wird wie unter [Digitale Zertifikatsbeschreibungen](#) beschrieben definiert.

Eingehende Kanäle (einschließlich RCVR-, RQSTR-, CLUSRCVR-, nicht qualifizierte SERVER- und SVRCONN-Kanäle) senden das konfigurierte Zertifikat nur, wenn die IBM MQ-Version des fernen Peers die Konfiguration der Zertifikatsbezeichnung vollständig unterstützt und der Kanal ein TLS-CipherSpec verwendet.

Ist dies nicht der Fall, ermittelt das Attribut **CERTLABL** des Warteschlangenmanagers das gesendete Zertifikat. Diese Einschränkung besteht, da der Auswahlmechanismus für die Zertifikatsbezeichnung für eingehende Kanäle von einer TLS-Protokollerweiterung abhängt, die nicht in allen Fällen unterstützt wird. Insbesondere Java -Clients und JMS -Clients unterstützen die erforderliche Protokollerweiterung nicht und empfangen nur das vom Warteschlangenmanagerattribut **CERTLABL** konfigurierte Zertifikat, unabhängig von der kanalspezifischen Kennsatzeinstellung.

Ein nicht qualifizierter Serverkanal ist ein Kanal, für den das Feld CONNAME nicht festgelegt wurde.

Dieses Attribut kann durch keine Verwaltungsschnittstelle abgefragt oder für CLUSSDR-Kanäle festgelegt werden. Sie erhalten eine MQRCCF_WRONG_CHANNEL_TYPE-Nachricht. Das Attribut ist allerdings in CLUSSDR-Kanalobjekten vorhanden (einschließlich MQCD-Strukturen) und kann bei Bedarf durch einen CHAD-Exit programmgesteuert festgelegt werden.

Weitere Informationen zu den möglichen Inhalten von Zertifikatsbezeichnungen finden Sie im Abschnitt [Digitale Zertifikatsbezeichnungen - Anforderungen](#).

Dieses Attribut ist für alle Kanaltypen gültig.

Anmerkung: Für SSL/TLS muss in der QMGR-Definition ein CERTLABL festgelegt sein. Optional können Sie das CERTLABL auch in der CHANNEL-Definition festlegen.

Das CERTLABL des Warteschlangenmanagers wird überprüft. Es muss ein gültiges persönliches Zertifikat sein, selbst wenn in der CHANNEL-Definition ein CERTLABL festgelegt ist.

CHANNEL (Kanalname)

Dieses Attribut gibt den Namen der Kanaldefinition an.

Der Name kann bis zu 20 Zeichen lang sein. Da jedoch beide Seiten eines Nachrichtenkanals denselben Namen haben müssen und bei anderen Implementierungen Größenbeschränkungen gelten können, muss die tatsächliche Zeichenzahl unter Umständen darunter liegen.

Kanalnamen dürfen auf zwei Warteschlangenmanagern in einem Netz miteinander verbundener Warteschlangenmanager nie doppelt vorkommen.

Der Name muss aus Zeichen der folgenden Liste bestehen:

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

CHLTYPE (Kanaltyp)

Dieses Attribut gibt den Typ des definierten Kanals an.

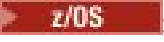
Mögliche Kanaltypen:

Nachrichtentypen:

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

MQI-Kanaltypen:

- Client-Verbindung (nur AIX, Linux, and Windows)

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

- Serververbindung
- AMQP

Beide Kanalseiten müssen denselben Namen und kompatible Typen haben:

- Sender mit Empfänger
- Requester mit Server
- Requester mit Sender (für Callback)
- Server mit Empfänger (Server wird als Sender verwendet)
- Clientverbindung mit Serververbindung
- Clustersender mit Clusterempfänger
- AMQP mit AMQP

CLNTWGHT (Clientkanalgewichtung)

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.

Anmerkung: Wenn eine JSON-CCDT verwendet wird, können mehrere Kanäle denselben Namen haben. Wenn mehrere Kanäle mit demselben Namen vorhanden sind und den Definitionstyp CLNTWGHT (0) aufweisen, werden die Kanäle in der Reihenfolge ausgewählt, in der sie in der JSON-CCDT definiert sind.

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.

CLUSNL (Clusternamensliste)

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

CLUSTER (Cluster)

Dieses Attribut ist der Name des Clusters, dem der Kanal zugeordnet ist.

Die maximale Länge beträgt 48 Zeichen gemäß den Regeln für die Benennung von IBM MQ-Objekten.

Maximal einer der resultierenden Werte von CLUSTER und CLUSNL kann andere Zeichen als nur Leerzeichen enthalten. Wenn einer der Werte andere Zeichen als nur Leerzeichen enthält, darf der andere Wert nur aus Leerzeichen bestehen.

Dieses Attribut ist nur für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

CLWLPRTY (Clusterauslastungspriorität)

Das Kanalattribut CLWLPRTY legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Kanalattribut CLWLPRTY legen Sie die Reihenfolge der Priorität der verfügbaren Clusterziele fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Nachrichten gehen an den Warteschlangenmanager mit dem Kanal mit der höchsten Priorität. Ist dieser Kanal nicht verfügbar, gehen die Nachrichten an den Warteschlangenmanager mit der nächsthöchsten Priorität. Warteschlangenmanager mit niedrigerer Priorität fungieren als Reserve.

Vor der Priorisierung der Kanäle prüft IBM MQ den Kanalstatus. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihr Status mehrmals, wobei einige Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherheit) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.
- Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherheitsziel gesendet werden, verwenden Sie CLWLPRTY nicht. Ziehen Sie die Verwendung separater Warteschlangen in Betracht, oder CLWLRANK mit einem manuellen Umschalten von der primären auf die Sicherheit.

CLWLRANK (Rangordnung der Clusterauslastung)

Das Kanalattribut **CLWLRANK** gibt die Ebene der Kanäle für die Verteilung der Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Verwenden Sie das Kanalattribut **CLWLRANK**, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Sie steuern die Auswahl des Zielorts, indem Sie den Rang der Kanäle festlegen, die einen Warteschlangenmanager am Schnittpunkt der Cluster mit den Gateway-Warteschlangenmanagern verbinden.

Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen. Ein Beispiel: Nachrichten kommen an einem

Gateway-Warteschlangenmanager an, der sie an einen der beiden Warteschlangenmanager weiterleiten kann, die Kanäle mit dem Rang 1 und 2 verwenden. Die Nachrichten werden automatisch an den Warteschlangenmanager gesendet, der durch einen Kanal mit dem höchsten Rang verbunden ist; in diesem Fall ist dies der Warteschlangenmanagerkanal mit dem Rang 2.

IBM MQ ruft den Rang von Kanälen noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Kanäle zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Würden Sie hingegen auch das Prioritätsattribut **CLWLPRTY** verwenden, würde IBM 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.

CLWLWGHT (Clusterauslastungsgewichtung)

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

Verwenden Sie CLWLWGHT, um mehr Nachrichten an Server mit einer größeren Verarbeitungskapazität zu senden. Je stärker ein Kanal gewichtet ist, desto mehr Nachrichten werden über diesen Kanal versendet.

Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Standardeinstellung des Parameters CLWLWGHT ist 50. Wird diese Einstellung für einen Kanal geändert, so wird der Lastausgleich abhängig von der Häufigkeit, die die einzelnen Kanäle für eine an eine Clusterwarteschlange gesendete Nachricht ausgewählt wurden. Weitere Informationen finden Sie unter [„Algorithmus für das Clusterauslastungsmanagement“](#) auf Seite 151.

COMPHDR (Headerkomprimierung)

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.

COMPMSG (Datenkomprimierung)

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 „[COMPHDR \(Headerkomprimierung\)](#)“ auf Seite 102.

Folgende Werte sind möglich:

KEINE

Es werden keine Nachrichtendaten komprimiert. Dies ist der Standardwert.

RLE

Nachrichtendaten werden mittels Lauflängencodierung komprimiert.

ZLIBFAST

Die Komprimierung der Nachrichtendaten erfolgt unter Verwendung der ZLIB-Komprimierungstechnik. Dabei wird eine kurze Komprimierungszeit bevorzugt.

ZLIBFAST kann optional durch die zEnterprise-Datenkomprimierungsfunktion entlastet werden. Weitere Informationen finden Sie unter [zEDC Express-Funktion](#).

ZLIBHIGH

Die Komprimierung der Nachrichtendaten erfolgt unter Verwendung der ZLIB-Komprimierungstechnik. Dabei wird eine hohe Komprimierungsstufe bevorzugt.

V 9.4.0 LZ4FAST

Die Komprimierung von Nachrichtendaten wird mit dem Komprimierungsverfahren LZ4 ausgeführt. Dabei wird eine kurze Komprimierungszeit bevorzugt.

V 9.4.0 LZ4HIGH

Die Nachrichtendatenkomprimierung erfolgt mit LZ4 -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.

AIX Ab IBM MQ 9.3.0 können ZLIBFAST- und ZLIBHIGH-Verfahren unter IBM MQ for AIX die hardwarebeschleunigte Bibliothek zlibNX verwenden, sofern diese installiert ist. Die Bibliothek zlibNX, eine erweiterte Version der Komprimierungsbibliothek zlib, unterstützt hardwarebeschleunigte Datenkomprimierung und -dekomprimierung mittels Co-Prozessoren, auch als Nest-Akzeleratoren (NX) bezeichnet, die auf prozessorbasierten IBM POWER9-Servern verfügbar sind. Die Bibliothek zlibNX ist in IBM AIX 7.2 ab Technology Level 4 verfügbar. Hoch komprimierbare Nachrichten mit einer Größe von mehr als 2 KB profitieren am meisten von der Nutzung der Bibliothek zlibNX, da sich dadurch die CPU-Belastung verringert. Um einem Nachrichtenkanalagenten (MCA) die Verwendung der Bibliothek zlibNX zu ermöglichen, setzen Sie die Umgebungsvariable AMQ_USE_ZLIBNX.

CONNNAME (Verbindungsname)

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 **CONNNAME** als durch Kommas getrennte Liste mit Namen von Maschinen für die angegebene **TRPTYPE** an. 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 Attribut **CLNTWGHT** angegeben 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.

Multi Unter [Multiplatforms](#) ist die Angabe des TCP/IP-Verbindungsnamensparameters eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM MQ automatisch einen Verbindungsnamen, wobei der Standardport vorausgesetzt und die aktuelle IP-Adresse des Systems verwendet wird. Sie können die Standardportnummer überschreiben, aber die aktuelle IP-Adresse des System weiter verwenden. Lassen Sie für jeden Verbindungsnamen den IP-Namen leer und übergeben Sie die Portnummer in runden Klammern; Beispiel:

```
(1415)
```

Die generierte **CONNNAME** wird immer in der Schreibweise mit Trennzeichen (IPv4) oder im Hexadezimalformat (IPv6) und nicht in Form eines alphanumerischen DNS-Hostnamens generiert.

Die maximale Länge des Namens ist von der Plattform abhängig:

- **Multi** 264 Zeichen.
- **z/OS** 48 Zeichen (siehe [Anmerkung 1](#)).

Transporttyp TCP

CONNNAME ist der Hostname oder die Netzadresse der fernen Maschine (oder der lokalen Maschine bei Clusterempfängerkanälen). Beispiel: (ABC.EXAMPLE.COM), (2001:DB8:0:0:0:0:0:0) oder (127.0.0.1). Sie kann die Portnummer enthalten, z. B. (MACHINE(123)).

z/OS Er kann den IP-Namen (IP_name) einer dynamischen DNS-Gruppe oder einen Network Dispatcher-Eingabeport enthalten.

Wenn Sie eine IPv6-Adresse in einem Netz verwenden, das nur IPv4 unterstützt, wird der Verbindungsname nicht aufgelöst. In einem Netz, das IPv4 und IPv6 verwendet, bestimmt der Verbindungsname zusammen mit der lokalen Adresse, welcher IP-Stack verwendet wird. Weitere Informationen hierzu finden Sie im Thema [„LOCLADDR \(Lokale Adresse\)“](#) auf Seite 109.

Transporttyp LU 6.2

Multi Wenn TPNAME und MODENAME angegeben sind, verwenden Sie den vollständig qualifizierten Namen der Partner-LU. Wenn TPNAME und MODENAME leer sind, geben Sie den Namen des CPI-C-Nebeninformationsobjekts für die bestimmte Plattform an.

z/OS Der Wert kann in zwei Formen angegeben werden:

- Name der logischen Einheit

Angaben zur logischen Einheit für den Warteschlangenmanager; diese setzen sich aus dem Namen der logischen Einheit, dem TP-Namen sowie (optional) dem Modusnamen zusammen. Dieser Name kann in einem von drei Formaten angegeben werden:

Tabelle 55. Namen von logischen Einheiten und Formate	
Format	Beispiel
LU-Name	IGY12355
LU-Name/TP-Name	IGY12345/APING
LU-Name/TP-Name/Modusname	IGY12345/APINGD/#INTER

Beim ersten Format müssen der Name des Transaktionsprogramms und der Modusname mit den Attributen TPNAME und MODENAME angegeben werden. Bei den beiden anderen Formaten müssen diese Attribute leer sein. Für Clientverbindungskanäle ist nur das erste Format erlaubt.

- Symbolischer Name

Symbolischer Bestimmungsname für die Angaben zur logischen Einheit für den Warteschlangenmanager, wie im Datensatz mit den Nebeninformationen definiert. Die Attribute TPNAME und MODENAME müssen leer sein. Beachten Sie, dass sich bei Clusterempfängerkanälen die Nebeninformationen in den anderen Warteschlangenmanagern des Clusters befinden. In diesem Fall kann es sich auch um einen Namen handeln, den ein Exit für die automatische Kanaldefinition in die entsprechenden LU-Informationen für den lokalen Warteschlangenmanager auflösen kann.

Der angegebene oder implizierte Name der logischen Einheit (LU) kann der LU-Name einer generischen VTAM-Ressourcengruppe sein.

Übertragungsprotokoll NetBIOS

CONNNAME ist der in der fernen Maschine definierte NetBIOS-Name.

Übertragungsprotokoll SPX

CONNNAME ist eine SPX-artige Adresse aus einer 4-Byte-Netzadresse, einer 6-Byte-Knotenadresse und einer 2-Byte-Socket-Nummer. Geben Sie diese Werte im Hexadezimalformat ein. Trennen Sie die Netz- und Knotenadresse durch einen Punkt, und setzen Sie die Socket-Nummer in eckige Klammern. For example:

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

Wenn Sie die Socket-Nummer übergehen, wird die SPX-Standardsocketnummer von IBM MQ verwendet. Der Standardwert ist X'5E86'.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server

Es ist optional für Serverkanäle, sofern der Serverkanal nicht ausgelöst wird. In diesem Fall muss es einen Verbindungsnamen angeben.

- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Anmerkung:

1. Bei der Länge des Namens können Sie die Begrenzung auf 48 Zeichen auf eine der folgenden Arten umgehen:
 - Richten Sie die DNS-Server z. B. so ein, dass Sie als Hostnamen "myserver" anstelle von "myserver.location.company.com" verwenden können.
 - Verwenden Sie IP-Adressen.
2. Die Definition des Übertragungsprotokolls ist in [TRPTYPE](#) enthalten.

CONVERT (Nachricht konvertieren)

Dieses Attribut gibt an, dass die Nachricht vor der Übertragung in das vom empfangenden System benötigte Format umgewandelt werden muss.

Anwendungsnachrichtendaten werden in der Regel von der empfangenden Anwendung umgewandelt. Wenn sich der ferne Warteschlangenmanager jedoch auf einer Plattform befindet, von der die Datenumwandlung nicht unterstützt wird, verwenden Sie dieses Kanalattribut, um anzugeben, dass die Nachricht **vor** der Übertragung in das Format umgewandelt werden muss, das für das empfangende System erforderlich ist.

Die gültigen Werte sind 'Ja' (yes) und 'Nein' (no). Wenn Sie 'Ja' (yes) angeben, werden die Anwendungsdaten in der Nachricht vor dem Senden umgewandelt, sofern Sie einen Namen der integrierten Formate angegeben haben oder ein Datenumwandlungsexit für ein benutzerdefiniertes Format verfügbar ist (siehe Abschnitt Datenumwandlungsexits schreiben). Wenn Sie **Nein** angeben, werden die Anwendungsdaten in der Nachricht vor dem Senden nicht umgewandelt.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Kanalattribute für MQSC-Schlüsselwörter (D-L)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter, beginnend mit den Buchstaben *D* bis *L*.

DEFRECON (Standardverbindungswiederholung)

Gibt an, ob eine Clientverbindung automatisch die Verbindung zu einer Clientanwendung wiederherstellt, wenn die Verbindung unterbrochen wird.

Folgende Werte sind möglich:

NO (Standardwert)

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

YES

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

QMGR

Wenn der Client nicht durch **MQCONN** überschrieben wird, stellt er die Verbindung automatisch wieder her, aber nur mit demselben Warteschlangenmanager. Die Option **QMGR** hat dieselbe Wirkung wie **MQCNO_RECONNECT_Q_MGR**.

Inaktiviert

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

Dieses Attribut ist nur für Clientverbindungskanäle gültig.

DESCR (Beschreibung)

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.

DISCINT (Unterbrechungsintervall)

Dieses Attribut definiert die Zeit, nach der ein Kanal geschlossen wird, wenn in diesem Zeitraum keine Nachricht mehr eintrifft.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Serververbindung nur unter Verwendung des TCP-Protokolls
- Clustersender
- Clusterempfänger

Dieses Attribut gibt ein Zeitlimit in Sekunden an.

Für Server-zu-Server-Nachrichtenkanäle (Server-, Clustersender-, Sender- und Clusterempfängerkanal) wird das Intervall ab dem Punkt gemessen, an dem ein Stapel endet, d. h. wenn die Stapelgröße erreicht ist 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 Abschlusss Austausch von Steuerdaten zwischen den beiden Enden des Nachrichtenkanals zwischen Servern enthält einen Hinweis auf die Ursache für das Schließen. Damit wird sichergestellt, dass die entsprechende Seite des Kanals für einen Neustart verfügbar bleibt.

Sie können eine beliebige Anzahl von Sekunden zwischen null und 999.999 angeben, wobei der Wert null bedeutet, dass die Verbindung nicht getrennt wird. Warten Sie unbegrenzt.

Der Standardwert ist 6000 Sekunden (100 Minuten) für Nachrichtenkanäle zwischen Servern und 0 (kein Zeitlimit) für Serververbindungskanäle. Sie können den Standardwert ändern, der für neue Kanäle verwendet wird, die Sie erstellen, indem Sie die Standardkanalobjekte ändern. Ändern Sie beispielsweise das Attribut DISCINT unter SYSTEM.DEF.SENDER, um einen neuen Standardwert für neue Senderkanäle festzulegen, die Sie definieren.

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 ein Serververbindungskanalprogramm für diese Dauer keine Kommunikation vom Partnerclient empfangen hat, wird die Verbindung beendet.

Das Inaktivitätsintervall für Serververbindung gilt zwischen IBM MQ -API-Aufrufen von einem Client.

Anmerkung: Ein MQGET mit Warteaufruf mit potenziell langer Laufzeit wird nicht als Inaktivität klassifiziert und überschreitet daher nie das Zeitlimit, da DISCINT abläuft.



Achtung: Der Wert für das Unterbrechungsintervall wirkt sich auf die Leistung aus.

A low value (for example a few seconds) can be detrimental to system performance by constantly stopping and re-starting the channel. Ein hoher Wert (mehr als eine Stunde) kann bedeuten, dass Systemressourcen ohne Vorteile verbraucht 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 für Nachrichtenkanäle zwischen Servern ist 6000 Sekunden (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. Wenn es für Ihre Umgebung geeignet ist, können Sie diesen Wert ändern, entweder für jeden

einzelnen Kanal oder durch Ändern des Attributs DISCONT in den Standardkanaldefinitionen (z. B. SYSTEM.DEF.SENDER für Senderkanäle), bevor Sie eigene Kanäle erstellen.

Weitere Informationen finden Sie im Abschnitt [Kanäle anhalten und stilllegen](#).

HBINT (Intervall der Überwachungssignale)

Dieses Attribut gibt die geschätzte Zeit zwischen zwei Überwachungssignalen an, die der sendende Nachrichtenkanalagent (Message Channel Agent, MCA) überbrücken muss, wenn keine Nachrichten in der Übertragungswarteschlange vorhanden sind.

Der Austausch von Überwachungssignalen hebt die Blockierung des empfangenden Nachrichtenkanalagenten auf, der auf eingehende Nachrichten wartet, bzw. darauf, dass das Unterbrechungsintervall abläuft. Wenn der empfangende MCA entblockt ist, kann er die Kanalverbindung trennen, ohne den Ablauf des Unterbrechungsintervalls abzuwarten. Durch den Austausch von Überwachungssignalen werden auch die für große Nachrichten zugewiesenen Speicherpuffer freigegeben und Warteschlangen geschlossen, die auf der Empfangsseite des Kanals offen gelassen wurden.

Der Wert wird in Sekunden angegeben und muss zwischen 0 und 999999 liegen. Bei Angabe von Null werden keine Überwachungssignale gesendet. Der Standardwert ist 300. Dieser Wert ist nur sinnvoll, wenn er deutlich kleiner ist als das Unterbrechungsintervall.

Bei Anwendungen, die IBM MQ classes for Java, JMS oder .NET-APIs verwenden, wird der HBINT-Wert auf eine der folgenden Arten bestimmt:

- Entweder durch den Wert im SVRCONN-Kanal, der von der Anwendung verwendet wird.
- Oder durch den Wert im CLNTCONN-Kanal, wenn die Anwendung so konfiguriert wurde, dass sie eine CCDT verwendet.

Bei Serververbindungs- und Clientverbindungskanälen können Überwachungssignale unabhängig voneinander sowohl vom Server als auch vom Client gesendet werden. Wenn innerhalb des Überwachungssignalintervalls keine Daten über den Kanal übertragen wurden, sendet der MQI-Agent der Clientverbindung ein Überwachungssignal, das vom MQI-Agenten der Serververbindung durch ein weiteres Überwachungssignal beantwortet wird. Dies geschieht unabhängig vom Status des Kanals, also zum Beispiel unabhängig davon, ob der Kanal während eines API-Aufrufs oder während des Wartens auf Benutzereingaben vom Client inaktiv ist. Zudem kann der MQI-Agent der Serververbindung den Austausch von Überwachungssignalen mit dem Client initialisieren, und zwar auch hier unabhängig vom Status des Kanals. Um zu verhindern, dass der MQI-Agent der Serververbindung und der MQI-Agent der Clientverbindung gleichzeitig an die jeweils andere Seite Überwachungssignale abgeben, wird das Überwachungssignal des Servers erst ausgegeben, wenn über den Kanal für die Länge des Überwachungssignalintervalls plus 5 Sekunden keine Daten mehr übertragen wurden.

Bei Serververbindungs- und Clientverbindungskanälen im Kanalmodus einer früheren Version von IBM WebSphere MQ 7.0 findet der Austausch von Überwachungssignalen nur statt, wenn ein Servernachrichtenkanalagent auf einen MQGET-Befehl mit angegebener WAIT-Option wartet, den er für eine Clientanwendung ausgegeben hat.


Weitere Informationen dazu, wie MQI-Kanäle in beiden Modi arbeiten, finden Sie im Abschnitt [Sharing-Conversations \(MQLONG\)](#).

KAINT (Keepalive-Intervall)

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 Multiplatforms können Sie auf den Parameter zugreifen und ihn ändern. Er wird jedoch nur gespeichert und weitergeleitet, eine funktionale Implementierung des Parameters findet nicht statt. Wenn Sie die Funktionalität des Parameters KAINTE benötigen, verwenden Sie den Parameter für das In-

tervall der Überwachungssignale (HBINT) gemäß Abschnitt „[HBINT \(Intervall der Überwachungssignale\)](#)“ auf Seite 108.

Damit dieses Attribut wirksam wird, muss die TCP/IP-Keepalive-Funktion aktiviert sein.

- **z/OS** Unter z/OS aktivieren Sie Keepalive, indem Sie den MQSC-Befehl ALTER QMGR TCPKEEP(YES) ausgeben.
- **Multi** Unter Multiplattformstritt sie auf, wenn der Parameter KEEPALIVE=YES in der TCP-Zeilengruppe in der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen, `qm.ini`, oder über IBM MQ Explorer angegeben ist.

Keepalive muss mithilfe des Datenbestands für die TCP-Profilkonfiguration auch in TCP/IP selbst aktiviert werden.

Der Wert gibt einen Zeitraum (in Sekunden) an und muss im Bereich 0 bis 99999 liegen. Der Keepalive-Intervallwert 0 zeigt an, dass kein kanalspezifisches Keepalive für den Kanal aktiviert ist und nur der systemweite in TCP/IP eingestellte Keepalive-Wert verwendet wird. Sie können KAINTE auch auf AUTO festlegen. (Dies ist der Standardwert.) Dann basiert der Keepalive-Wert wie folgt auf dem vereinbarten Wert des Intervalls der Überwachungssignale (HBINT):

Vereinbarter HBINT-Wert	KAINTE
>0	Vereinbarter HBINT-Wert + 60 Sekunden
0	0

Dieses Attribut ist für alle Kanaltypen gültig.

Der Wert wird für alle Kanäle ignoriert, bei denen TransportType (TRPTYPE) eine andere Einstellung als TCP oder SPX besitzt.

LOCLADDR (Lokale Adresse)

Dieses Attribut gibt die lokale Kommunikationsadresse für den Kanal an.

Anmerkung: AMQP-Kanäle unterstützen nicht dasselbe Format von LOCLADDR wie andere IBM MQ-Kanäle. Weitere Informationen finden Sie unter „[#unique_51/unique_51_Connect_42_locladdr_amqp](#)“ auf Seite 112.

LOCLADDR für alle Kanäle außer AMQP-Kanälen

Dieses Attribut gilt nur, wenn der Transporttyp (TRPTYPE) TCP/IP ist. 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_ADD-RESS_LENGTH.

Wenn Sie **LOCLADDR** nicht angeben, wird automatisch eine lokale Adresse zugeordnet.

Beachten Sie, dass Sie **LOCLADDR** für einen C-Client mithilfe der Definitionstabelle für Clientkanäle (CCDT) definieren können.

Alle Parameter sind optional. Durch Übergehen des ip-addr-Teils der Adresse wird die Konfiguration einer festen Portnummer für eine IP-Firewall aktiviert. Das Übergehen der Portnummer ist hilfreich, um einen bestimmten Netzadapter auszuwählen, ohne eine eindeutige, lokale Portnummer identifizieren zu müssen. Der TCP/IP-Stack generiert eine eindeutige Portnummer.

Geben Sie [, [ip-addr] [(low-port[,high-port])]] für jede zusätzliche lokale Adresse mehrmals an. Verwenden Sie mehrere lokale Adressen, wenn Sie eine bestimmte Untergruppe von lokalen Netzadaptern angeben möchten. Sie können auch [, [ip-addr] [(low-port[,high-port])]] verwenden, um eine bestimmte lokale Netzadresse auf verschiedenen Servern darzustellen, die Teil einer Multi-Instanz-Warteschlangenmanagerkonfiguration sind.

ip-addr

ip-addr (IP-Adresse) wird in einem von drei Formaten angegeben:

IPv4-Dezimalschreibweise mit Punkten

Beispiel: 192.0.2.1

IPv6-Hexadezimalschreibweise

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

Alphanumerisches Hostnamensformat

Beispiel: WWW.EXAMPLE.COM

low-port und high-port

low-port (niedrigster_Port) und high-port (höchster_Port) sind Portnummern in runden Klammern.

Die folgende Tabelle zeigt, wie der Parameter **LOCLADDR** verwendet werden kann:

<i>Tabelle 57. Beispiele für die Verwendung des Parameters LOCLADDR</i>	
LOCLADDR	Bedeutung
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

Tabelle 57. Beispiele für die Verwendung des Parameters **LOCLADDR** (Forts.)

LOCLADDR	Bedeutung
(1000)	Kanal wird lokal an Port 1000 gebunden.
(1000,2000)	Kanal wird lokal an einen Port im Bereich von 1000 bis 2000 gebunden.

Beim Start eines Kanals bestimmen die Werte für den Verbindungsnamen (CONNAME) und die lokale Adresse (LOCLADDR), welcher IP-Stack für die Kommunikation verwendet wird. Der verwendete IP-Stack wird wie folgt bestimmt:

- Wenn für das System nur ein IPv4-Stack konfiguriert ist, wird immer der IPv4-Stack verwendet. Wenn Sie eine lokale Adresse (LOCLADDR) oder einen Verbindungsnamen (CONNAME) als IPv6-Netzadresse angeben, wird ein Fehler generiert und der Start des Kanals schlägt fehl.
- Wenn für das System nur ein IPv6-Stack konfiguriert ist, wird immer der IPv6-Stack verwendet. Wenn 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 Sie die lokale Adresse (LOCLADDR) als Hostname angeben, der in IPv4- und IPv6-Adressen aufgelöst wird, bestimmt der Verbindungsname (CONNAME), welcher der Stacks verwendet wird. Wenn 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.

Multi Unter Multiplatforms können Sie einen lokalen Standardwert für die lokale Adresse festlegen, der für alle Senderkanäle verwendet wird, für die keine lokale Adresse definiert ist. Der Standardwert wird definiert, indem die Umgebungsvariable MQ_LCLADDR vor dem Starten des Warteschlangenmanagers festgelegt wird. Das Format des Werts stimmt mit dem des MQSC-Attributs LOCLADDR überein.

Lokale Adressen für Clustersenderkanäle

Clustersenderkanäle übernehmen immer die im Zielwarteschlangenmanager definierte Konfiguration des entsprechenden Clusterempfängerkanals. Dies gilt auch dann, wenn ein lokal definierter Clustersenderkanal mit demselben Namen vorhanden ist; in diesem Fall wird die manuelle Definition nur für die Eingangskommunikation verwendet.

Aus diesem Grund sollte man sich nicht auf die im Clusterempfängerkanal definierte lokale Adresse (LOCLADDR) verlassen, da die IP-Adresse wahrscheinlich nicht dem System gehört, auf dem die Clustersender erstellt werden. Aus diesem Grund sollte die lokale Adresse (LOCLADDR) auf dem Clusterempfänger nur verwendet werden, wenn es einen Grund gibt, nur die Ports, nicht jedoch die IP-Adresse für alle 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.

LOCLADDR für AMQP-Kanäle

AMQP-Kanäle unterstützen ein anderes Format von LOCLADDR als andere IBM MQ-Kanäle:

LOCLADDR (*ip-addr*)

LOCLADDR ist die lokale Kommunikationsadresse für den Kanal. Verwenden Sie diesen Parameter, um das Verwenden einer bestimmten IP-Adresse durch den Client zu erzwingen. LOCLADDR ist auch nützlich, um einen Kanal zu zwingen, eine IPv4 -oder IPv6 -Adresse zu verwenden, wenn eine Auswahl verfügbar ist, oder um einen bestimmten Netzadapter auf einem System mit mehreren Netzadaptern zu verwenden.

Die maximale Länge von LOCLADDR ist MQ_LOCAL_ADDRESS_LENGTH.

Wenn Sie LOCLADDR nicht angeben, wird automatisch eine lokale Adresse zugeordnet.

ip-addr

ip-addr ist eine einzelne Netzadresse, die in einem von drei Formaten angegeben wird:

IPv4-Dezimalschreibweise mit Punkten

Beispiel: 192.0.2.1

IPv6-Hexadezimalschreibweise

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

Alphanumerisches Hostnamensformat

Beispiel: WWW.EXAMPLE.COM

Bei Angabe einer IP-Adresse wird nur das Adressformat überprüft. Eine Überprüfung der eigentlichen IP-Adresse findet nicht statt.

Weitere Informationen finden Sie unter [Mit automatisch definierten Clustersenderkanälen arbeiten](#) .

LONGRTY (Zähler für lange Wiederholungen)

Dieses Attribut gibt die maximale Anzahl an Wiederholungsversuchen des Kanals an, seinem Partner eine Sitzung zuzuordnen.

Das Attribut **long retry count** kann zwischen 0 und 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Wenn der erste Zuordnungsversuch fehlschlägt, wird die Anzahl *Zähler für kurze Wiederholungen* verringert und der Kanal wiederholt die verbleibende Anzahl von Versuchen. Wenn er immer noch fehlschlägt, wiederholt er *Zähler für lange Wiederholungsversuche* so oft mit einem Intervall von *Langes Wiederholungsintervall* 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.

z/OS Unter z/OS kann ein Kanal keine Wiederholung eingeben, wenn die maximale Anzahl Kanäle (**MAXCHL**) überschritten wurde.

Multi Bei Multiplatforms muss ein Kanalinitiator aktiv sein, damit ein Wiederholungsversuch unternommen werden kann. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

Wenn der -Kanalinitiator (unter z/OS) oder der -Kanal (unter Multiplatforms) gestoppt wird, während der Kanal wiederholt wird, werden der *Zähler für kurze Wiederholungen* und der *Zähler für lange Wiederholungen* zurückgesetzt, wenn der Kanalinitiator oder der Kanal erneut gestartet wird oder wenn eine Nachricht erfolgreich im Senderkanal eingereicht wird. Wenn jedoch der -Kanalinitiator (unter z/OS) oder -Warteschlangenmanager (auf Multiplatforms) beendet und erneut gestartet wird, werden der *Zähler für kurze Wiederholungen* und der *Zähler für lange Wiederholungen* 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.

Multi Unter Multiplatforms:

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, wenn die erste Nachricht erfolgreich über den Kanal versendet wird, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., wenn der lokale Kanal die Anzahl der versendeten Nachrichten bestätigt hat.
2. Der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* werden beim Neustart des Kanals zurückgesetzt.

LONGTMR (Langes Wiederholungsintervall)

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, *Zähler für lange Wiederholungsversuche* Mal in diesem langen Intervall eine Verbindung herzustellen, nachdem er *Zähler für kurze Wiederholungen* 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

Kanalattribute für MQSC-Schlüsselwörter (M)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter beginnend mit dem Buchstaben *M*.

MAXINST (Maximale Anzahl Instanzen)

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals oder AMQP-Kanals an, die gestartet werden können.

Maximale Anzahl von Instanzen bei Verbindungen über Serververbindungskanäle

Für einen Serververbindungskanal gibt dieses Attribut die maximale Anzahl gleichzeitiger Instanzen eines Serververbindungskanals an, die gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wird der Wert heruntergesetzt, sodass er unter der Anzahl der momentan aktiven Instanzen des Serververbindungskanals liegt, hat dies keine Auswirkungen auf die aktiven Kanäle. Neue Instanzen können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

Maximale Anzahl von Instanzen bei Verbindungen über AMQP-Kanäle

Für einen AMQP-Kanal gibt dieses Attribut die maximale Anzahl gleichzeitiger Instanzen eines AMQP-Kanals an, die gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wenn ein Client versucht, eine Verbindung herzustellen, und die Anzahl der verbundenen Clients den Wert von MAXINST erreicht hat, schließt der Kanal die Verbindung mit einem Schließvorgangsframe. Der Schließvorgangsframe enthält die folgende Nachricht:

```
amqp:resource-limit-exceeded
```

Wenn ein Client eine Verbindung mit einer ID herstellt, die bereits verbunden ist (d. h., er führt eine Clientübernahme durch), ist die Übernahme erfolgreich, unabhängig davon, ob die Anzahl der verbundenen Clients MAXINST erreicht hat.

Weitere Informationen finden Sie unter [Grenzwerte für Serververbindungskanäle](#).

MAXINSTC (Maximale Anzahl Instanzen pro Client)

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals an, die auf einem einzelnen Client gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wird der Wert heruntergesetzt, sodass er unter der Anzahl der momentan auf den einzelnen Clients aktiven Instanzen des Serververbindungskanals liegt, hat dies keine Auswirkungen auf die aktiven Kanäle. Neue Instanzen auf diesen Clients können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

Dieses Attribut ist nur für Serververbindungskanäle gültig.

Weitere Informationen finden Sie unter [Grenzwerte für Serververbindungskanäle](#).

MAXMSGL (Maximale Nachrichtenlänge)

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

Multi Bei Multiplatforms geben Sie einen Wert an, der größer oder gleich Null und kleiner oder gleich der maximalen Nachrichtenlänge für den Warteschlangenmanager ist. Weitere Informationen finden Sie im Abschnitt zum MAXMSGL-Parameter des Befehls ALTER QMGR in [ALTER QMGR](#).

z/OS Geben Sie unter IBM MQ for z/OS einen Wert an, der größer-gleich null und kleiner-gleich 104 857 600 Byte ist (d. h. 100 MB).

Da es auf IBM MQ-Systemen unterschiedliche Implementierungsarten auf verschiedenen Plattformen gibt, ist in manchen Anwendungen die für die Nachrichtenverarbeitung verfügbare Länge möglicherweise

begrenzt. Dieser Wert muss einer Größe entsprechen, die Ihr System problemlos verarbeiten kann. Beim Kanalstart wird die niedrigere der zwei Zahlen für die beiden Kanaldaten übernommen.

Anmerkung: Sie können für den Kanal eine maximale Nachrichtenlänge von 0 verwenden. Dies bedeutet, dass die Größe auf den Maximalwert des lokalen Warteschlangenmanagers gesetzt werden soll.

Beim Hinzufügen der digitalen Signatur und des Schlüssels zur Nachricht erhöht Advanced Message Security die Länge der Nachricht.

Dieses Attribut ist für alle Kanaltypen gültig.

MCANAME (Name des Nachrichtenkanalagenten)

Dieses Attribut ist reserviert, wenn es nur auf Leerzeichen gesetzt werden darf und eine maximale Länge von 20 Zeichen hat.

MCATYPE (Nachrichtenkanalagententyp)

Dieses Attribut kann den Nachrichtenkanalagenten als Prozess oder Thread angeben.

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

Anmerkung: 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. Wenn Sie `thread` angeben, wird der MCA in einem Thread des `AMQRMPPA`-Prozesses oder des `RUNMQCHI`-Prozesses ausgeführt, wenn `MQNOREMPOOL` angegeben ist. Auf der Maschine, die die eingehenden Zuordnungen empfängt, wird der MCA als Thread ausgeführt, wenn Sie `RUNMQLSR` verwenden. Bei Eingabe von `inetd` wird er als Prozess ausgeführt.

 Unter IBM MQ für z/OS wird dieses Attribut nur für Kanäle vom Typ Clusterempfänger unterstützt.

 Auf anderen Plattformen ist dieses Attribut für Kanaltypen gültig:

- Sender
- Server
- Requester
- Clustersender
- Clusterempfänger

MCAUSER (Benutzer-ID des Nachrichtenkanalagenten)

Dieses Attribut ist die Benutzer-ID (eine Zeichenfolge), die vom MCA zur Autorisierung für den Zugriff auf IBM MQ-Ressourcen verwendet wird.

Anmerkung: Alternativ dazu kann eine Benutzer-ID für einen Kanal, unter der dieser ausgeführt werden soll, über die Verwendung von Kanalauthentifizierungsdatensätzen bereitgestellt werden. Über Kanala-

Authentifizierungsdatensätze 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 MQ for Windows kann die Benutzer-ID mit dem Format `user@domain` domänenqualifiziert werden, 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

MODENAME (LU 6.2-Modusname)

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 nur für folgende Kanaltypen gültig:

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

MONCHL (Überwachung)

Dieses Attribut steuert die Erfassung von Online-Überwachungsdaten.

Mögliche Werte:

QMGR

Die Erfassung der Daten aus der Online-Überwachung wird aus der Einstellung des Attributs MONCHL des Warteschlangenmanagerobjekts übernommen. Dies ist der Standardwert.

OFF

Die Erfassung von Onlineüberwachungsdaten wird für diesen Kanal inaktiviert.

LOW

Niedrige Datenerfassungsrate mit minimalen Auswirkungen auf die Systemleistung. Die abgebildeten Überwachungsergebnisse sind möglicherweise nicht aktuell.

MITTEL

Mittlere Datenerfassungsrate mit begrenzten Auswirkungen auf die Leistung des Systems.

HIGH

Hohe Datenerfassungsrate mit möglichen Auswirkungen auf die Leistung des Systems. Die abgebildeten Überwachungsergebnisse sind jedoch sehr aktuell.

Dieses Attribut ist für folgende Kanaltypen gültig:

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

Weitere Informationen zu Überwachungsdaten finden Sie im Abschnitt [Überwachungsdaten von Warteschlangen und Kanälen anzeigen](#).

MRDATA (Benutzerdaten des Nachrichtenwiederholungsexits)

Dieses Attribut gibt Daten an, die an den Kanalexit für Nachrichtenwiederholung übergeben werden, wenn er aufgerufen wird.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

MREXIT (Name des Nachrichtenwiederholungsexits)

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 „RCVEXIT (Name des Empfangsexits)“ auf [Seite 122](#). Sie können jedoch nur einen Exit für Nachrichtenwiederholungen angeben.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

MRRTY (Nachrichtenwiederholungszähler)

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

MRTMR (Nachrichtenerholungsintervall)

Dieses Attribut gibt das Mindestintervall in Millisekunden an, das verstreichen muss, bevor der Kanal die MQPUT-Operation wiederholen kann.

Dieses Attribut steuert die Aktion des MCA nur, wenn kein Name für den Exit für Nachrichtenerholung 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

MSGDATA (Benutzerdaten des Nachrichtensexits)

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. Weitere Informationen finden Sie unter [„RCVDATA \(Benutzerdaten des Empfangsexits\)“](#) auf Seite 122.

Dieses Attribut ist für folgende Kanaltypen gültig:

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

MSGEXIT (Nachrichtensexitname)

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 [„RCVEXIT \(Name des Empfangsexits\)“](#) auf Seite 122.

Dieses Attribut ist für folgende Kanaltypen gültig:

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

Kanalattribute für MQSC-Schlüsselwörter (N-R)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter, beginnend mit den Buchstaben *N* bis *R*.

NETPRTY (Netzverbindungspriorität)

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Verwenden Sie das Attribut NETPRTY, um ein Netz als primäres Netz und ein weiteres Netz als Sicherungsnetz festzulegen. Bei einer Gruppe von Kanälen mit gleichem Rang wird durch Clustering der Pfad mit der höchsten Priorität ausgewählt, wenn mehrere Pfade zur Verfügung stehen.

Üblicherweise wird das Kanalattribut NETPRTY verwendet, um zwischen Netzen zu unterscheiden, die die gleichen Zielorte verbinden, aber unterschiedliche Aufwände verursachen oder unterschiedliche Geschwindigkeiten aufweisen.

Anmerkung: Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).

NPMSPEED (Geschwindigkeit nicht persistenter Nachrichten)

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](#).

Anmerkungen:

1.  Wenn die aktiven Wiederherstellungsprotokolle für IBM MQ for z/OS häufiger als erwartet wechseln und archivieren, da die über einen Kanal gesendeten Nachrichten nicht persistent sind, kann das Festlegen von NPMSPEED (FAST) sowohl auf der sendenden als auch auf der empfangenden Seite des Kanals das SYSTEM.CHANNEL.SYNCQ -Aktualisierungen.
2. Wenn es wegen der Aktualisierungen von SYSTEM.CHANNEL.SYNCQ zu einer hohen CPU-Auslastung kommt, kann das Festlegen von NPMSPEED(FAST) die CPU-Auslastung erheblich verringern.

Dieses Attribut ist für folgende Kanaltypen gültig:

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

Kennwort (Kennwort)

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.

z/OS Unter IBM MQ for z/OS ist dieses Attribut nur für Clientverbindungskanäle gültig.

Multi Auf anderen Plattformen ist dieses Attribut für Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender

PORT (Portnummer)

Geben Sie die Portnummer an, die für die Verbindung des AMQP-Clients verwendet wird.

Der Standardport für AMQP 1.0-Verbindungen ist 5672. Falls Port 5672 bereits verwendet wird, können Sie einen anderen Port angeben.

PUTAUT (PUT-Berechtigung)

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

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).

z/OS Unter z/OS die markierten Benutzer-IDs, deren Anzahl von der Einstellung für das hlq.RESLEVEL-Profil der MQADMIN-RACF-Klasse abhängt. Je nachdem, welche Zugangsstufe die Benutzer-ID des Kanalinitiators für das Profil hlq.RESLEVEL hat, sind null, ein oder zwei Benutzer-IDs markiert. Unter RESLEVEL- und Kanalinitiator-Verbindungen finden Sie Anweisungen zur Einsicht in die markierten Benutzer-IDs. Weitere Informationen zu markierten Benutzer-IDs finden Sie unter [Vom Kanalinitiator verwendete Benutzer-IDs](#).

Sie können eine der folgenden Optionen wählen:

Prozesssicherheit, auch als Standardberechtigung (DEF) bezeichnet

Die Standard-Benutzer-ID wird verwendet.

Multi Auf Multiplattformen ist die Benutzer-ID, die zum Prüfen der Öffnungsberechtigung für die Warteschlange verwendet wird, die des Prozesses oder Benutzers, der den MCA am empfangenden Ende des Nachrichtenkanals ausführt.

z/OS Unter z/OS wird je nach Anzahl der zu überprüfenden Benutzer-IDs entweder die aus dem Netz erhaltene oder die von [MCAUSER](#) abgeleitete Benutzer-ID verwendet.

Die Warteschlangen werden mit dieser Benutzer-ID und der Öffnungsoption MQOO_SET_ALL_CONTEXT geöffnet.

Kontextsicherheit (CTX)

Die Benutzer-ID aus den Kontextinformationen der Nachricht wird als alternative Benutzer-ID verwendet.

Die *UserIdentifier* im Nachrichtendeskriptor wird in das Feld *AlternateUserId* im Objektdeskriptor verschoben. Die Warteschlange wird mit den Öffnungsoptionen `MQOO_SET_ALL_CONTEXT` und `MQOO_ALTERNATE_USER_AUTHORITY` geöffnet.

Multi Auf Multiplatforms ist die Benutzer-ID, die verwendet wird, um die Öffnungsberechtigung für die Warteschlange für `MQOO_SET_ALL_CONTEXT` und `MQOO_ALTERNATE_USER_AUTHORITY` zu überprüfen, die des Prozesses oder Benutzers, der den MCA am empfangenden Ende des Nachrichtenkanals ausführt. Zur Überprüfung der Öffnungsberechtigung für die Warteschlange für `MQOO_OUTPUT` wird der *UserIdentifier* des Nachrichtendeskriptors verwendet.

z/OS Unter z/OS kann je nach Anzahl der zu überprüfenden Benutzer-IDs die aus dem Netz erhaltene oder die von `MCAUSER` abgeleitete Benutzer-ID verwendet werden, ebenso wie die Benutzer-ID aus den Kontextinformationen des Nachrichtendeskriptors.

Kontextsicherheit (CTX) wird für Serververbindungskanäle nicht unterstützt.

z/OS Sicherheit ONLYMCA (Nur Nachrichtenkanalagent)

Die von `MCAUSER` abgeleitete Benutzer-ID wird verwendet.

Die Warteschlangen werden mit der Öffnungsoption `MQOO_SET_ALL_CONTEXT` geöffnet.

Dieser Wert wird nur unter z/OS unterstützt.

z/OS Sicherheit ALTMCA (Alternativer Nachrichtenkanalagent)

Abhängig von der Anzahl der zu überprüfenden Benutzer-IDs kann die Benutzer-ID aus den Kontextinformationen (Feld *UserIdentifier*) im Nachrichtendeskriptor sowie die von `MCA`-Benutzer abgeleitete Benutzer-ID verwendet werden.

Dieser Wert wird nur unter z/OS unterstützt.

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

Weitere Informationen zu Sicherheit finden Sie hier:

- [Sicherheit](#)
- **ALW** [Sicherheit unter AIX, Linux, and Windows einrichten](#)
- **IBM i** [Sicherheit unter IBM i einrichten](#)
- **z/OS** [Sicherheit unter z/OS einrichten](#)

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- **z/OS** Serververbindung (nur z/OS)
- Clusterempfänger

QMNAME (Name des Warteschlangenmanagers)

Sein Attribut gibt den Namen des Warteschlangenmanagers oder der Warteschlangenmanagergruppe an, zu dem bzw. der eine IBM MQ MQI client -Anwendung eine Verbindung anfordern kann.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clientverbindung

QSGDISP (Disposition)

z/OS

Dieses Attribut gibt die Disposition des Kanals in einer Gruppe mit gemeinsamer Warteschlange an. Dieser Parameter ist nur unter z/OS gültig.

Folgende Werte sind möglich:

QMGR

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

RCVDATA (Benutzerdaten des Empfangsexits)

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. For example:

```
RCVDATA(exit1_data exit2_data)
MSGDATA(exit1_data,exit2_data)
SENDDATA(exit1_data, exit2_data)
```

ALW

Auf IBM MQ for UNIX- und Windows-Systemen ist die Länge der Zeichenfolge aus Exitnamen und Zeichenfolgen aus Benutzerdaten auf 500 Zeichen begrenzt.

IBM i

In IBM MQ for IBM i können Sie bis zu 10 Exitnamen angeben werden und die Länge der Benutzerdaten ist jeweils auf 32 Zeichen beschränkt.

z/OS

Unter IBM MQ for z/OS können Sie bis zu acht Zeichenfolgen aus Benutzerdaten angeben, von denen jede 32 Zeichen lang sein kann.

Dieses Attribut ist für alle Kanaltypen gültig.

RCVEXIT (Name des Empfangsexits)

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. Lassen Sie das Feld leer, wenn kein Kanalempfangsbenutzerexit aktiv ist.

Das Format und die maximale Länge dieses Attributs hängen von der Plattform ab:

- **z/OS** Unter z/OS handelt es sich um einen Lademodulnamen mit einer maximalen Länge von 8 Zeichen bzw. bei Clientverbindungskanälen mit maximal 128 Zeichen.

- ▶ **IBM i** Auf IBM i wird folgendes Format verwendet:

```
libname/progname
```

wenn es in CL-Befehlen verwendet wird.

Bei Angabe in IBM MQ-Befehlen (MQSC) hat es das Format:

```
progname libname
```

Dabei belegt *progname* die ersten 10 Zeichen und *libname* die zweiten 10 Zeichen (beide mit Leerzeichen aufgefüllt, falls erforderlich). Die maximale Länge der Zeichenfolge beträgt 20 Zeichen.

- ▶ **Linux** ▶ **AIX** Auf AIX and Linuxn wird folgendes Format verwendet:

```
libraryname(functionname)
```

Die maximale Länge der Zeichenfolge beträgt 40 Zeichen.

- ▶ **Windows** Auf Windowsn wird folgendes Format verwendet:

```
dllname(functionname)
```

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

▶ **z/OS** Bei der automatischen Definition des Clustersenderkanals unter z/OS werden Kanalexitnamen in das z/OS-Format konvertiert. Wenn Sie die Konvertierung von Exitnamen steuern möchten, können Sie einen Exit für die automatische Kanaldefinition eingeben. Weitere Informationen finden Sie unter [Exitprogramm für die automatische Kanaldefinition](#).

Sie können eine Namensliste von Empfangs-, Sende- oder Nachrichtenexitprogrammen angeben. Die Namen müssen durch ein Komma, ein Leerzeichen oder beides getrennt sein. For example:

```
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.

- ▶ **IBM i** Unter IBM MQ for IBM i können Sie bis zu 10 Exitnamen auflisten.
- ▶ **z/OS** Unter IBM MQ for z/OS können Sie bis zu acht Exitnamen auflisten.

Dieses Attribut ist für alle Kanaltypen gültig.

Kanalattribute für MQSC-Schlüsselwörter (S)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter, beginnend mit dem Buchstaben S.

SCYDATA (Benutzerdaten des Sicherheitsexits)

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.

SCYEXIT (Name des Sicherheitsexits)

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 hängen von der Plattform ab, wie bei [RCVEXIT](#). Sie können jedoch nur ein Sicherheitsexit angeben.

Weitere Informationen zu Sicherheitsexits finden Sie unter [Kanalsicherheitsexitprogramme](#).

Dieses Attribut ist für alle Kanaltypen gültig.

SENDDATA (Benutzerdaten des Sendeexits)

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 [RCVDATA](#).

Dieses Attribut ist für alle Kanaltypen gültig.

SENDEXIT (Sendeexitname)

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 hängen von der Plattform ab, wie bei [RCVEXIT](#).

Dieses Attribut ist für alle Kanaltypen gültig.

SEQWRAP (Folgenummernumlauf)

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 Kanalenden 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

SHORTRTY (Zähler für kurze Wiederholungen)

Dieses Attribut gibt die maximale Anzahl an Wiederholungsversuchen des Kanals an, seinem Partner eine Sitzung zuzuordnen.

Das Attribut SHORTRTY kann von 0 bis 999 999 999 gesetzt werden.

Wenn mehrere IP-Adressen im Kanal definiert wurden und eine erneute Verbindung erforderlich ist, wertet IBM MQ die Kanaldefinition aus und versucht, eine Verbindung zu jeder IP-Adresse in der Reihenfolge

herzustellen, in der sie definiert ist, bis entweder eine erfolgreiche Verbindung hergestellt wurde oder alle Adressen versucht wurden.

In diesem Fall bezieht sich SHORTRTY auf die Gesamtzahl der Versuche, die der gesamte Kanal versucht, die Verbindung wiederherzustellen, und nicht auf die einzelnen IP-Adressen.

Wenn der erste Zuordnungsversuch fehlschlägt, wird der Wert für *Zähler für kurze Wiederholungen* 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 *Zähler für lange Wiederholungsversuche* die Anzahl der Wiederholungen mit einem Intervall von *Langes Wiederholungsintervall* zwischen den einzelnen Versuchen. Kann die Operation weiterhin nicht ausgeführt werden, wird der Kanal geschlossen.

z/OS Unter z/OS kann ein Kanal keine Wiederholung eingeben, wenn die maximale Anzahl Kanäle (**MAXCHL**) überschritten wurde.

Multi Bei Multiplatforms muss ein Kanalinitiator aktiv sein, damit ein Wiederholungsversuch unternommen werden kann. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

Wenn der -Kanalinitiator (unter z/OS) oder der -Kanal (unter Multiplatforms) wird gestoppt, während der Kanal erneut versucht wird, der *Zähler für kurze Wiederholungen* und der *Zähler für lange Wiederholungen* werden zurückgesetzt, wenn der Kanalinitiator oder der Kanal erneut gestartet wird oder wenn eine Nachricht erfolgreich im Senderkanal eingereicht wird. Wenn jedoch der -Kanalinitiator (unter z/OS) oder der Warteschlangenmanager (auf Multiplatforms) wird beendet und erneut gestartet, der *Zähler für kurze Wiederholungen* und der *Zähler für lange Wiederholungen* werden 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.

Multi Unter Multiplatforms:

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, wenn die erste Nachricht erfolgreich über den Kanal versendet wird, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., wenn der lokale Kanal die Anzahl der versendeten Nachrichten bestätigt hat.
2. Der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* werden beim Neustart des Kanals zurückgesetzt.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

SHORTTMR (Kurzes Wiederholungsintervall)

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.

Wenn mehrere IP-Adressen im Kanal definiert wurden und eine erneute Verbindung erforderlich ist, wertet IBM MQ die Kanaldefinition aus und versucht, eine Verbindung zu jeder IP-Adresse in der Reihenfolge herzustellen, in der sie definiert ist, bis entweder eine erfolgreiche Verbindung hergestellt wurde oder alle Adressen versucht wurden.

In diesem Fall bezieht sich SHORTTMR darauf, wie lange der gesamte Kanal auf den Neustart des Verbindungsprozesses wartet, und nicht auf die einzelnen IP-Adressen.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

SPLPROT (Sicherheitsrichtlinienschutz)



Dieses Attribut gibt an, wie ein Server-zu-Server-Nachrichtenkanalagent mit dem Nachrichtenschutz umgehen soll, wenn AMS aktiv ist und eine gültige Richtlinie vorhanden ist.

Dieses Attribut kann auf Folgendes gesetzt werden:

PASSTHRU

Auf Sender-, Server-, Empfänger- und Requesterkanälen

REMOVE

Auf Sender- und Serverkanälen

ASPOLICY

Auf Empfänger- und Requesterkanälen

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester

SSLCAUTH (SSL-Clientauthentifizierung)

Das Attribut **SSLCAUTH** gibt an, ob der Kanal ein TLS-Zertifikat von einem TLS-Client empfangen und authentifizieren muss.

SSLCAUTH ist ein optionales Attribut. Für dieses Attribut sind die folgenden Werte möglich:

OPTIONAL

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

ERFORDERLICH

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

Der Standardwert ist REQUIRED.

Sie können einen Wert für **SSLCAUTH** in einer Nicht-TLS-Kanaldefinition angeben. Das heißt, eine Kanaldefinition, in der das Attribut [SSLCIPH](#) fehlt oder leer ist.

Weitere Informationen zu **SSLCAUTH** finden Sie in den Abschnitten [DEFINE CHANNEL \(MQTT\)](#) und [Sicherheit](#).

Das Attribut **SSLCAUTH** ist für alle Kanaltypen gültig, die einen Kanalinitialisierungsfluss empfangen können, mit Ausnahme von Senderkanälen. Dieses Attribut ist für folgende Kanaltypen gültig:

- Server
- Empfänger
- Requester
- Serververbindung

- Clusterempfänger

SSLCIPH (SSL-Verschlüsselungsspezifikation)

Das Attribut **SSLCIPH** gibt einen Alias oder eine einzelne benannte CipherSpec für eine TLS-Verbindung an.

Jede IBM MQ-Kanaldefinition enthält das Attribut **SSLCIPH**. Der Wert besteht aus einer Zeichenfolge mit einer maximalen Länge von 32 Zeichen.

Das Attribut **SSLCIPH** ist nur für Kanäle vom Transporttyp (**TRPTYPE**) TCP-gültig. Ist **TRPTYPE** nicht TCP-, werden die Daten ignoriert und es wird keine Fehlernachricht ausgegeben.

Anmerkungen:

- Das Attribut **SSLCIPH** kann einen leeren Wert enthalten, d. h., Sie verwenden kein TLS. Wenn ein Ende des Kanals über ein leeres Attribut **SSLCIPH** verfügt, muss auch das andere Ende des Kanals über ein leeres Attribut **SSLCIPH** verfügen.

► **Multi** Falls SecureCommsOnly aktiviert ist, wird die Kommunikation mit unverschlüsseltem Text nicht unterstützt und der Kanal kann nicht gestartet werden.

- Wenn **SSLCIPH** einen nicht leeren Wert enthält, kann der Wert alternativ ein Alias oder eine benannte CipherSpec sein. Die Kanäle vereinbaren die stärkste CipherSpec, die von beiden Kanalenden unterstützt wird.
- Ein vollständig verwalteter .NET-Client kann den speziellen Wert *NEGOTIATE festlegen. Diese Option ermöglicht dem Kanal die Auswahl der neuesten Protokollversion, die vom .NET-Framework unterstützt wird sowie das Aushandeln eines CipherSpec-Werts, der vom Server unterstützt wird.

Das Attribut **SSLCIPH** ist nur für Kanäle vom Transporttyp (**TRPTYPE**) TCP-gültig. Ist **TRPTYPE** nicht TCP-, werden die Daten ignoriert und es wird keine Fehlernachricht ausgegeben.

Weitere Informationen zu **SSLCIPH** finden Sie unter [KANAL DEFINIEREN](#) und [CipherSpecs angeben](#).

SSLPEER (SSL-Peer)

Das Attribut **SSLPEER** wird verwendet, um den definierten Namen (DN) des Zertifikats vom Peer-Warteschlangenmanager oder Client am anderen Ende eines IBM MQ-Kanals zu überprüfen.

Anmerkung: Alternativ können zur Beschränkung von Verbindungen auf bestimmte Kanäle durch Überprüfung des definierten TLS-Namens auch Kanalauthentifizierungsdatensätze verwendet werden. Über die Authentifizierungsdatensätze für Kanäle können verschiedene Muster für definierte Namen des Zertifikatinhabers in TLS auf denselben Kanal angewendet werden. Wenn sowohl **SSLPEER** im Kanal als auch 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 DN nicht mit dem Wert **SSLPEER** 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.

Die maximale Länge des Attributs **SSLPEER** hängt von der Plattform ab:

- ► **z/OS** Unter z/OS beträgt die maximale Länge des Attributs 256 Byte.
- ► **Multi** Auf allen anderen Plattformen beträgt sie 1024 Byte.

Kanalauthentifizierungsdatensätze bieten eine größere Flexibilität bei der Verwendung von **SSLPEER** und unterstützen eine maximale Länge von 1024 Byte auf allen Plattformen.

Die Überprüfung der **SSLPEER**-Attributwerte hängt auch von der Plattform ab:

- **z/OS** 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.
- **Multi** Unter Multiplatforms, die **SSLPEER** unterstützen, wird die Gültigkeit der Zeichenfolge bei der ersten Eingabe geprüft.

Sie können einen Wert für **SSLPEER** in einer Nicht-TLS-Kanaldefinition angeben, in der das **SSLCIPH**-Attribut fehlt oder leer ist. In diesem Fall können Sie TLS temporär für die Fehlerbehebung inaktivieren, ohne die TLS-Parameter entfernen und später erneut eingeben zu müssen.

Das Attribut **SSLPEER** ist für alle Kanaltypen gültig.

Weitere Informationen zu **SSLPEER** finden Sie unter SET CHLAUTH, Securing und Kanalauthentifizierungsdatensätze.

STATCHL (Kanalstatistik)

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.

z/OS Anz/OS Systeme: Durch die Aktivierung dieses Parameters wird einfach die Erfassung statistischer Daten aktiviert, unabhängig von dem von Ihnen ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Dieses Attribut ist für folgende Kanaltypen gültig:

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

Kanalattribute für MQSC-Schlüsselwörter (T-Z)

Eine alphabetische Liste der Kanalattribute für MQSC-Schlüsselwörter, beginnend mit den Buchstaben T bis Z.

TPNAME (LU 6.2 Transaktionsprogrammname)

Dieses Attribut wird bei LU 6.2-Verbindungen verwendet. Es handelt sich um den Namen bzw. den generischen Namen des Transaktionsprogramms (MCA), das am fernen Ende der Verbindung ausgeführt wird.

Bei Verwendung der Nebeninformationen für die SNA-Kommunikation ist der Transaktionsprogrammname im CPI-C-Nebenobjekt für Kommunikation oder in den APPC-Nebeninformationen definiert, und dieses Attribut muss leer bleiben. Andernfalls wird dieser Name von den Sender- und Requesterkanälen benötigt.

Der Name kann bis zu 64 Zeichen lang sein.

Der Name muss auf den SNA-Transaktionsprogrammnamen festgelegt sein, es sei denn, der CONNAME enthält einen Nebenobjektnamen, dann muss der Name leer bleiben. Der eigentliche Name wird stattdessen dem CPI-C-Nebenobjekt für Kommunikation bzw. der APPC-Datei für Nebeninformationen entnommen.

Diese Informationen werden auf verschiedenen Plattformen auf verschiedene Weisen festgelegt; weitere Informationen zum Einrichten der Kommunikation für Ihre Plattform finden Sie im Abschnitt [Verteilte Steuerung von Warteschlangen konfigurieren](#).

Dieses Attribut ist für folgende Kanaltypen gültig:

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

TPROOT (Topicstammverzeichnis)

Dieses Attribut gibt den Themen-Root für einen AMQP-Kanal an.

Mit dem Attribut TPROOT können Sie einen Themen-Root für einen AMQP-Kanal angeben. Durch die Verwendung dieses Attributs ist sichergestellt, dass eine MQ Light-Anwendung bei ihrer Implementierung auf einem Warteschlangenmanager Nachrichten nicht in den Bereichen der Themenstruktur veröffentlicht, die von anderen Anwendungen verwendet werden, oder Nachrichten aus derartigen Bereichen subskribiert.

Der Standardwert für TPROOT ist SYSTEM.BASE.TOPIC. Bei diesem Wert weist die Themenzeichenfolge, die von einem AMQP-Client für die Veröffentlichung oder Subskription verwendet wird, kein Präfix auf und der Client kann Nachrichten mit anderen MQ-Publish/Subscribe-Anwendungen austauschen. Damit AMQP-Clients unter einem Themenpräfix Veröffentlichungen und Subskriptionen vornehmen können, müssen Sie zunächst ein MQ-Themenobjekt mit einer Themenzeichenfolge erstellen, die mit dem gewünschten Präfix festgelegt wird. Anschließend müssen Sie den Wert des Attributs TPROOT für den AMQP-Kanal in den Namen des von Ihnen erstellten MQ-Themenobjekts ändern. Im folgenden Beispiel ist ein Themen-Root dargestellt, der für den AMQP-Kanal MYAMQP auf den Wert APPGROUP1.BASE.TOPIC gesetzt wurde:

```
DEFINE CHANNEL(MYAMQP) CHLTYPE(AMQP) TPROOT(APPGROUP1.BASE.TOPIC) PORT(5673)
```

Anmerkung: Wenn der TPROOT-Attributwert oder die von ihm unterstützte Themenzeichenfolge geändert wird, kann dies dazu führen, dass bereits vorhandene AMQP-Themen und ihre Nachrichten verwaist sind.

TRPTYPE (Transporttyp)

Dieses Attribut gibt den zu verwendenden Transporttyp an.

Folgende Werte sind möglich:

Tabelle 58. Optionen für Transporttyp	
Wert	Transporttyp
LU62	LU 6.2
TCP	TCP/IP
NETBIOS	NetBIOS „1“ auf Seite 130
SPX	SPX „1“ auf Seite 130

Anmerkungen:

1. Zur Verwendung unter Windows. Kann auch unter z/OS verwendet werden, um Clientverbindungskanäle für die Verwendung unter Windows zu definieren.

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

USECLTID (Client-ID verwenden)

Geben Sie an, ob die Client-ID für die Verbindung bei einem AMQP-Kanal verwendet werden soll. Setzen Sie den Wert auf Yes oder No.

USEDLQ (Warteschlange für nicht zustellbare Mail verwenden)

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 Attribut DEADQ des Warteschlangenmanagers den Namen einer Warteschlange für nicht zustellbare Nachrichten angibt, wird diese Warteschlange verwendet. Andernfalls ist das Verhalten wie bei NO.

USERID (Benutzer-ID)


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.

Wenn auf der Empfangsseite verschlüsselte Kennwörter verwendet werden und die LU 6.2-Software ein anderes Verschlüsselungsverfahren verwendet, schlägt der Kanalstart mit Einzelangaben zu ungültigen Sicherheitsbedingungen fehl. Dies kann vermieden werden, indem die empfangende SNA-Konfiguration dahingehend geändert wird, dass entweder

- die Kennwortersetzung inaktiviert wird oder
- eine Sicherheitsbenutzer-ID und das entsprechende Kennwort definiert werden.

 Unter IBM MQ für z/OS ist dieses Attribut nur für Clientverbindungskanäle gültig.

 Auf anderen Plattformen ist dieses Attribut für Kanaltypen gültig:

- Sender
- Server

- Requester
- Clientverbindung
- Clustersender

XMITQ (Name der Übertragungswarteschlange)

Dieses Attribut gibt den Namen der Übertragungswarteschlange an, aus der Nachrichten abgerufen werden.

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 Kanäle des Typs Sender oder Server erforderlich und für andere Kanaltypen nicht gültig.

IBM MQ -Clusterbefehle und -Attribute

Es gibt MQSC- und PCF-Clusterbefehle, mit denen Sie einen Cluster aktualisieren oder zurücksetzen oder einen Clusterwarteschlangenmanager anzeigen, fortsetzen oder aussetzen können. Außerdem haben die MQSC- und PCF-Befehle, die Kanäle, Warteschlangen und Warteschlangenmanager definieren, Attribute, die für Cluster gelten. Einige dieser Attribute werden vom Algorithmus für das Clusterauslastungsmanagement verwendet.

MQSC-Befehle

Die MQSC-Befehle werden angezeigt, als ob Sie vom Systemadministrator in der Befehlskonsole eingegeben worden wären. Sie müssen diese Befehle aber nicht auf diese Weise ausgeben. Je nach Plattform gibt es verschiedene andere Methoden zur Ausgabe dieser Befehle, zum Beispiel:

-  In IBM MQ for IBM i führen Sie MQSC-Befehle interaktiv über Option 26 von **WRKMQM** aus. Ebenso können Sie CL-Befehle verwenden oder MQSC-Befehle in einer Datei speichern und den CL-Befehl **STRMQMMQSC** verwenden.
-  Unter z/OS können Sie die Funktion COMMAND des Dienstprogramms **CSQUTIL**, die Betriebs- 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 Clustername, der mit dem Attribut CLUSTER angegeben wird, bis zu 48 Zeichen lang sein.

Eine Liste mit Clusternamen, die mit dem Attribut CLUSNL angegeben wird, kann bis zu 256 Namen enthalten. Zur Erstellung einer Cluster-Namensliste verwenden Sie den Befehl **DEFINE NAMELIST**.

IBM MQ Explorer

Die grafische Benutzerschnittstelle von IBM MQ Explorer kann einen Cluster mit Repository-Warteschlangenmanagern unter IBM WebSphere MQ for z/OS 6.0 oder höher verwalten. Die Benennung eines weiteren Repositorys auf einem separaten System ist nicht erforderlich. Bei früheren Versionen von IBM MQ for z/OS kann IBM MQ Explorer keinen Cluster mit Repository-Warteschlangenmanagern verwalten. Daher müssen Sie ein zusätzliches Repository auf einem System benennen, das von IBM MQ Explorer verwaltet werden kann.

Unter IBM MQ for Windows und IBM MQ for Linux können Sie auch IBM MQ Explorer für die Arbeit mit Clustern verwenden. Ebenso können Sie den eigenständigen IBM MQ Explorer-Client verwenden.

Bei Verwendung von IBM MQ Explorer können Sie Clusterwarteschlangen anzeigen und den Status von Clustersender- und Clusterempfängerkanälen abfragen. In IBM MQ Explorer gibt es zwei Assistenten, die Sie durch folgende Tasks führen:

- Cluster erstellen
- Unabhängigen Warteschlangenmanager mit einem Cluster verbinden

PCF-Entsprechungen zu MQSC-Befehlen für die Verwaltung von Clustern

MQSC-Befehl	Entsprechender PCF-Befehl
DISPLAY CLUSQMGR	MQCMD_INQUIRE_CLUSTER_Q_MGR
REFRESH CLUSTER	MQCMD_REFRESH_CLUSTER
RESET CLUSTER	MQCMD_RESET_CLUSTER
RESUME QMGR	MQCMD_RESUME_Q_MGR_CLUSTER
SUSPEND QMGR	MQCMD_SUSPEND_Q_MGR_CLUSTER

Zugehörige Informationen

[Clustering: Best Practices für REFRESH CLUSTER verwenden](#)

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

Die Befehle DEFINE CHANNEL, ALTER CHANNEL und DISPLAY CHANNEL beinhalten zwei bestimmte CHLTYPE-Parameter für Cluster: CLUSRCVR und CLUSSDR. Um einen Clusterempfängerkanal zu definieren, verwenden Sie den Befehl DEFINE CHANNEL, wobei Sie CHLTYPE (CLUSRCVR) angeben. Viele der Attribute in einer Definition für einen Clusterempfängerkanal sind die gleichen Attribute wie die in einer Definition für einen Empfänger- oder Senderkanal. Um einen Clustersenderkanal zu definieren, verwenden Sie den Befehl DEFINE CHANNEL, wobei Sie CHLTYPE (CLUSSDR) angeben, sowie viele der Attribute, die Sie auch bei der Definition eines Senderkanals verwenden.

Für die Definition eines Clustersenderkanals ist es nicht mehr erforderlich, den Namen des vollständigen Warteschlangenmanager-Repositorys anzugeben. Wenn Ihnen die Namenskonvention bekannt ist, die für Kanäle in Ihrem Cluster verwendet wird, können Sie eine CLUSSDR-Definition mithilfe der Konstruktion +QMNAME+ vornehmen. Die +QMNAME+-Konstruktion wird unter z/OS nicht unterstützt. Nachdem die Verbindung hergestellt wurde, ändert IBM MQ den Namen des Kanals und ersetzt +QMNAME+ durch den korrekten Namen des vollständigen Warteschlangenmanager-Repositorys. 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.

CLLWGHT

Der Parameter CLLWGHT weist zum Zweck einer gleichmäßigen Clusterauslastung den Kanälen einen Gewichtungsfaktor zu. CLLWGHT 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 CLLWGHT, um die Auswahl der Zieladresse zu beeinflussen, sodass mehr Nachrichten über einen bestimmten Kanal versendet werden können. Alle Kanalgewichtungsattribute haben standardmäßig den gleichen Standardwert. Das Gewichtungsattribut ermöglicht es Ihnen, einem Kanal auf einer leistungsfähigen UNIX-Maschine eine stärkere Gewichtung zu geben als einem Kanal auf einem kleinen Desktop-PC. Eine stärkere Gewichtung bedeutet, dass der Algorithmus für Clusterauslastung die UNIX-Maschine häufiger als Zieladresse für Nachrichten auswählt als den PC.

CONNAME

Der Parameter CONNAME, der in der Definition für einen Clusterempfängerkanal angegeben ist, wird im gesamten Cluster verwendet, um die Netzadresse des Warteschlangenmanagers zu bestimmen. Achten Sie darauf, für den Parameter CONNAME einen Wert auszuwählen, der in Ihrem gesamten IBM MQ-Cluster aufgelöst wird. Verwenden Sie keinen generischen Namen. Beachten Sie, dass der Wert, der im Clusterempfängerkanal angegeben wird, Vorrang vor jedem Wert hat, der in einem entsprechenden Clustersenderkanal angegeben wird.

Diese Attribute in den Befehlen DEFINE CHANNEL und ALTER CHANNEL gelten ebenso für den Befehl DISPLAY CHANNEL.

Anmerkung: Automatisch definierte Clustersenderkanäle erhalten ihre Attribute von der jeweiligen Definition für den entsprechenden Clusterempfängerkanal im Empfangswarteschlangenmanager. Auch wenn es sich um einen manuell definierten Clustersenderkanal handelt, werden seine Attribute automatisch geändert, um sicherzustellen, dass sie mit den Attributen der entsprechenden Clusterempfängerdefinition übereinstimmen. Sie können beispielsweise einen CLUSRCVR-Kanal definieren, ohne eine Portnummer im Parameter CONNAME anzugeben, und dabei manuell einen CLUSSDR-Kanal definieren, bei dem eine Portnummer angegeben wird. Wenn der automatisch definierte CLUSSDR den manuell definierten 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 von automatisch definierten Clustersenderkanälen zu prüfen.

Verwenden Sie den Befehl DISPLAY CHSTATUS, um den Status von Clustersender- oder Clusterempfängerkanälen anzuzeigen. Dieser Befehl zeigt den Status sowohl von manuell definierten Kanälen als auch von automatisch definierten Kanälen an.

Die entsprechenden PCFs lauten MQCMD_CHANGE_CHANNEL, MQCMD_COPY_CHANNEL, MQCMD_CREATE_CHANNEL und MQCMD_INQUIRE_CHANNEL.

Weglassen des Werts CONNAME in einer CLUSRCVR-Definition

Unter Umständen können Sie den Wert CONNAME in einer CLUSRCVR-Definition weglassen. Sie dürfen den Wert CONNAME unter z/OS nicht übergehen.

Multi Unter [Multiplatforms](#) ist die Angabe des TCP/IP-Verbindungsnamensparameters eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM MQ automatisch einen Verbindungsnamen, wobei der Standardport vorausgesetzt und die aktuelle IP-Adresse des Systems verwendet wird. Sie können die Standardportnummer überschreiben, aber die aktuelle IP-Adresse des System weiter verwenden. Lassen Sie für jeden Verbindungsnamen den IP-Namen leer und übergeben Sie die Portnummer in runden Klammern; Beispiel:

```
(1415)
```

Die generierte **CONNAME** wird immer in der Schreibweise mit Trennzeichen (IPv4) oder im Hexadezimalformat (IPv6) und nicht in Form eines alphanumerischen DNS-Hostnamens generiert.

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 IBM 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 aus dem lokalen Warteschlangenmanager ausgeben, können Sie sehen, dass für CONNAME ein Leerzeichen angegeben ist.

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

Zugehörige Konzepte

[Lastausgleich in Clustern](#)

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

z/OS [Asynchronous behavior of CLUSTER commands on z/OS](#)

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

[In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute](#)

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

[In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute](#)

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

[DISPLAY CLUSQMGR](#)

Verwenden Sie den Befehl DISPLAY CLUSQMGR, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

[REFRESH CLUSTER](#)

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

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.

„Kanalattribute für Clusterauslastungsausgleich“ auf Seite 147

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

Die Befehle `DEFINE QLOCAL`, `DEFINE QREMOTE` und `DEFINE QALIAS`

Die Clusterattribute in den Befehlen `DEFINE QLOCAL`, `DEFINE QREMOTE` und `DEFINE QALIAS` und die drei äquivalenten `ALTER`-Befehle sind:

Cluster

Gibt den Namen des Clusters an, dem die Warteschlange zugeordnet ist.

CLUSNL

Gibt eine Namensliste mit Clusternamen an.

DEFBIND

Gibt die Bindung an, die verwendet werden soll, wenn `MQOO_BIND_AS_Q_DEF` im `MQOPEN`-Aufruf einer Anwendung angegeben wird. Folgende Optionen stehen für dieses Attribut zur Verfügung:

- Wird `DEFBIND(OPEN)` angegeben, wird die Warteschlangenkennung beim Öffnen der Warteschlange an eine bestimmte Instanz der Clusterwarteschlange gebunden. `DEFBIND(OPEN)` ist die Standardeinstellung für dieses Attribut.
- Wird `DEFBIND(NOTFIXED)` angegeben, so wird die Warteschlangenkennung nicht an eine Instanz der Clusterwarteschlange gebunden.
- Wird `DEFBIND(GROUP)` angegeben, kann eine Anwendung fordern, dass alle Nachrichten einer Nachrichtengruppe an dieselbe Zielinstanz übergeben werden.

Wenn mehrere Warteschlangen mit demselben Namen in einem Warteschlangenmanagercluster zugänglich gemacht werden, können Anwendungen auswählen, ob alle Nachrichten aus dieser Anwendung an eine einzelne Instanz (`MQOO_BIND_ON_OPEN`) gesendet werden sollen, damit der Lastausgleichsalgorithmus für jede Nachricht das am besten geeignete Ziel auswählen kann (`MQOO_BIND_NOT_FIXED`), oder ob eine Anwendung anfordern kann, dass eine "Gruppe" von Nachrichten vollständig derselben Zielinstanz zugeordnet wird (`MQOO_BIND_ON_GROUP`). Der Lastausgleich wird zwischen Nachrichtengruppen erneut ausgeführt (ohne dass ein `MQCLOSE` und ein `MQOPEN` für die Warteschlange erforderlich sind).

Wenn Sie in einer Warteschlangendefinition `DEFBIND` angeben, wird die Warteschlange mit einem der Attribute `MQBND_BIND_ON_OPEN`, `MQBND_BIND_NOT_FIXED` oder `MQBND_BIND_ON_GROUP` definiert. Entweder `MQBND_BIND_ON_OPEN` oder `MQBND_BIND_ON_GROUP` muss bei der Verwendung von Gruppen mit Clustern angegeben werden.

Sie sollten das Attribut `DEFBIND` in allen Instanzen derselben Clusterwarteschlange auf denselben Wert setzen.

CLWLANK

Weist zu Auslastungsmanagementzwecken Warteschlangen einen Rangfolgefaktor zu. Der `CLWLANK`-Parameter wird von Modellwarteschlangen nicht unterstützt. Der Algorithmus für Clusterauslastung

wählt eine Zielwarteschlange mit dem höchsten Rang aus. Standardmäßig wird CLWLRANK für alle Warteschlangen auf null gesetzt.

Wenn es sich bei dem Zielort um einen Warteschlangenmanager in einem anderen Cluster handelt, können Sie den Rang jedes zwischengeschalteten Gateway-Warteschlangenmanagers am Schnittpunkt benachbarter Cluster festlegen. Nachdem unter den zwischengeschalteten Warteschlangenmanagern eine Rangfolge festgelegt wurde, wählt der Algorithmus für Clusterauslastung ordnungsgemäß jeweils den Zielwarteschlangenmanager aus, der sich näher am Zielort befindet.

Das Gleiche gilt für Aliaswarteschlangen. Die Auswahl nach Rangfolge erfolgt noch, bevor der Kanalstatus überprüft wird. Daher stehen auch nicht verfügbare Warteschlangenmanager für die Auswahl zur Verfügung. Dadurch wird es einer Nachricht ermöglicht, durch das Netz weitergeleitet zu werden, anstatt dass sie zwischen zwei möglichen Zieladressen wählen muss (wie es bei der Prioritätseinstellung der Fall wäre). Wenn nun ein Kanal nicht in dem Bereich gestartet wird, den sein Rang angibt, wird die Nachricht nicht an den nächsthöheren Rang weitergeleitet, sondern wartet, bis ein Kanal für diese Zieladresse verfügbar ist (d. h. die Nachricht wird in der Übertragungswarteschlange gespeichert).

CLWLPRTY

Weist zu Auslastungsmanagementzwecken Warteschlangen einen Prioritätsfaktor zu. Der Algorithmus für Clusterauslastung wählt eine Zielwarteschlange mit der höchsten Priorität aus. Standardmäßig wird für alle Warteschlangen die Priorität null angegeben.

Bei zwei möglichen Zielwarteschlangen können Sie dieses Attribut dazu verwenden, eine Zieladresse als Ausfallsicherung für die andere Zieladresse einzurichten. Die Prioritätsauswahl erfolgt nach der Überprüfung des Kanalstatus. Alle Nachrichten werden an die Warteschlange mit der höchsten Priorität gesendet, es sei denn, der Status des Kanals zu dieser Zieladresse ist nicht so günstig wie der Status anderer Kanäle zu anderen Zieladressen. Dies bedeutet, dass nur die am besten zugänglichen Zieladressen für die Auswahl zur Verfügung stehen. Dadurch erfolgt eine Priorisierung unter mehreren Zieladressen, die alle verfügbar sind.

CLWLUSEQ

Gibt das Verhalten einer MQPUT-Operation für eine Warteschlange an. Dieser Parameter gibt das Verhalten einer MQPUT-Operation an, wenn die Zielwarteschlange über eine lokale Instanz und mindestens eine ferne Clusterinstanz verfügt (außer wenn der MQPUT-Aufruf von einem Clusterkanal stammt). Dieser Parameter ist nur für lokale Warteschlangen gültig.

Folgende Werte sind möglich: QMGR (das Verhalten entspricht dem, was im CLWLUSEQ-Parameter der Warteschlangenmanagerdefinition angegeben ist), ANY (der Warteschlangenmanager behandelt die lokale Warteschlange zum Zweck der Lastverteilung wie eine weitere Instanz der Clusterwarteschlange) und LOCAL (die lokale Warteschlange ist das einzige Ziel der MQPUT-Operation, wobei angegeben wird, dass für die lokale Warteschlange das Einreihen aktiviert ist). Das Verhalten der MQPUT-Operation ist vom Algorithmus für das Clusterauslastungsmanagement abhängig.

Befehle DISPLAY QUEUE und DISPLAY QCLUSTER

Die Attribute in den Befehlen DEFINE QLOCAL, DEFINE QREMOTE und DEFINE QALIAS gelten ebenso für den DISPLAY QUEUE-Befehl.

Um Informationen zu Clusterwarteschlangen anzuzeigen, geben Sie im Befehl DISPLAY QUEUE einen Warteschlangentyp wie QCLUSTER oder das Schlüsselwort CLUSINFO an, oder verwenden Sie den Befehl DISPLAY QCLUSTER.

Die Befehle DISPLAY QUEUE und DISPLAY QCLUSTER geben den Namen des Warteschlangenmanagers zurück, der die Warteschlange enthält (oder die Namen aller Warteschlangenmanager, wenn es sich um mehrere Instanzen einer Warteschlange handelt). Außerdem werden der Systemname für jeden 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 Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

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

DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

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.

„Clusterauslastungsausgleich-Warteschlangenattribute“ auf Seite 149

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

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

Um anzugeben, dass ein WS-Manager 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 geben Sie anschließend das Attribut `REPOSNL(namelist)` im Befehl **ALTER QMGR** an:

```
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 Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

DISPLAY CLUSQMGR

Verwenden Sie den Befehl DISPLAY CLUSQMGR, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

REFRESH CLUSTER

Geben Sie den Befehl REFRESH CLUSTER in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

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.

„Clusterauslastungsausgleich-Warteschlangenmanager-Attribute“ auf Seite 150

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

DISPLAY CLUSQMGR

Verwenden Sie den Befehl DISPLAY CLUSQMGR, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

Wenn Sie diesen Befehl in einem Warteschlangenmanager mit einem vollständigen Repository ausgeben, gelten die zurückgegebenen Informationen für jeden Warteschlangenmanager im Cluster. Andernfalls gelten die zurückgegebenen Informationen nur für die jeweils betroffenen Warteschlangenmanager, d. h. für jeden Warteschlangenmanager, an den versucht wurde, eine Nachricht zu senden, und für jeden Warteschlangenmanager mit einem vollständigen Repository.

Die Daten beinhalten die meisten Kanalattribute sowohl für Clustersender- als auch für Clusterempfängerkanäle. Zusätzlich können die folgenden Attribute angezeigt werden:

CHANNEL

Der Name des Clusterempfängerkanals für den Warteschlangenmanager.

CLUSDATE

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

Cluster

Gibt an, zu welchen Clustern der Warteschlangenmanager gehört.

CLUSTIME

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

DEFTYPE

Gibt die Definition des Warteschlangenmanagers an. Für DEFTYPE sind folgende Werte möglich:

CLUSSDR

Ein Clustersenderkanal wurde auf dem lokalen Warteschlangenmanager administrativ definiert, aber noch nicht vom Ziel-Warteschlangenmanager erkannt. Dieser Status tritt auf, wenn ein lokaler Warteschlangenmanager einen manuellen Clustersenderkanal definiert hat, der empfangende Warteschlangenmanager die Clusterinformationen jedoch nicht akzeptiert hat. Ursache hierfür kann sein, dass der Kanal aufgrund fehlender Verfügbarkeit oder eines Fehlers in der Clustersenderkonfiguration nicht aufgebaut wurde, zum Beispiel aufgrund einer fehlenden Übereinstimmung für die Eigenschaft CLUSTER in der Sender- und Empfängerdefinition. Hierbei handelt es sich um einen vorübergehenden Zustand oder Fehlerstatus, der untersucht werden sollte.

CLUSSDRA

Dieser Wert gibt einen automatisch erkannten Cluster-Warteschlangenmanager an, lokal ist kein Clustersenderkanal definiert. Dies ist der DEFTYPE-Wert für Cluster-Warteschlangenmanager, für die der lokale Warteschlangenmanager nicht über eine lokale Konfiguration verfügt, aber über die er informiert ist. Beispiel:

- Wenn der lokale Warteschlangenmanager ein Warteschlangenmanager für ein vollständiges Repository ist, muss der DEFTYPE-Wert für alle Teilrepository-Warteschlangenmanager im Cluster verwendet werden.
- Wenn der lokale Warteschlangenmanager ein Teilrepository ist, kann dies der Host einer Clusterwarteschlange sein, der von diesem lokalen Warteschlangenmanager oder von einem zweiten Warteschlangenmanager für ein vollständiges Repository verwendet wird, mit dem dieser Warteschlangenmanager zusammenarbeiten soll.

Wenn der DEFTYPE-Wert CLUSSDRA lautet und sowohl der lokale als auch der ferne Warteschlangenmanager vollständige Repositories für den benannten Cluster sind, ist die Konfiguration

nicht korrekt, da ein lokal definierter Clustersenderkanal zum Umwandeln in den DEFTYPE-Wert CLUSSDRB definiert sein muss.

CLUSSDRB

Ein Clustersenderkanal wurde auf dem lokalen Warteschlangenmanager administrativ definiert und als gültiger Clusterkanal vom Ziel-Warteschlangenmanager erkannt. Dies ist der erwartete DEFTYPE-Wert für einen manuell konfigurierten Warteschlangenmanager für ein vollständiges Repository eines Teilrepository-Warteschlangenmanagers. Außerdem muss er der DEFTYPE-Wert für jeden CLUSQMGR von einem vollständigen Repository zu einem anderen vollständigen Repository im Cluster sein. Manuelle Clustersenderkanäle dürfen nicht zu Teilrepositorys 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 Repositorys für den Cluster sind, finden Sie unter der Eigenschaft QMTYPE.

Weitere Informationen zum Definieren von Clusterkanälen finden Sie unter Clusterkanäle.

QMTYPE

Gibt an, ob der Warteschlangenmanager über ein vollständiges Repository oder nur über ein Teilrepository verfügt.

STATUS

Gibt den Status des Clustersenderkanals für diesen Warteschlangenmanager an.

SUSPEND

Gibt an, ob der Warteschlangenmanager ausgesetzt wurde.

Version

Die Version der IBM MQ Installation, der der Cluster-Warteschlangenmanager zugeordnet ist.

Die Version hat das Format VVRRMMFF:

- VV: Version
- RR: Release
- MM: Wartungsstufe
- FF: Fix-Level

XMITQ

Die vom Warteschlangenmanager verwendete Clusterübertragungswarteschlange.

Siehe dazu den Befehl `DISPLAY QCLUSTER`. Er wird kurz in `DISPLAY QUEUE` und im Abschnitt Die Befehle `DISPLAY QUEUE` und `DISPLAY QCLUSTER` von „In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute“ auf Seite 135 beschrieben. Beispiele zur Verwendung des Befehls `DISPLAY QCLUSTER` finden Sie in den Informationen zu "`DISPLAY QCLUSTER`" und "`DIS QCLUSTER`".

Zugehörige Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

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

REFRESH CLUSTER

Geben Sie den Befehl REFRESH CLUSTER in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

Befehle 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.

MQSC-Befehl **DISPLAY CLUSQMGR**

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: Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle

interessierten Warteschlangenmanager hochladen, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

Zugehörige Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

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

DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

Befehle 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.

Zugehörige Informationen

Clustering: Best Practices für REFRESH CLUSTER verwenden

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

Sie können den Befehl **RESET CLUSTER** nur in Warteschlangenmanagern für ein vollständiges Repository ausgeben. Je nachdem, ob Sie mit Namen oder ID auf den Warteschlangenmanager verweisen, kann der Befehl zwei Formen annehmen.

1.

```
RESET CLUSTER( clustername
) QMNAME( qmname ) ACTION(FORCEREMOVE) QUEUES(NO)
```

2.

```
RESET CLUSTER( clustername
) QMID( qmid ) ACTION(FORCEREMOVE) QUEUES(NO)
```


Sie können nicht beide Optionen QMNAME und QMID angeben. Wenn Sie QMNAME verwenden, wird der Befehl nicht ausgeführt, wenn sich im Cluster mehrere Warteschlangenmanager mit diesem Namen befinden. Verwenden Sie QMID anstelle von QMNAME, um sicherzustellen, dass der Befehl **RESET CLUSTER** ausgeführt wird.

Bei der Angabe von QUEUES(NO) in einem **RESET CLUSTER**-Befehl handelt es sich um die Standardeinstellung. Durch die Angabe von QUEUES(YES) werden Verweise auf Clusterwarteschlangen entfernt, die zu den Warteschlangenmanagern des Clusters gehören. Diese Verweise werden zusätzlich zur Entfernung des Warteschlangenmanagers aus dem Cluster selbst entfernt.

Die Verweise werden entfernt, auch wenn der Clusterwarteschlangenmanager im Cluster nicht sichtbar ist, weil dessen Entfernung zuvor ohne die Option QUEUES 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 IBM MQ diese Definitionen entfernt (was automatisch geschieht), können Sie den Befehl **RESET CLUSTER** ausgeben, um die Definitionen bereits früher zu löschen. Alle anderen Warteschlangenmanager im Cluster werden daraufhin informiert, dass dieser Warteschlangenmanager nicht mehr verfügbar ist.

Wenn ein Warteschlangenmanager vorübergehend beschädigt ist, können Sie die anderen Warteschlangenmanager im Cluster benachrichtigen, bevor sie versuchen, Nachrichten an diesen Warteschlangenmanager zu senden. Der Befehl **RESET CLUSTER** entfernt den beschädigten Warteschlangenmanager. Später, wenn der beschädigte Warteschlangenmanager wieder funktioniert, verwenden Sie den Befehl **REFRESH CLUSTER**, um die Wirkung von **RESET CLUSTER** umzukehren und den Warteschlangenmanager an den Cluster zurückzugeben. Wenn sich der Warteschlangenmanager in einem Publish/Subscribe-Cluster befindet, müssen Sie alle erforderlichen Proxy-Subskriptionen wiederherstellen. Siehe [Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster](#).

Anmerkung: Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager hochladen, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt [Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken](#).

Die Verwendung des Befehls **RESET CLUSTER** ist die einzige Möglichkeit, automatisch definierte Cluster-senderkanäle zu löschen.

Wichtig: Wenn der zu entfernende automatisch definierte Kanal unbestätigt ist, entfernt **RESET CLUSTER** diesen Kanal nicht sofort. In dieser Situation müssen Sie vor dem Befehl **RESET CLUSTER** einen Befehl [RESOLVE CHANNEL](#) ausgeben.

Es ist unwahrscheinlich, dass Sie diesen Befehl unter normalen Umständen benötigen. Der IBM Support empfiehlt Ihnen möglicherweise, den Befehl auszugeben, um die Clusterinformationen zu bereinigen, die von Clusterwarteschlangenmanagern gehalten werden. Verwenden Sie diesen Befehl nicht als verkürztes Verfahren, um Warteschlangenmanager aus einem Cluster zu entfernen. Die korrekte Vorgehensweise beim Entfernen eines Warteschlangenmanagers aus einem Cluster wird im Abschnitt [Warteschlangenmanager aus einem Cluster entfernen](#) beschrieben.

Repositorys speichern Daten nur für 90 Tage. Daher kann ein Warteschlangenmanager, dessen Entfernung erzwungen wurde, nach diesem Zeitraum die Verbindung zum Cluster wiederherstellen. Der Warteschlangenmanager stellt die Verbindung automatisch wieder her, es sei denn, er wurde gelöscht. Wenn Sie einen Warteschlangenmanager davon abhalten wollen, die Verbindung zu einem Cluster wiederherzustellen, müssen Sie entsprechende Sicherheitsmaßnahmen ergreifen.

Mit Ausnahme von **DISPLAY CLUSQMgr** agieren alle Clusterbefehle asynchron. Befehle zum Ändern von Objektattributen unter Einbeziehung von Clustering aktualisieren das Objekt und senden eine Anforderung an den Repositoryprozessor. Befehle für die Arbeit mit Clustern werden im Hinblick auf die Syntax überprüft und es wird eine Anforderung an den Repositoryprozessor gesendet.

Die an den Repositoryprozessor gesendeten Anforderungen werden gemeinsam mit den von anderen Clustermitgliedern empfangenen Clusteranforderungen asynchron verarbeitet. Die Verarbeitung nimmt

möglicherweise eine längere Zeit in Anspruch, wenn die Anforderungen an den gesamten Cluster gesendet werden müssen, um festzustellen, ob sie erfolgreich ausgeführt wurden oder nicht.

Zugehörige Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition 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.

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 (Cluster zurücksetzen)

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 Konzepte

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus

für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Tasks

Verwalten eines Warteschlangenmanagers

Zugehörige Verweise

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

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

DISPLAY CLUSQMGR

Verwenden Sie den Befehl **DISPLAY CLUSQMGR**, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

REFRESH CLUSTER

Geben Sie den Befehl **REFRESH CLUSTER** in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

SUSPEND QMGR

RESUME QMGR

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Geeignete Ziele werden vom Algorithmus für das Clusterauslastungsmanagement auf der Basis der Verfügbarkeit des Warteschlangenmanagers und der Warteschlange sowie einer Reihe von clusterauslastungsspezifischen Attributen ausgewählt, die Kanälen, Warteschlangen und Warteschlangenmanagern zugeordnet sind. Diese Attribute werden in den Unterthemen beschrieben.

Wenn die Konfiguration der spezifischen Attribute zur Clusterauslastung sich nicht wie erwartet verhält, untersuchen Sie die Einzelheiten zur Auswahl eines Warteschlangenmanagers durch den Algorithmus. Siehe „Algorithmus für das Clusterauslastungsmanagement“ auf Seite 151. Wenn die Ergebnisse dieses Algorithmus nicht Ihren Bedürfnissen entsprechen, können Sie ein Benutzerexitprogramm für Clusterauslastung schreiben und diesen Exit verwenden, um Nachrichten zu der Warteschlange Ihrer Wahl im Cluster weiterzuleiten. Weitere Informationen finden Sie unter Exits für Clusterauslastung schreiben und kompilieren.

<i>Tabelle 59. Zusammenfassung der für die Clusterauslastung spezifischen Attribute</i>	
Attributname	Beschreibung
Kanalattribute	

Tabelle 59. Zusammenfassung der für die Clusterauslastung spezifischen Attribute (Forts.)

Attributname	Beschreibung
CLWLPRTY	Gibt die Prioritätsreihenfolge für Kanäle für die Verteilung der Clusterauslastung an
CLWLRANK	Gibt die Rangordnung der Kanäle für die Verteilung der Clusterauslastung an
CLWLWGHT	Gibt die Wertigkeit an, die auf CLUSSDR -und CLUSRCVR -Kanäle für die Verteilung der Clusterauslastung angewendet wird.
NETPRTY	Gibt die Priorität für einen CLUSRCVR -Kanal an.
Warteschlangenattribute	
CLWLPRTY	Gibt die Priorität von lokalen, fernen oder Aliaswarteschlangen für die Verteilung der Clusterauslastung an
CLWLRANK	Gibt den Rang einer lokalen, fernen oder Aliaswarteschlange für die Verteilung der Clusterauslastung an
CLWLUseQ	Gibt an, ob eine lokale Instanz einer Warteschlange als Ziel gegenüber anderen Instanzen in einem Cluster bevorzugt wird.
Warteschlangenmanagerattribute	
CLWLMRUC	Legt die Anzahl der zuletzt ausgewählten Kanäle fest.
CLWLUseQ	Gibt an, ob eine lokale Instanz einer Warteschlange als Ziel vor anderen Instanzen der Warteschlange in einem Cluster bevorzugt wird.

Zugehörige Konzepte

 [Asynchronous behavior of CLUSTER commands on z/OS](#)

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

Zugehörige Verweise

[In Kanaldefinitionsbefehlen verfügbare Clusterattribute](#)

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

[In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute](#)

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

[In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute](#)

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

[DISPLAY CLUSQMGR](#)

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

[REFRESH CLUSTER](#)

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

[RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen](#)

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

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.

Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

Anmerkung: Die Kanalattribute für die Clusterauslastung müssen Sie auf den Ziel-Warteschlangenmanagern auf den Clusterempfängerkanälen angeben. Lastausgleichsattribute, die Sie auf den zugehörigen Clustersenderkanälen angeben, werden vermutlich ignoriert. Siehe [Clusterkanäle](#).

CLWLPRTY (Clusterauslastungspriorität)

Das Kanalattribut `CLWLPRTY` legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Kanalattribut `CLWLPRTY` legen Sie die Reihenfolge der Priorität der verfügbaren Clusterziele fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Nachrichten gehen an den Warteschlangenmanager mit dem Kanal mit der höchsten Priorität. Ist dieser Kanal nicht verfügbar, gehen die Nachrichten an den Warteschlangenmanager mit der nächsthöchsten Priorität. Warteschlangenmanager mit niedrigerer Priorität fungieren als Reserve.

Vor der Priorisierung der Kanäle prüft IBM MQ den Kanalstatus. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihr Status mehrmals, wobei einige Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherheit) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.
- Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherheitsziel gesendet werden, verwenden Sie `CLWLPRTY` nicht. Ziehen Sie die Verwendung separater Warteschlangen in Betracht, oder `CLWLRANK` mit einem manuellen Umschalten von der primären auf die Sicherheit.

CLWLRANK (Rangordnung der Clusterauslastung)

Das Kanalattribut `CLWLRANK` gibt die Ebene der Kanäle für die Verteilung der 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 Warte-

schlangenmanager gesendet, der durch einen Kanal mit dem höchsten Rang verbunden ist; in diesem Fall ist dies der Warteschlangenmanagerkanal mit dem Rang 2.

IBM MQ ruft den Rang von Kanälen noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Kanäle zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Würden Sie hingegen auch das Prioritätsattribut **CLWLPRTY** verwenden, würde IBM 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.

CLWLWGHT (Clusterauslastungsgewichtung)

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

Verwenden Sie CLWLWGHT, um mehr Nachrichten an Server mit einer größeren Verarbeitungskapazität zu senden. Je stärker ein Kanal gewichtet ist, desto mehr Nachrichten werden über diesen Kanal versendet.

Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Standardeinstellung des Parameters CLWLWGHT ist 50. Wird diese Einstellung für einen Kanal geändert, so wird der Lastausgleich abhängig von der Häufigkeit, die die einzelnen Kanäle für eine an eine Clusterwarteschlange gesendete Nachricht ausgewählt wurden. Weitere Informationen finden Sie unter „Algorithmus für das Clusterauslastungsmanagement“ auf Seite 151.

NETPRTY (Netzverbindungspriorität)

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Verwenden Sie das Attribut NETPRTY, um ein Netz als primäres Netz und ein weiteres Netz als Sicherungsnetz festzulegen. Bei einer Gruppe von Kanälen mit gleichem Rang wird durch Clustering der Pfad mit der höchsten Priorität ausgewählt, wenn mehrere Pfade zur Verfügung stehen.

Üblicherweise wird das Kanalattribut NETPRTY verwendet, um zwischen Netzen zu unterscheiden, die die gleichen Zielorte verbinden, aber unterschiedliche Aufwände verursachen oder unterschiedliche Geschwindigkeiten aufweisen.

Anmerkung: Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).

Zugehörige Konzepte

[Algorithmus für das Clusterauslastungsmanagement](#)

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

Zugehörige Verweise

[Clusterauslastungsausgleich-Warteschlangenattribute](#)

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

Clusterauslastungsausgleich-Warteschlangenmanager-Attribute

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

„In Kanaldefinitionsbefehlen verfügbare Clusterattribute“ auf Seite 132

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

Clusterauslastungsausgleich-Warteschlangenattribute

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Warteschlangenattribut **CLWLPRTY** legen Sie die Priorität einer Zielwarteschlange fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

IBM MQ ruft die Priorität von Warteschlangenmanagern erst nach der Überprüfung des Kanalstatus ab. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

Anmerkung:

Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihr Status mehrmals, wobei einige Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherung) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.

Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherungsziel gesendet werden, verwenden Sie nicht **CLWLPRTY**. Ziehen Sie die Verwendung separater Warteschlangen in Betracht oder **CLWLRANK** mit einem manuellen Umschalten von der Primärdatenbank zur Sicherung.

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.

CLWLRANK

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Das Warteschlangenattribut **CLWLRANK** sollten Sie verwenden, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen.

Stellen Sie sich zum Beispiel vor, Sie hätten zwei identisch konfigurierte Gateway-Warteschlangenmanager, durch die die Verfügbarkeit eines Gateways verbessert werden soll. Dazu hätten Sie an den Gateways Cluster-Aliaswarteschlangen für eine im Cluster definierte lokale Warteschlange konfiguriert. Sollte die lokale Warteschlange nicht zur Verfügung stehen, soll eine Nachricht an einem der Gateways zurückgehalten werden, bis die Warteschlange wieder zur Verfügung steht. Um die Cluster-Aliaswarteschlange an

einem der beiden Gateways zurückzustellen, muss die lokale Warteschlange mit einem höheren Rang definiert sein als die Aliaswarteschlangen am Gateway.

Hätte die lokale Warteschlange den gleichen Rang wie die Aliaswarteschlangen und stünde die lokale Warteschlange nicht zur Verfügung, würde die Nachricht zwischen den Gateways hin und her übertragen werden. Sobald der erste Gateway-Warteschlangenmanager feststellt, dass die lokale Warteschlange nicht verfügbar ist, würde er die Nachricht an das andere Gateway senden. Das andere Gateway würde erneut versuchen, die Nachricht der lokalen Warteschlange zuzustellen. Stünde die lokale Warteschlange nach wie vor nicht zur Verfügung, würde die Nachricht zurück zum ersten Gateway geleitet werden. Die Nachricht wird auf diese Weise zwischen den beiden Gateways hin und her gesendet, bis die lokale Warteschlange wieder zur Verfügung steht. Erhält die lokale Warteschlange aber einen höheren Rang, würde die Nachricht selbst bei Nichtverfügbarkeit der Warteschlange keinem Ziel mit niedrigerem Rang zugestellt werden.

IBM MQ ruft den Warteschlangenrang noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Warteschlangen zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

Wenn Sie hingegen das Prioritätsattribut 'CLWLPRTY' verwenden, wählt IBM MQ nur zwischen verfügbaren Zielen aus. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

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.

Beliebig

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

[Kanalattribute für Clusterauslastungsausgleich](#)

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

[Clusterauslastungsausgleich-Warteschlangenmanager-Attribute](#)

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

„In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute“ auf Seite 135

[Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.](#)

Clusterauslastungsausgleich-Warteschlangenmanager-Attribute

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

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.

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.

Beliebig

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

Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

Clusterauslastungsausgleich-Warteschlangenattribute

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

„In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute“ auf Seite 137

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

Algorithmus für das Clusterauslastungsmanagement

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

Immer, wenn ein Ziel ausgewählt werden muss, wird der Workload-Management-Algorithmus ausgeführt:

- Sie wird an dem Punkt verwendet, an dem eine Clusterwarteschlange unter Verwendung der Option MQOO_BIND_ON_OPEN geöffnet wird.
- Sie wird verwendet, wenn eine Nachricht in eine Clusterwarteschlange eingereicht wird, wenn sie mit MQOO_BIND_NOT_FIXED geöffnet wird.
- Sie wird bei jedem Start einer neuen Nachrichtengruppe verwendet, wenn MQOO_BIND_ON_GROUP verwendet wird, um eine Clusterwarteschlange zu öffnen.
- Er wird für Topic-Host-Routing bei jedem Veröffentlichen einer Nachricht für ein Clusterthema (Cluster-Topic) verwendet. Wenn es sich beim lokalen Warteschlangenmanager nicht um den Host für dieses Thema handelt, wird mit dem Algorithmus ein Host-Warteschlangenmanager ausgewählt, über den die Nachricht gesendet wird.

Im folgenden Abschnitt wird der Algorithmus für das Workload-Management beschrieben, der zur Bestimmung der Zielorte der in Clusterwarteschlangen befindlichen Nachrichten verwendet wird. Dessen

Regeln werden durch die Einstellungen der folgenden Attribute für Warteschlangen, Warteschlangenmanager und Kanäle beeinflusst:

Tabelle 60. Attribute für das Clusterauslastungsmanagement		
Warteschlangen	Warteschlangenmanager	Kanäle
<ul style="list-style-type: none"> • CLWLPRTY¹ • CLWLRANK¹ • CLWLUSEQ¹ • PUT / PUB 	<ul style="list-style-type: none"> • CLWLMRUC • CLWLUSEQ¹ 	<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 (also Kanäle) zu einem Warteschlangenmanager, der eine Warteschlange mit dem Namen `ObjectName` bereitstellt, die sich in einem der Cluster befindet, zu denen der Warteschlangenmanager gehört.

Der Algorithmus durchläuft die folgenden Regeln, um die Liste der möglichen Ziele zu reduzieren.

1. Ferne Instanzen von Warteschlangen oder Themen oder ferne CLUSRCVR-Kanäle, die keinen Cluster mit dem lokalen Warteschlangenmanager gemeinsam nutzen, werden ausgeschlossen.
2. Wenn eine Warteschlange oder ein Themename angegeben ist, werden ferne CLUSRCVR-Kanäle ausgeschlossen, die sich nicht in demselben Cluster wie die Warteschlange oder das Thema befinden.

Anmerkung: Alle verbleibenden Warteschlangen, Themen und Kanäle werden in dieser Phase dem Exit für Clusterauslastung zur Verfügung gestellt, wenn dieser konfiguriert ist.

3. Alle Kanäle zu Warteschlangenmanagern oder Warteschlangenmanager-Aliasnamen, die einen `CLWLRANK`-Wert aufweisen, der kleiner ist als der maximale Rang aller verbleibenden Kanäle oder Warteschlangenmanager-Aliasnamen, werden ausgeschlossen.
4. Alle Warteschlangen (keine Warteschlangenmanager-Aliasnamen) mit einem `CLWLRANK`-Wert, der kleiner als der maximale Rang aller verbleibenden Warteschlangen ist, werden ausgeschlossen.
5. Wenn mehr als eine Instanz einer Warteschlange, eines Themas oder eines Warteschlangenmanager-Aliasnamens übrig bleibt und für mindestens eine davon 'pub put' aktiviert ist, werden alle Instanzen ausgeschlossen, auf denen 'put' inaktiviert ist.

Anmerkung: Wenn ausschließlich Instanzen übrig bleiben, auf denen 'put' inaktiviert ist, können nur noch Abfrageoperationen erfolgreich ausgeführt werden, alle anderen Operationen schlagen mit `MQRC_CLUSTER_PUT_INHIBITED` fehl.

6. Wenn bei der Auswahl einer Warteschlange die verbleibende Gruppe der Warteschlangen die lokale Instanz der Warteschlange enthält, wird in der Regel die lokale Instanz verwendet. Die lokale Instanz der Warteschlange wird verwendet, wenn eine der folgenden Bedingungen zutrifft:
 - Das Attribut `CLWLUSEQ` der Warteschlange ist auf `LOCAL` gesetzt.
 - Die beiden folgenden Aussagen treffen zu:
 - Das Attribut `CLWLUSEQ` der Warteschlange ist auf `QMGR` gesetzt.
 - Das Attribut `CLWLUSEQ` des Warteschlangenmanagers ist auf `LOCAL` gesetzt.
 - Die Nachricht wurde nicht direkt von einer lokalen Anwendung eingereicht, sondern über einen Clusterkanal empfangen.

¹ Dieses Attribut gilt bei Auswahl einer Clusterwarteschlange, nicht bei Auswahl eines Themas.

- Auf lokal definierte Warteschlangen, die mit CLWLUSEQ(ANY) definiert sind oder diese Einstellung aus dem Warteschlangenmanager übernehmen, treffen folgende Punkte innerhalb der breiter gesteckten Gruppe von geltenden Bedingungen zu:
 - Die lokale Warteschlange wird auf der Basis der status der lokal definierten CLUSRCVR-Kanäle in demselben Cluster wie die Warteschlange ausgewählt. Dieser Status wird mit dem Status der CLUSSDR-Kanäle verglichen, die die Nachricht an remote definierte gleichnamige Warteschlangen leiten würden.

Beispiel: Es gibt einen CLUSRCVR im gleichen Cluster wie die Warteschlange. Dieser CLUSRCVR hat den Status STOPPING, während die anderen Warteschlangen desselben Namens im Cluster den Status RUNNING oder INACTIVE haben. In diesem Fall werden die fernen Kanäle ausgewählt und die lokalen CLUSSDR-Kanäle nicht verwendet.
 - Die lokale Warteschlange wird auf der Basis der number von CLUSRCVR-Kanälen ausgewählt (im Vergleich zu CLUSSDR-Kanälen mit demselben Status), die die Nachricht an fern definierte Warteschlangen mit demselben Namen senden würden.

Beispiel: Es gibt vier CLUSRCVR-Kanäle im selben Cluster wie die Warteschlange und einen CLUSSDR-Kanal. Alle Kanäle haben denselben Status, d. h. entweder INACTIVE oder RUNNING. Daher stehen fünf Kanäle zur Auswahl und zwei Instanzen der Warteschlange. Vier Fünftel (80 Prozent) der Nachrichten gehen an die lokale Warteschlange.
7. Wenn mehr als ein Warteschlangenmanager übrig bleibt und mindestens einer davon nicht vorübergehend gesperrt ist, werden alle anderen ausgeschlossen, die vorübergehend gesperrt sind.
 8. Wenn mehrere ferne Instanzen einer Warteschlange oder eines Themas verbleiben, werden alle Kanäle mit den Status INACTIVE oder RUNNING aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
 - MQCHS_INACTIVE
 - MQCHS_RUNNING
 9. Wenn keine ferne Instanz einer Warteschlange oder eines Themas verbleibt, werden alle rev="d1">Kanäle mit den Status BINDING, INITIALIZING, STARTING oder STOPPING aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
 - MQCHS_BINDING
 - MQCHS_INITIALIZING
 - MQCHS_STARTING
 - MQCHS_STOPPING
 10. Wenn keine Instanz einer Warteschlange oder eines Themas verbleibt, werden alle Kanäle eingeschlossen, für die ein erneuter Verbindungsversuch durchgeführt wird. Die Statuskonstante wird aufgeführt:
 - MQCHS_RETRYING
 11. Wenn keine ferne Instanz einer Warteschlange oder eines Themas verbleibt, werden alle Kanäle mit den Status REQUESTING, PAUSED oder STOPPED aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
 - MQCHS_REQUESTING
 - MQCHS_PAUSED
 - MQCHS_STOPPED
 - MQCHS_SWITCHING
 12. Wenn mehr als eine ferne Instanz einer Warteschlange oder eines Themas auf einem beliebigen Warteschlangenmanager verbleibt, werden Kanäle mit dem höchsten NETPRTY-Wert für die einzelnen Warteschlangenmanager ausgewählt.
 13. Alle verbleibenden Kanäle und Warteschlangenmanager-Aliasnamen mit Ausnahme der Kanäle und Aliasnamen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen. Verbleiben dabei War-

teschlangenmanager-Aliasnamen, so bleiben die Kanäle zu diesem Warteschlangenmanager eingeschlossen.

14. Bei Auswahl einer Warteschlange:

- Alle Warteschlangen mit Ausnahme der Warteschlangen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen, Kanäle bleiben aber erhalten.

15. Die verbleibenden Kanäle werden dann durch Entfernen der Kanäle mit den niedrigsten MQWDR.DestSeqNumber-Werten auf nicht mehr als die maximal zulässige Anzahl der zuletzt verwendeten Kanäle (CLWLMRUC) reduziert.

Anmerkung: Interne Steuernachrichten für das Cluster werden ggf. mit dem gleichen Algorithmus für die Clusterauslastung gesendet.

Nachdem die Liste der gültigen Ziele berechnet wurden, wird ein Lastausgleich für die Nachrichten mithilfe der folgenden Logik durchgeführt:

- Wenn mehrere ferne Instanzen eines Ziels verbleiben und für alle Kanäle zu diesem Ziel die Option CLWLWGHT auf die Standardeinstellung 50 gesetzt ist, wird der am längsten nicht mehr verwendete Kanal ausgewählt. Dies entspricht etwa einem Umlaufstil beim Durchführen des Lastausgleichs, wenn mehrere ferne Instanzen vorhanden sind.
- Wenn mehrere Instanzen eines Ziels verbleiben und für mindestens einen Kanal zu diesen Warteschlangen die Option CLWLWGHT auf eine andere Einstellung als die Standardeinstellung gesetzt ist (selbst wenn alle Kanäle einen übereinstimmenden, nicht standardmäßigen Wert haben), ist die Weiterleitung abhängig von den relativen Gewichtungen jedes Kanals und der Gesamtzahl, mit der jeder Kanal zuvor beim Senden von Nachrichten ausgewählt wurde.
- Beim Beobachten der Verteilung von Nachrichten für eine einzelne Clusterwarteschlange mit mehreren Instanzen führt dies scheinbar zu einer nicht ausgeglichenen Verteilung in einer Untergruppe der Warteschlangeninstanzen. Dies liegt daran, dass nicht nur die Nachrichtenübertragung für diese Warteschlange ausgeglichen wird, sondern die historische Verwendung jedes Clustersenderkanals aus diesem Warteschlangenmanager berücksichtigt wird. Wenn dieses Verhalten nicht erwünscht ist, führen Sie einen der folgenden Schritte aus:
 - Setzen Sie CLWLWGHT auf allen Clusterempfängerkanälen auf 50, wenn eine gleichmäßige Verteilung erforderlich ist.
 - oder, falls bestimmte Warteschlangeninstanzen anders als andere gewichtet werden müssen, definieren Sie diese Warteschlangen in einem dedizierten Cluster mit definierten dedizierten Clusterempfängerkanälen. Durch diese Aktion wird der Lastausgleich dieser Warteschlangen vom Lastausgleich anderer Warteschlangen im Cluster eingegrenzt.
- Die historischen Daten, mit denen der Lastausgleich für die Kanäle durchgeführt wird, werden zurückgesetzt, wenn ein beliebiges Attribut zur Clusterauslastung der verfügbaren Clusterempfängerkanäle geändert oder der Status eines Clusterempfängerkanals verfügbar wird. Durch Änderungen an den Auslastungsattributen manuell definierter Clustersenderkanäle werden die historischen Daten nicht zurückgesetzt.
- Wenn Sie erwägen, die Logik des Exits für die Clusterauslastung zu verwenden, wird der Kanal mit dem niedrigsten Wert für MQWDR.DestSeqFactor ausgewählt. Bei jeder Auswahl des Kanals wird dieser Wert um etwa 1000/CLWLWGHT erhöht. Falls mehrere Kanäle mit diesem Wert vorliegen, wird einer der Kanäle mit dem niedrigsten Wert für MQWDR.DestSeqNumber ausgewählt.

Die Verteilung der Benutzernachrichten erfolgt nicht immer exakt nach diesen Regeln, da auch durch die Clusterverwaltung Nachrichten über Kanäle übertragen werden. Dies führt zu einer zunächst ungleichmäßigen Verteilung der Benutzernachrichten, die sich erst nach einer gewissen Zeit stabilisiert. Aufgrund der Mischung aus Verwaltungs- und Benutzernachrichten sollten Sie sich nicht auf die genaue Verteilung von Nachrichten während des Lastausgleichs verlassen.

Zugehörige Verweise

Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

Clusterauslastungsausgleich-Warteschlangenattribute

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

Clusterauslastungsausgleich-Warteschlangenmanager-Attribute

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

For both REFRESH CLUSTER and RESET CLUSTER, message CSQM130I is sent to the command issuer indicating that a request has been sent. This message is followed by message CSQ9022I to indicate that the command has completed successfully, in that a request has been sent. It does not indicate that the cluster request has been completed successfully.

Any errors are reported to the z/OS console on the system where the channel initiator is running, they are not sent to the command issuer.

The asynchronous behavior is in contrast to CHANNEL commands. A message indicating that a channel command has been accepted is issued immediately. At some later time, when the command has been completed, a message indicating either normal or abnormal completion is sent to the command issuer.

Related concepts

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Related tasks

Checking that async commands for distributed networks have finished

Related reference

In Kanaldefinitionsbefehlen verfügbare Clusterattribute

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

In Warteschlangendefinitionsbefehlen verfügbare Clusterattribute

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

In WS-Manager-Definitionsbefehlen verfügbare Clusterattribute

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

DISPLAY CLUSQMGR

Verwenden Sie den Befehl DISPLAY CLUSQMGR, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

REFRESH CLUSTER

Geben Sie den Befehl REFRESH CLUSTER in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

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

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.

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.

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 IBM MQ-Empfangsprogramms), 'runmqchl' (Ausführung des IBM MQ-Kanals) und 'runmqchi' (Ausführung des IBM MQ-Kanalinitiators) können in der Befehlszeile eingegeben werden.

'amqcrsta' wird für TCP-Kanäle auf Systemen unter AIX and Linux mit 'inetd' aufgerufen, auf denen kein Empfangsprogramm gestartet wurde.

'amqcrs6a' wird bei Verwendung von LU6.2 als Transaktionsprogramm aufgerufen.

IBM i Jobs für übergreifende Kommunikation unter IBM i

Die folgenden Jobs sind der übergreifenden Kommunikation unter IBM i zugeordnet. Die Namen werden in der folgenden Tabelle aufgeführt.

Jobname	Beschreibung
AMQCLMAA	Empfangsprogramm ohne Threads
AMQCRSTA	Responder-Job ohne Threads
AMQRMPPA	Channel-Pool-Job
RUNMQCHI	Kanalinitiator
RUNMQCHL	Channel-Job
RUNMQLSR	Empfangsprogramm mit Threads

IBM i Kanalzustände unter IBM i

Der Kanalstatus wird in der Anzeige "Work with Channels" (Mit Kanälen arbeiten) eingeblendet.

Statusname	Bedeutung
STARTING	Der Kanal ist zur Verhandlung mit dem Ziel-MCA bereit.
BINDING	Einrichten einer Sitzung und erster Datenaustausch
REQUESTING	Requesterkanal initialisiert eine Verbindung
RUNNING	Übertragung wird ausgeführt oder bereit für Übertragung

Tabelle 63. Kanalzustände unter IBM i (Forts.)

Statusname	Bedeutung
PAUSED	Warten auf Nachrichtenwiederholungs- intervall
STOPPING	Aufbau von Wiederholung oder Stopp
RETRYING	Warten auf nächsten Versuch
STOPPED	Kanal aufgrund eines Fehlers oder der Ausgabe eines "end-channel"-Befehls gestoppt
INACTIVE	Kanal hat die Verarbeitung normal beendet oder Kanal wurde nie gestartet
*Keine	Kein Status (nur für Serververbindungskanäle)

ALW Beispiel: Nachrichtenkanal unter AIX, Linux, and Windows planen

Diese Informationen enthalten ein ausführliches Beispiel dafür, wie zwei Warteschlangenmanager miteinander verbunden werden, damit Nachrichten zwischen ihnen gesendet werden können.

Informationen zu diesem Vorgang

In allen Beispielen werden die MQSC-Beispiele so angezeigt, wie sie in einer Befehlsdatei dargestellt und auch in einer Befehlszeile eingegeben würden. Die zwei Vorgehensweisen sehen identisch aus, aber um einen Befehl in die Befehlszeile einzugeben, müssen Sie zuerst `runmqsc` für den Standardwarteschlangenmanager bzw. `runmqsc qmname` eingeben, wobei *qmname* der Name des betreffenden Warteschlangenmanagers ist. Geben Sie dann eine beliebige Anzahl Befehle ein, wie in den Beispielen gezeigt.

Eine alternative Möglichkeit besteht darin, eine Datei mit diesen Befehlen zu erstellen. Jegliche Fehler in den Befehlen können so leicht korrigiert werden. Wenn Sie Ihre Datei "mqsc.in" aufgerufen haben, um sie für den Warteschlangenmanager QMNAME auszuführen, geben Sie Folgendes ein:

```
runmqsc QMNAME < mqsc.in > mqsc.out
```

Durch folgende Eingabe können Sie die Befehle in Ihrer Datei verifizieren, bevor Sie sie ausführen:

```
runmqsc -v QMNAME < mqsc.in > mqsc.out
```

Die Zeilenlänge für Ihre Befehle sollte 72 Zeichen nicht übersteigen, um eine einwandfreie Portierbarkeit zu ermöglichen. Geben Sie ein Verkettungszeichen ein, wenn sich der Befehl über mehr als eine Zeile erstreckt. Verwenden Sie auf Windows-Systemen die Tastenkombination Strg + Z, um die Eingabe in die Befehlszeile zu beenden. Verwenden Sie auf Systemen mit AIX and Linux die Tastenkombination Strg + D. Alternativ können Sie den Befehl **end** eingeben.

Abbildung 7 auf Seite 158 zeigt das Beispielszenario.

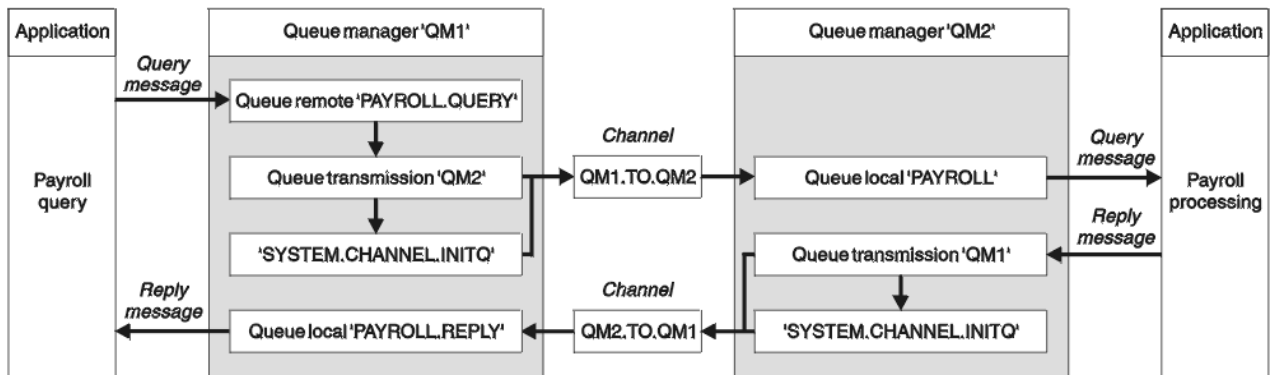


Abbildung 7. Nachrichtenkanalbeispiel für Systeme mit AIX, Linux, and Windows

Im Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragenachrichten an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen. Die Lohnbuchhaltungsabfragenachrichten werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

In den Beispielformatdefinitionen für TCP/IP hat QM1 die Hostadresse 192.0.2.0 und ist am Anschluss 1411 empfangsbereit, und QM2 hat die Hostadresse 192.0.2.1 und ist empfangsbereit am Anschluss 1412. Voraussetzung bei diesem Beispiel ist, dass diese Einstellungen bereits in Ihrem System vorgenommen wurden und die Warteschlangenmanager zur Verfügung stehen.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 7 auf Seite 158 zeigt ein entsprechendes Anordnungsdiagramm.

Vorgehensweise

Unter

- „[Nachrichtenkanalbeispiel für AIX, Linux, and Windows einrichten](#)“ auf Seite 159 für Details zum Einrichten der Nachrichtenkanäle
- „[Beispiel für AIX, Linux, and Windows ausführen und erweitern](#)“ auf Seite 160 enthält Vorschläge, wie Sie andere Produkte verwenden können, z. B. CICS, und wie Sie weitere Anwendungen und Benutzersitzungen verbinden können.

ALW

Nachrichtenkanalbeispiel für AIX, Linux, and Windows einrichten

Diese Objektdefinitionen ermöglichen Anwendungen, die mit Warteschlangenmanager QM1 verbunden sind, Anforderungsnachrichten an eine Warteschlange mit dem Namen PAYROLL auf QM2 zu senden und Antworten in einer Warteschlange mit dem Namen PAYROLL.REPLY auf QM1 und ermöglichen es Anwendungen, die mit dem Warteschlangenmanager QM2 verbunden sind, Anforderungsnachrichten aus einer lokalen Warteschlange mit dem Namen PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in eine Warteschlange mit dem Namen PAYROLL.REPLY auf Warteschlangenmanager QM1.

Informationen zu diesem Vorgang

Alle Objektdefinitionen wurden mit den Attributen DESCR und REPLACE bereitgestellt. Die anderen angegebenen Attribute sind das Minimum, das erforderlich ist, damit die Beispiele funktionieren. Die nicht bereitgestellten Attribute übernehmen die Standardwerte für die Warteschlangenmanager QM1 und QM2.

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.

Prozedur

- Führen Sie auf Warteschlangenmanager QM1 folgende Befehle aus:
 - a) Definieren Sie die Definition der fernen Warteschlangen:

```
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.

- b) Definieren Sie die Übertragungswarteschlangendefinition:

```
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.

- c) Definieren Sie die Senderkanaldefinition:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
```

```
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('192.0.2.1(1412)')
```

d) Richten Sie die Empfängerkanaldefinition ein:

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

e) Definieren Sie die Definition für 'reply-to_queue':

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QM2')
```

Die Empfangswarteschlange für Antworten ist als PUT(ENABLED) definiert. Dies stellt sicher, dass Antwortnachrichten in die Warteschlange gestellt werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereiht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

- Führen Sie auf Warteschlangenmanager QM2 folgende Befehle aus:

a) Definieren Sie die 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.

b) Definieren Sie die Übertragungswarteschlangendefinition:

```
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.

c) Definieren Sie die Senderkanaldefinition:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('192.0.2.0(1411)')
```

d) Richten Sie die Empfängerkanaldefinition ein:

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

ALW

Beispiel für AIX, Linux, and Windows ausführen und erweitern

Informationen zum Starten des Kanalinitiators und des Empfangsprogramms sowie Vorschläge zum Erweitern dieses Szenarios.

Informationen zu diesem Vorgang

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.

Sie können das Beispiel auch erweitern.

Vorgehensweise

1. Starten Sie den Kanalinitiator und das Empfangsprogramm.

Siehe [Kommunikation für Windows einrichten](#) und [Kommunikation auf AIX and Linux -Systemen einrichten](#).

2. Sie können dieses Beispiel wie folgt 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 Verbindungsverchlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden Sie Warteschlangenmanager-Aliasnamen und Aliasnamen für die Warteschlange für Antwortnachrichten, um weitere Kenntnisse darüber zu erlangen, wie diese in der Organisation Ihres Warteschlangenmanagernetzes verwendet werden können.

IBM i

Beispiel: Nachrichtenkanal unter IBM i planen

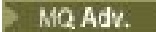

Ein ausführliches Beispiel dafür, wie zwei IBM i -Warteschlangenmanager miteinander verbunden werden, damit Nachrichten zwischen ihnen gesendet werden. Im Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragenachrichten an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen.

Informationen zu diesem Vorgang

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht.

In diesem Beispiel wird SYSTEM.CHANNEL.INITQ als Initialisierungswarteschlange verwendet. Diese Warteschlange ist in IBM MQ bereits definiert. Sie können eine andere Initialisierungswarteschlange verwenden, müssen diese jedoch selbst definieren, mit dem Befehl STRMQMCHLI eine neue Instanz des Kanalinitiators starten und ihr den Namen Ihrer Initialisierungswarteschlange bereitstellen. Weitere Informationen zum Auslösen von Kanälen finden Sie im Abschnitt [Kanäle auslösen](#).

Anmerkung:   Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Weitere Informationen finden Sie unter [Aspera gateway -Verbindung unter Linux oder Windows definieren](#).

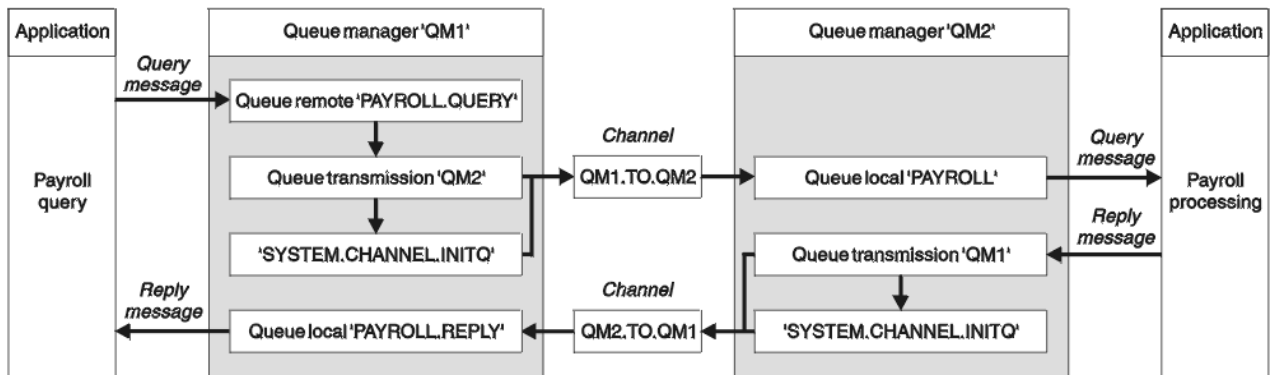


Abbildung 8. Nachrichtenkanalbeispiel für IBM MQ for IBM i

Die Lohnbuchhaltungsabfragenachrichten werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

Bei beiden Warteschlangenmanagern wird vorausgesetzt, dass sie unter IBM i ausgeführt werden. In den Beispielformulierungen hat QM1 die Hostadresse 192.0.2.0 und ist an Port 1411 empfangsbereit. QM2 hat die Hostadresse 192.0.2.1 und ist an Port 1412 empfangsbereit. In diesem Beispiel wird davon ausgegangen, dass diese Warteschlangenmanager bereits auf dem IBM i-System definiert wurden und die Warteschlangenmanager zur Verfügung stehen.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 8 auf Seite 162 zeigt ein entsprechendes Anordnungsdiagramm.

Vorgehensweise

Unter

- „Nachrichtenkanalagenten unter IBM i einrichten“ auf Seite 163 für Details zum Einrichten der Nachrichtenkanäle

- „Beispiel für IBM i ausführen und erweitern“ auf Seite 165 für Vorschläge, wie Sie weitere Anwendungen und Benutzerexits verbinden können.

IBM i Nachrichtenkanalagenten unter IBM i einrichten

Die folgenden Objektdefinitionen ermöglichen Anwendungen, die mit dem Warteschlangenmanager QM1 verbunden sind, Anforderungsnachrichten an eine Warteschlange namens PAYROLL auf QM2 zu senden, um Antworten in einer Warteschlange namens PAYROLL.REPLY auf QM1 zu ermöglichen. Anwendungen, die mit dem Warteschlangenmanager QM2 verbunden sind, das Abrufen von Anforderungsnachrichten aus einer lokalen Warteschlange mit dem Namen PAYROLL und das Einreihen von Antworten auf diese Anforderungsnachrichten in eine Warteschlange mit dem Namen PAYROLL.REPLY auf dem Warteschlangenmanager QM1.

Informationen zu diesem Vorgang

Sämtliche Objektdefinitionen werden in den TEXT-Attributen genannt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die nicht bereitgestellten Attribute übernehmen die Standardwerte für die Warteschlangenmanager QM1 und QM2.

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 dem Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt der Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Prozedur

- Führen Sie auf dem Warteschlangenmanager QM1 folgende Befehle aus:
 - a) Richten Sie die Definition der fernen Warteschlange ein, indem Sie den Befehl CRTMQMQ mit den folgenden Attributen verwenden:

QNAME	'PAYROLL.QUERY'
QTYPE	*RMT
TEXT	'Ferne Warteschlange für QM2'
PUTENBL	*YES
TMQNAME	'QM2' (Standard: Name des fernen Warteschlangenmanagers)
RMTQNAME	'PAYROLL'
RMTMQMNAME	'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 den Warteschlangenmanager QM2 gesendet werden können.

- b) Richten Sie die Definition der Übertragungswarteschlange ein, indem Sie den Befehl CRTMQMQ mit den folgenden Attributen verwenden:

QNAME	QM2
QTYPE	*LCL
TEXT	'Übertragungswarteschlange zu QM2'
NUTZUNG	*TMQ
PUTENBL	*YES

GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM1.TO.QM2

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

- c) Richten Sie die Senderkanaldefinition mit folgendem Befehl CRTMQCHL ein:

CHLNAME	QM1.TO.QM2
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Senderkanal zu QM2'
TMQNAME	QM2
CONNNAME	'192.0.2.1(1412)'

- d) Richten Sie die Empfängerkanaldefinition mit dem Befehl CRTMQCHL und den folgenden Attributen ein:

CHLNAME	QM2.TO.QM1
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Empfängerkanal von QM2'

- e) Richten Sie die Definition der Empfangswarteschlange für Antworten mit dem Befehl CRTMQMQ und den folgenden Attributen ein:

QNAME	PAYROLL.REPLY
QTYPE	*LCL
TEXT	'Antwortwarteschlange für Antworten zu Abfragenachrichten, die an QM2 gesendet wurden'
PUTENBL	*YES
GETENBL	*YES

Die Empfangswarteschlange für Antworten ist als PUT(ENABLED) definiert. Diese Definition stellt sicher, dass Antwortnachrichten in die Warteschlange eingereiht werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereiht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

- Führen Sie auf Warteschlangenmanager QM2 folgende Befehle aus:

- a) Richten Sie die Definition der lokalen Warteschlange ein, indem Sie den Befehl CRTMQMQ mit den folgenden Attributen verwenden:

QNAME	PAYROLL
QTYPE	*LCL

TEXT	'Lokale Warteschlange für Lohnbuchhaltungsdetails von QM1'
PUTENBL	*YES
GETENBL	*YES

Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

- b) Richten Sie die Definition der Übertragungswarteschlange ein, indem Sie den Befehl CRTMQMQ mit den folgenden Attributen verwenden:

QNAME	QM1
QTYPE	*LCL
TEXT	'Übertragungswarteschlange zu QM1'
NUTZUNG	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM2.TO.QM1

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal.

- c) Richten Sie die Definition des Senderkanals mithilfe des Befehls CRTMQMCHL mit den folgenden Attributen ein:

CHLNAME	QM2.TO.QM1
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Senderkanal zu QM1'
TMQNAME	QM1
CONNAME	'192.0.2.0(1411)'

- d) Richten Sie die Empfängerkanaldefinition mithilfe des Befehls CRTMQMCHL mit den folgenden Attributen ein:

CHLNAME	QM1.TO.QM2
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Empfängerkanal von QM1'

Informationen zu diesem Vorgang

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.

Die Anwendungen können sich daraufhin gegenseitig Nachrichten zusenden. Das Starten der Kanäle wird durch die erste Nachricht ausgelöst, die in jeder Übertragungswarteschlange eintrifft, daher muss der Befehl STRMQMCHL nicht ausgegeben werden.

Sie können das Beispiel auch erweitern.

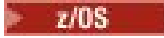
Vorgehensweise

1. Starten Sie den Kanalinitiator und das Empfangsprogramm.

Weitere Informationen zum Starten eines Kanalinitiators und eines Listeners finden Sie unter [Kanäle unter IBM überwachen und steuern](#).

2. Sie können dieses Beispiel wie folgt erweitern:

- Fügen Sie weitere Warteschlangen- und Kanaldefinitionen hinzu, damit andere Anwendungen Nachrichten zwischen den beiden Warteschlangenmanagern senden können.
- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverchlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden Sie Warteschlangenmanager-Aliasnamen und Aliasnamen für die Warteschlange für Antwortnachrichten, um weitere Kenntnisse darüber zu erlangen, wie diese Objekte in der Organisation Ihres Warteschlangenmanagernetzes verwendet werden können.

 Eine Version dieses Beispiels, das MQSC-Befehle verwendet, finden Sie unter [„Example: planning a message channel on z/OS“](#) auf Seite 166.

 z/OS



Example: planning a message channel on z/OS

How to connect z/OS or MVS queue managers together so that messages can be sent between them. This example involves a payroll query application connected to queue manager QM1 that sends payroll query messages to a payroll processing application running on queue manager QM2. The payroll query application needs the replies to its queries sent back to QM1.

About this task

The example illustrates the preparations needed to allow an application using queue manager QM1 to put messages on a queue at queue manager QM2. An application running on QM2 can retrieve these messages, and send responses to a reply queue on QM1.

The example illustrates the use of both TCP/IP and LU 6.2 connections. The example assumes that channels are to be triggered to start when the first message arrives on the transmission queue they are servicing.

Note:   Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf eine IBM Aspera faspio Gatewayverweisen, die einen schnellen TCP/IP-Tunnel bereitstellt, der den Netzdurchsatz erheblich erhöhen kann. Weitere Informationen finden Sie unter [Aspera gateway -Verbindung unter Linux oder Windows definieren](#).

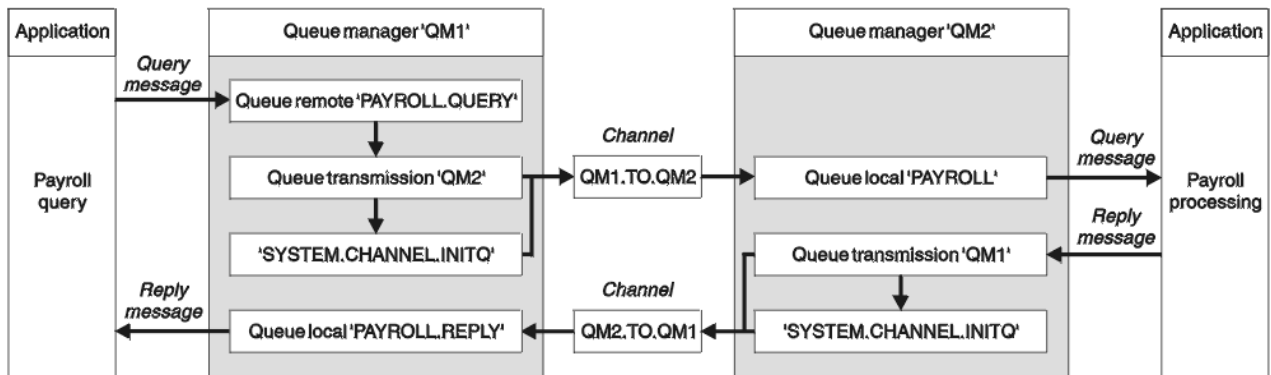


Figure 9. The first example for IBM MQ for z/OS

The payroll query messages are sent from QM1 to QM2 on a sender-receiver channel called QM1.TO.QM2, and the reply messages are sent back from QM2 to QM1 on another sender-receiver channel called QM2.TO.QM1. Both of these channels are triggered to start as soon as they have a message to send to the other queue manager.

The payroll query application puts a query message to the remote queue "PAYROLL.QUERY" defined on QM1. This remote queue definition resolves to the local queue "PAYROLL" on QM2. In addition, the payroll query application specifies that the reply to the query is sent to the local queue "PAYROLL.REPLY" on QM1. The payroll processing application gets messages from the local queue "PAYROLL" on QM2, and sends the replies to wherever they are required; in this case, local queue "PAYROLL.REPLY" on QM1.

Both queue managers are assumed to be running on z/OS. In the example definitions for TCP/IP, QM1 has a host address of 192.0.2.0 and is listening on port 1411, and QM2 has a host address of 192.0.2.1 and is listening on port 1412. In the definitions for LU 6.2, QM1 is listening on a symbolic luname called LUNAME1 and QM2 is listening on a symbolic luname called LUNAME2. The example assumes that these lunames are already defined on your z/OS system and available for use. To define them, see ["Example: setting up IBM MQ cross-platform communication on z/OS"](#) on page 46.

The object definitions that need to be created on QM1 are:

- Remote queue definition, PAYROLL.QUERY
- Transmission queue definition, QM2 (default=remote queue manager name)
- Sender channel definition, QM1.TO.QM2
- Receiver channel definition, QM2.TO.QM1
- Reply-to queue definition, PAYROLL.REPLY

The object definitions that need to be created on QM2 are:

- Local queue definition, PAYROLL
- Transmission queue definition, QM1 (default=remote queue manager name)
- Sender channel definition, QM2.TO.QM1
- Receiver channel definition, QM1.TO.QM2

The example assumes that all the SYSTEM.COMMAND.* and SYSTEM.CHANNEL.* queues required to run DQM have been defined as shown in the supplied sample definitions, **CSQ4INSG** and **CSQ4INSX**.

The connection details are supplied in the CONNAME attribute of the sender channel definitions.

You can see a diagram of the arrangement in [Figure 9 on page 167](#).

Procedure

See:

- ["Setting up the message channel agent on z/OS"](#) on page 168 for details on setting up the message channels

- “[Running and expanding the example for z/OS](#)” on page 170 for suggestions on how you can connect more applications and user exits.

z/OS

Setting up the message channel agent on z/OS

The following object definitions allow applications connected to queue manager QM1 to send request messages to a queue called PAYROLL on QM2 and also allows applications to receive replies on a queue called PAYROLL.REPLY on QM1. The definitions also allow applications connected to queue manager QM2 to retrieve request messages from a local queue called PAYROLL, and to put replies to these request messages to a queue called PAYROLL.REPLY on queue manager QM1.

About this task

All the object definitions have been provided with the DESCR and REPLACE attributes and are the minimum required to make the example work. The attributes that are not supplied take the default values for queue managers QM1 and QM2.

You do not need to provide a remote queue definition to enable the replies to be returned to QM1. The message descriptor of the message retrieved from local queue PAYROLL contains both the reply-to queue and the reply-to queue manager names. Therefore, as long as QM2 can resolve the reply-to queue manager name to that of a transmission queue on queue manager QM2, the reply message can be sent. In this example, the reply-to queue manager name is QM1 and so queue manager QM2 requires a transmission queue of the same name.

Procedure

- Run the following commands on queue manager QM1:
 - a) Setup the remote queue definition:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

Note: The remote queue definition is not a physical queue, but a means of directing messages to the transmission queue, QM2, so that they can be sent to queue manager QM2.

- b) Setup the transmission queue definition:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM1.TO.QM2) INITQ(SYSTEM.CHANNEL.INITQ)
```

When the first message is put on this transmission queue, a trigger message is sent to the initiation queue, SYSTEM.CHANNEL.INITQ. The channel initiator gets the message from the initiation queue and starts the channel identified in the trigger data. The channel initiator can only get trigger messages from the SYSTEM.CHANNEL.INITQ queue, so do not use any other queue as the initiation queue.

- c) Setup the sender channel definition:

For a TCP/IP connection:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNNAME('192.0.2.1(1412)')
```

For an LU 6.2 connection:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNNAME('LUNAME2')
```


d) Setup the receiver channel definition:

For a TCP/IP connection:

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

For an LU 6.2 connection:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(LU62) +  
REPLACE DESCR('Receiver channel from QM2')
```

e) Setup the reply-to queue definition:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to query messages sent to QM2')
```

The reply-to queue is defined as PUT(ENABLED) which ensures that reply messages can be put to the queue. If the replies cannot be put to the reply-to queue, they are sent to the dead-letter queue on QM1 or, if this queue is not available, remain on transmission queue QM1 on queue manager QM2. The queue has been defined as GET(ENABLED) to allow the reply messages to be retrieved.

- Run the following commands on queue manager QM2:

a) Setup the local queue definition:

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Local queue for QM1 payroll details')
```

This queue is defined as PUT(ENABLED) and GET(ENABLED) for the same reason as the reply-to queue definition on queue manager QM1.

b) Setup the transmission queue definition:

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
TRIGDATA(QM2.TO.QM1) INITQ(SYSTEM.CHANNEL.INITQ)
```

When the first message is put on this transmission queue, a trigger message is sent to the initiation queue, SYSTEM.CHANNEL.INITQ. The channel initiator gets the message from the initiation queue and starts the channel identified in the trigger data. The channel initiator can only get trigger messages from SYSTEM.CHANNEL.INITQ so do not use any other queue as the initiation queue.

c) Setup the sender channel definition:

For a TCP/IP connection:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNNAME('192.0.2.0(1411)')
```

For an LU 6.2 connection:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(LU62) +  
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNNAME('LUNAME1')
```

d) Setup the receiver channel definition:

For a TCP/IP connection:

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

For an LU 6.2 connection:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(LU62) +  
REPLACE DESCR('Receiver channel from QM1')
```

Running and expanding the example for z/OS

Information about starting the channel initiator and listener and suggestions for expanding on this example.

About this task

Once these definitions have been created, you need to:

- Start the channel initiator on each queue manager.
- Start the listener for each queue manager.

The applications can then send messages to each other. Because the channels are triggered to start by the arrival of the first message on each transmission queue, you do not need to issue the START CHANNEL MQSC command.

You can also expand the example.

Procedure

1. Start the channel initiator and listener.

See [Starting a channel initiator](#), and [Starting a channel listener](#) for details on how to start a channel initiator and listener.

2. You can expand this example by:

- Adding more queues, and channel definitions to allow other applications to send messages between the two queue managers.
- Adding user exit programs on the channels to allow for link encryption, security checking, or additional message processing.
- Using queue manager aliases and reply-to queue aliases to understand more about how these aliases can be used in the organization of your queue manager network.

Example: planning a message channel for z/OS using queue sharing groups

This example illustrates the preparations needed to allow an application using queue manager QM3 to put a message on a queue in a queue sharing group that has queue members QM4 and QM5, and also shows the IBM MQ commands (MQSC) that you can use in IBM MQ for z/OS for distributed queuing with queue sharing groups.

About this task

Ensure you are familiar with the example in [“Example: planning a message channel on z/OS” on page 166](#) before trying this one. This example expands the payroll query scenario of that example, to show how to add higher availability of query processing by adding more serving applications to serve a shared queue.

The payroll query application is now connected to queue manager QM3 and puts a query to the remote queue 'PAYROLL QUERY' defined on QM3. This remote queue definition resolves to the shared queue

'PAYROLL' hosted by the queue managers in the queue sharing group QSG1. The payroll processing application now has two instances running, one connected to QM4 and one connected to QM5.

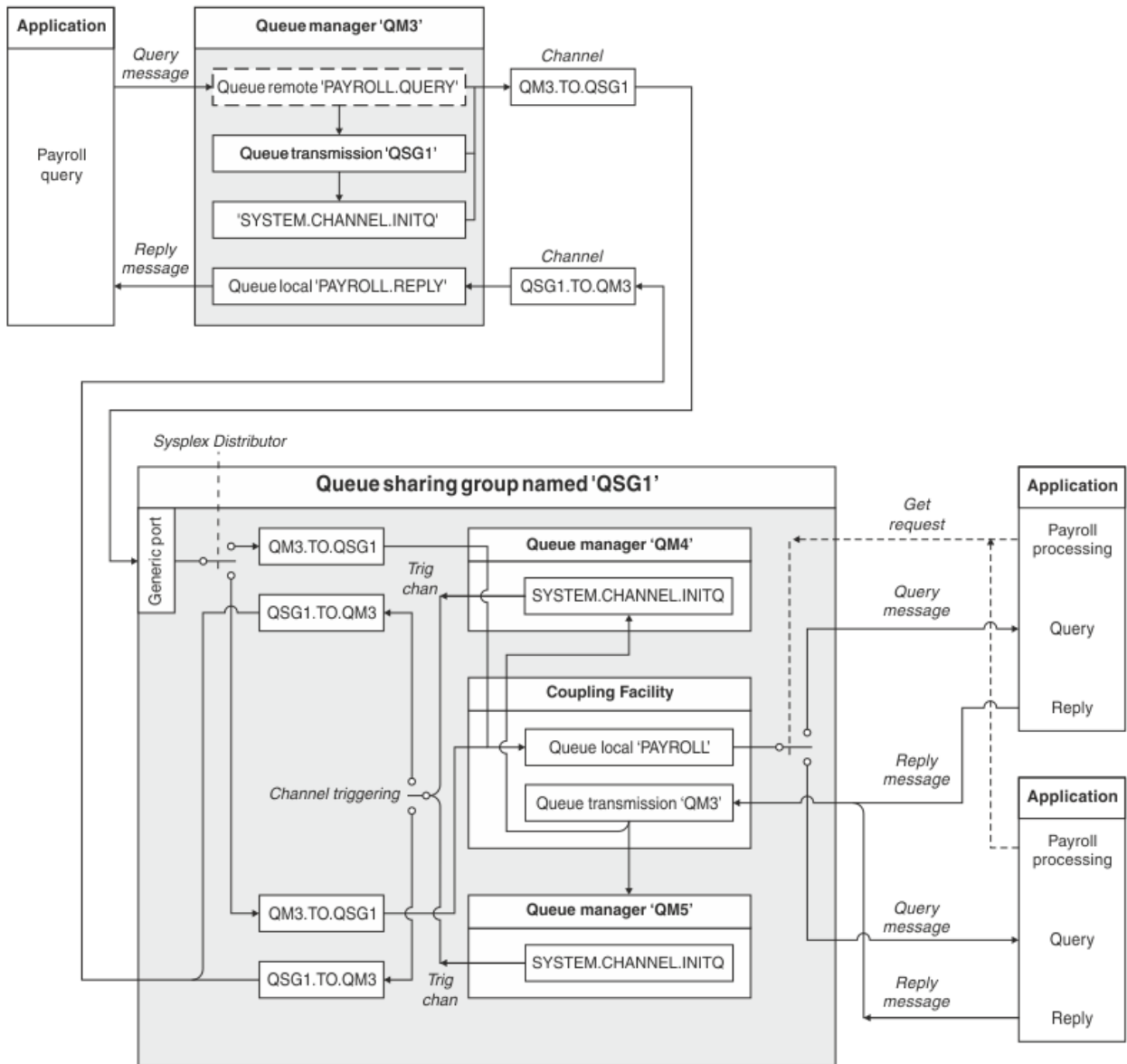


Figure 10. Message channel planning example for IBM MQ for z/OS using queue sharing groups

All three queue managers are assumed to be running on z/OS. In the example definitions for TCP/IP, QM4 has a VIPA address of MVSIP01 and QM5 has a VIPA address of MVSIP02. Both queue managers are listening on port 1414. The generic address that Sysplex Distributor provides for this group is QSG1.MVSIP. QM3 has a host address of 192.0.2.0 and is listening on port 1411.

In the example definitions for LU6.2, QM3 is listening on a symbolic luname called LUNAME1. The name of the generic resource defined for VTAM for the lunames listened on by QM4 and QM5 is LUQSG1. The example assumes that they are already defined on your z/OS system and are available for use. To define them see [“Establishing an LU 6.2 connection into a queue sharing group”](#) on page 51.

In this example QSG1 is the name of a queue sharing group, and queue managers QM4 and QM5 are the names of members of the group.

Procedure

See:

- “Setting up the queue sharing group definitions and a queue manager QM3 not in the queue sharing group” on page 172 for details on setting up the definitions.
- “Running the queue sharing group example for z/OS” on page 173 for details on starting the channel initiators and listeners for each queue manager.

z/OS Setting up the queue sharing group definitions and a queue manager QM3 not in the queue sharing group

Producing the following object definitions for one member of the queue sharing group makes them available to all the other members. QM3 is not a member of the queue sharing group.

About this task

Queue managers QM4 and QM5 are members of the queue sharing group. The definitions produced for QM4 are also available for QM5.

The coupling facility list structure is assumed to be called 'APPLICATION1'. If it is not called 'APPLICATION1', you must use your own coupling facility list structure name for the example.

As QM3 is not a member of the queue sharing group you need the object definitions for that queue manager to allow it to put messages to a queue in the queue sharing group.

Procedure

- Setup the shared objects for the queue sharing group definition:
 - a) Use the following commands to setup the shared object definitions that are stored in Db2, and their associated messages that are stored within the coupling facility.

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) REPLACE PUT(ENABLED) GET(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Shared queue for payroll details')

DEFINE QLOCAL(QM3) QSGDISP(SHARED) REPLACE USAGE(XMITQ) PUT(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Transmission queue to QM3') TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QSG1.TO.QM3) GET(ENABLED) INITQ(SYSTEM.CHANNEL.INITQ)
```

- Use the following commands to setup the group object definitions that are stored in Db2[®]. Each queue manager in the queue sharing group creates a local copy of the defined object.
 - a) Setup the sender channel:

Sender channel definition for a TCP/IP connection:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNAME('192.0.2.0(1411)')
```

Sender channel definition for an LU 6.2 connection:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNAME('LUNAME1')
```

- b) Setup the receiver channel:

Receiver channel definition for a TCP/IP connection:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

Receiver channel definition for an LU 6.2 connection:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

- Setup queue manager QM3 object definitions.

a) Setup the CONNAME

The CONNAME for this channel is the generic address of the queue sharing group, which varies according to transport type.

For a TCP/IP connection:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('QSG1.MVSIP(1414)')
```

For an LU 6.2 connection:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('LUQSG1') TPNAME('MQSERIES') MODENAME('#INTER')
```

b) Setup the other definitions.

These definitions are required for the same purposes as those used in the sub topics for [“Example: planning a message channel on z/OS”](#) on page 166.

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QSG1') REPLACE +
PUT(ENABLED) XMITQ(QSG1) RNAME(APPL) RQMNAME(QSG1)
```

```
DEFINE QLOCAL(QSG1) DESCR('Transmission queue to QSG1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM3.TO.QSG1) INITQ(SYSTEM.CHANNEL.INITQ)
```

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QSG1')
```

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QSG1')
```

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QSG1')
```

z/OS

Running the queue sharing group example for z/OS

Information about starting the channel initiators and listeners.

About this task

After you have created the required objects, you need to:

- Start the channel initiator for all three queue managers.
- Start the listeners for both queue managers.

Procedure

1. Start the channel initiators.

See [Starting a channel initiator](#) for details on how to start a channel initiator.

2. Start the listeners.

See [Starting a channel listener](#) for details on how to start a listener.

For a TCP/IP connection, each member of the group must have a group listener started that is listening on port 1414.

```
STA LSTR PORT(1414) IPADDR(MVSIP01) INDISP(GROUP)
```

The previous entry starts the listener on QM4, for example.

For an LU6.2 connection, each member of the group must have a group listener started that is listening on a symbolic luname. This luname must correspond to the generic resource LUQSG1.

```
STA LSTR PORT(1411)
```

The previous entry starts the listener on QM3.

Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek

Sie können ein Alias definieren, das auf eine MQ-Bibliothek in Ihrer Jobsteuersprache verweist, statt den Namen der MQ-Bibliothek direkt anzugeben. Wenn sich später der Name der MQ-Bibliothek ändert, brauchen Sie lediglich das Alias zu löschen und erneut zu definieren.

Beispiel

Im folgenden Beispiel wird ein Alias MQM.SCSQANLE definiert, das auf die MQ-Bibliothek MQM.V600.SCSQANLE verweist:

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SCSQANLE)
DEFINE ALIAS (NAME(MQM.SCSQANLE) RELATE(MQM.V600.SCSQANLE))
/*
```

Anschließend können Sie in Ihrer Jobsteuersprache über das Alias MQM.SCSQANLE auf die Bibliothek MQM.V600.SCSQANLE verweisen.

Anmerkung: Der Bibliotheksname und der Aliasname müssen sich in demselben Katalog befinden; verwenden Sie deshalb für beide dasselbe übergeordnete Qualifikationsmerkmal (im vorliegenden Beispiel MQM).

Managed File Transfer -Konfigurationsreferenz

Die Referenzinformationen unterstützen Sie bei der Konfiguration von Managed File Transfer.

Verwendung von Umgebungsvariablen in MFT-Eigenschaften

Es ist möglich, dass Umgebungsvariablen in Managed File Transfer -Eigenschaften verwendet werden, die Datei- oder Verzeichnispositionen darstellen. Dadurch können die Positionen der Dateien oder Verzeichnisse, die bei der Ausführung von Teilen des Produkts verwendet werden, abhängig von der aktuellen Umgebung variieren (z. B. der Benutzer, der einen Befehl ausführt).

In den folgenden Eigenschaften können Datei- oder Verzeichnispositionen auch in Form von Umgebungsvariablen angegeben werden:

- agentQMgrAuthenticationCredentialsFile
- agentSslKeyStore
- agentSslKeyStoreCredentialsFile
- agentSslTrustStore
- agentSslTrustStoreCredentialsFile
- cdNodeKeystoreCredentialsFile
- cdNodeTruststoreCredentialsFile
- cdTmpDir

- cdNodeKeystore
- cdNodeTruststore
- commandPath
- connectionQMgrAuthenticationCredentialsFile
- connectionSslKeyStore
- connectionSslKeyStoreCredentialsFile
- connectionSslTrustStore
- connectionSslTrustStoreCredentialsFile
- coordinationSslKeyStore
- coordinationSslKeyStoreCredentialsFile
- coordinationQMgrAuthenticationCredentialsFile
- coordinationSslTrustStore
- coordinationSslTrustStoreCredentialsFile
- exitClassPath
- exitNativeLibraryPath
- javaCoreTriggerFile
- loggerQMgrAuthenticationCredentialsFile
- sandboxRoot
- transferRoot
- wmqfte.database.credentials.file

Beispiel für Windows

Windows Im nachfolgenden Beispiel für ein Windows-System verwendet der Benutzer `fteuser` die Umgebungsvariable `USERPROFILE`:

```
wmqfte.database.credentials.file=%USERPROFILE%\logger\mqmftcredentials.xml
```

Dies wird in folgenden Dateipfad aufgelöst:

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

Beispiel für AIX and Linux

Linux **AIX** Im nachfolgenden Beispiel für ein UNIX-System verwendet der Benutzer `fteuser` die Umgebungsvariable `HOME`:

```
transferRoot=$HOME/fte/
```

Dies wird in folgenden Dateipfad aufgelöst:

```
/home/fteuser/fte/
```

Zugehörige Verweise

„Die MFT-Datei `'coordination.properties'`“ auf Seite 204

Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden.

„Die MFT-Datei 'command.properties'“ auf Seite 209

In der Datei `command.properties` ist der Befehlswarteschlangenmanager angegeben, zu dem eine Verbindung hergestellt werden muss, wenn Befehle ausgegeben werden. Außerdem enthält die Datei Informationen, die Managed File Transfer für den Kontakt zu diesem Warteschlangenmanager benötigt.

„Die MFT agent.properties-Datei“ auf Seite 180

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.properties`, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei `agent.properties` angegeben sein.

SSL/TLS-Eigenschaften für MFT

„Die MFT-Datei 'logger.properties'“ auf Seite 213

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.

MFT-Agenteneigenschaften für Benutzerexits

Format der Protokollbrückeneigenschaftendatei

Format der Datei mit den Connect:Direct-Prozessdefinitionen

Format der Datei mit den Connect:Direct-Knoteneigenschaften

Die MFT-Datei 'installation.properties'

Die Datei `installation.properties` gibt den Namen der Standardgruppe von Konfigurationsoptionen an. Dieser Eintrag verweist Managed File Transfer an eine strukturierte Gruppe mit Verzeichnissen und Eigenschaftendateien, welche die zu verwendende Konfiguration enthält. Gewöhnlich ist der Name einer Gruppe von Konfigurationsoptionen der Name des zugeordneten Koordinations-WS-Managers.

Diese Datei wird vom Installationsprogramm erstellt und kann über den Befehl **fteChangeDefaultConfigurationOptions** geändert werden.

Die Datei `installation.properties` befindet sich im Ihrem Verzeichnis `MQ_DATA_PATH`. Unter Windows ist die Standarddateiposition beispielsweise `MQ_DATA_PATH\mqft\installations\installation_name`; auf AIX and Linux-Systemen lautet die Standarddateiposition `/var/mqm/mqft/installations/installation_name`.

Für den Redistributable Managed File Transfer Agent wird der Datenpfad bei der Ausführung des Befehls **fteCreateEnvironment** gesetzt. Wird der Befehl unter Angabe des gewünschten Verzeichnisses (Parameter **-d**) ausgeführt, wird der Datenpfad auf diesen Wert gesetzt. Wenn Sie im Befehl **fteCreateEnvironment** keinen Datenpfad angeben, wird im Stammverzeichnis das Verzeichnis `mftdata` erstellt, in dem der Redistributable Managed File Transfer Agent dann extrahiert wird. Die Datei `installation.properties` für den Redistributable Managed File Transfer Agent befindet sich im Verzeichnis `MQ_DATA_PATH\mqft\installations\MFTZipInstall`.

Die Datei `installation.properties` enthält die folgenden Werte:




Tabelle 64. Grundlegende Eigenschaften

Eigenschaftsname	Beschreibung	Standardwert
commandMessagePriority	<p>Legt die Priorität sowohl von internen Nachrichten als auch von Befehlsnachrichten für die Befehle fteStopAgent, fteCancelTransfer und ftePingAgent fest.</p> <p>Wenn Sie zum Beispiel sehr viele Übertragungsanforderungen zur Übertragung vieler kleiner Dateien in kurzer Folge übergeben, werden die neuen Übertragungsanforderungen möglicherweise in die Befehlswarteschlange des Quellagenten eingereiht. Für die externen und internen Nachrichten gilt die standardmäßige IBM MQ-Nachrichtenpriorität, sodass die internen Nachrichten durch die neuen Übertragungsanforderungen blockiert werden. Dies kann dazu führen, dass die zulässige Zeit für die Übertragungsvereinbarung überschritten wird und die Übertragungen in den Wiederherstellungsstatus versetzt werden.</p> <p>Sie können mit der Eigenschaft 'commandMessagePriority' auch die Priorität von internen Empfangsbestätigungsnachrichten und von Nachrichten, für die eine Empfangsbestätigung erwartet wird, festlegen.</p> <p>Um den internen Managed File Transfer-Nachrichten eine höhere Priorität als neuen Übertragungsanforderungen zuzuordnen, müssen Sie diese Eigenschaft auf einen Wert zwischen 1 (niedrigste Priorität) und 9 (höchste Priorität) setzen.</p> <p>Die Eigenschaft 'commandMessagePriority' hat den Standardwert 8. Wenn also das IBM MQ-Attribut DEFPRTY (Standardpriorität) in einer Agentenbefehlswarteschlange kleiner-gleich 7 ist, haben interne Vereinbarungsnachrichten Priorität vor neuen Übertragungsanforderungen. Wenn der Wert des Attributs DEFPRTY auf 8 oder 9 gesetzt ist, müssen Sie DEFPRTY oder die Eigenschaft 'commandMessagePriority' ändern, damit die Wirksamkeit der Eigenschaft 'commandMessagePriority' beibehalten bleibt.</p>	<p>Der Standardwert ist 8.</p> <p style="text-align: right;">Konfigurationsreferenz 177</p>

Tabelle 64. Grundlegende Eigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
commonCredentialsKeyFile	<p>Der vollständig qualifizierte Pfadname der Datei mit dem Berechtigungsnachweisschlüssel, der beim Verschlüsseln von Berechtigungsnachweisen verwendet wird. Der häufigste Name der MFT-Berechtigungsnachweisdatei ist <code>MQMFTCcredentials.xml</code>.</p> <p>Weitere Informationen zur Verwendung der Eigenschaft <code>commonCredentialsKeyFile</code> finden Sie unter Berechtigungsnachweise entschlüsseln.</p>	Der vollständig qualifizierte Pfad der Schlüsseldatei.
defaultProperties	Der Name des Standardsatzes an Konfigurationsoptionen. Bei diesem Wert handelt es sich um den Namen eines Verzeichnisses im Konfigurationsverzeichnis, in dem Verzeichnisse und Eigenschaftendateien mit Konfigurationsinformationen enthalten sind.	Kein Standardwert
enableFunctionalFixPack	<p>Die zu aktivierende Funktionsstufe des Fixpacks. Neue Funktionen, die in einem Fixpack enthalten sind, werden standardmäßig nicht aktiviert. Setzen Sie diese Eigenschaft auf eine Versions-ID, um die neuen Funktionen zu aktivieren, die für die betreffende Version verfügbar sind.</p> <p>Sie können die Versions-ID mit Punkten oder ohne Punkte (.) angeben. Wenn Sie beispielsweise die mit IBM MQ 8.0.0 Fix Pack 2 verfügbare Funktion verwenden möchten, setzen Sie diese Eigenschaft auf <code>8002</code> oder <code>8.0.0.2</code>.</p>	Kein Standardwert

Tabelle 64. Grundlegende Eigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
messagePublicationFormat	<p>Ermöglicht Ihnen die Angabe des Formats zur Veröffentlichung von Nachrichten, das von MFT-Agenten für die zugehörigen XML-Statusnachrichten verwendet wird. Für diese Eigenschaft sind die folgenden Werte möglich:</p> <p>messagePublicationFormat=mixed Nachrichten werden ohne ein MQMD-Format (MQFMT_NONE) veröffentlicht, mit Ausnahme der in der Themenstruktur '/LOG' veröffentlichten Nachrichten, welche im MQMD-Format MQFMT_STRING veröffentlicht werden.</p> <p>messagePublicationFormat=MQFMT_NONE Nachrichten werden ohne ein MQMD-Format veröffentlicht.</p> <p>messagePublicationFormat=MQFMT_STRING Nachrichten werden in einem Zeichenfolgeformat veröffentlicht.</p>	messagePublicationFormat=mixed
<p> z/OS-spezifisch:</p>		
productID	<p>Produkttyp für den die MFT-Nutzung erfasst werden soll:</p> <ul style="list-style-type: none"> • Eigenständiges Managed File Transfer-Produkt. (MFT ist die productID). • Komponente eines IBM MQ Advanced-Produkts. (ADVANCED ist die productID). •  Komponente eines IBM MQ Advanced for z/OS Value Unit Edition-Produkts. (ADVANCEDVUE ist die productID). <p>Weitere Informationen zur Aufzeichnung der Produktnutzung finden Sie unter Produktinformationen melden.</p> <p> Diese Eigenschaft wird unter Multiplatforms ignoriert.</p>	MFT

Der folgende Text ist ein Beispiel für den Inhalt einer Datei `installation.properties`.

```
defaultProperties=ERIS
```

ERIS ist der Name eines Verzeichnisses, das sich in demselben Verzeichnis wie die Datei `installation.properties` befindet. Das Verzeichnis ERIS enthält Verzeichnisse und Eigenschaftendateien, die eine Gruppe von Konfigurationsoptionen beschreiben.

Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

Zugehörige Verweise

[fteChangeDefaultConfigurationOptions](#)

Die MFT `agent.properties`-Datei

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.properties`, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei `agent.properties` angegeben sein.

Die Datei `agent.properties` wird vom Installationsprogramm oder vom Befehl **`fteCreateAgent`**, **`fteCreateBridgeAgent`** oder **`fteCreateCDAgent`** erstellt. Alle diese Befehle können Sie mit dem Flag **`-f`** verwenden, um die grundlegenden Eigenschaften des Agentenwarteschlangenmanagers und die erweiterten Agenteneigenschaften, die dem zu erstellenden Agententyp zugeordnet sind, zu ändern. Zum Ändern oder Hinzufügen der erweiterten Agenteneigenschaften muss die Datei in einem Texteditor bearbeitet werden.

Multi Bei der Multiplattform-Version befindet sich die Datei `agent.properties` für einen Agenten im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name`.

z/OS Unter z/OS befindet sich die Datei `agent.properties` an der Position `$BFG_CONFIG variable/mqft/config/coordination_qmgr_name/agents/agent_name`.

Nach einer Änderung der Datei `agent.properties` ist ein Neustart des Agenten erforderlich, damit die Änderungen übernommen werden.

In einigen Managed File Transfer-Eigenschaften, die Datei- oder Verzeichnispositionen darstellen, können Sie Umgebungsvariablen verwenden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt). Weitere Informationen finden Sie unter [„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“](#) auf Seite 174.

Windows

Anmerkung: Unter Windowszwei Eigenschaften:

- `windowsService`
- `windowsServiceVersion`

werden der Datei `agent.properties` mit den MFT-Befehlen hinzugefügt, mit denen ein Agent für die Ausführung als Windows-Dienst konfiguriert wird.

Sie sollten die Eigenschaften nicht manuell hinzufügen oder ändern, da dies die ordnungsgemäße Funktion des Agenten verhindert.

Weitere Informationen zum Einrichten eines Agenten als Windows Service finden Sie unter [Starten eines MFT Agent als Windows Service](#).

Grundlegende Agenteneigenschaften

Jede Datei `MFT agent.properties` enthält die folgenden grundlegenden Agenteneigenschaften:

Tabelle 65. Grundlegende Agenteneigenschaften		
Eigenschaftsname	Beschreibung	Standardwert
agentName	Der Name des Agenten. Der Agentenname muss den Bezeichnungskonventionen für IBM MQ-Objekte entsprechen. Weitere Informationen finden Sie unter „Konventionen zum Benennen von MFT-Objekten“ auf Seite 241.	Kein Standardwert
agentDesc	Die Beschreibung des Agenten, sofern Sie sich dazu entschließen, eine Beschreibung anzugeben.	Kein Standardwert
agentQMGr	Der Name des Agentenwarteschlangenmanagers.	Kein Standardwert
agentQMGrHost	Der Hostname oder die IP-Adresse des Agentenwarteschlangenmanagers.	Kein Standardwert
agentQMGrPort	Die Portnummer, die für Clientverbindungen zum Agentenwarteschlangenmanager verwendet wird.	1414
agentQMGrChannel	Der SVRCONN-Kanalname, der für die Verbindung zum Agentenwarteschlangenmanager verwendet wird.	SYSTEM.DEF.SVRCONN
agentType	Der Typ des Agenten: <ul style="list-style-type: none"> • Standardmäßiger Nicht-Bridgeagent (STANDARD) • Protokollbridgeagent (BRIDGE) • Connect:Direct-Bridgeagent (CD_BRIDGE) • Integrierter Agent, wie von IBM Integration Bus verwendet (EMBEDDED) • In Sterling File Gateway integrierter Agent (SFG) 	STANDARD

Wenn Sie keinen Wert für die Eigenschaft 'agentQMGrHost' angeben, wird standardmäßig der Bindungsmodus verwendet.

Wenn Sie für die Eigenschaft 'agentQMGrHost' einen Wert angeben, nicht jedoch für die Eigenschaften 'agentQMGrPort' und 'agentQMGrChannel', werden standardmäßig Portnummer 1414 und Kanal SYSTEM.DEF.SVRCONN verwendet.

Erweiterte Agenteneigenschaften

Managed File Transfer bietet auch erweiterte Agenteneigenschaften, mit denen Sie Agenten konfigurieren können. Wenn Sie eine der folgenden Eigenschaften verwenden möchten, bearbeiten Sie die Datei `agent.properties` manuell, um die erforderlichen erweiterten Eigenschaften hinzuzufügen. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash

(\) als Escapezeichen versehen werden.  Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Weitere Informationen zur Verwendung von Escapezeichen in Java -Eigenschaftendateien finden Sie in der Oracle -Dokumentation [Javadoc für die Eigenschaftsklasse](#).

- [Eigenschaften für die Agentengröße](#)
- [Codepageeigenschaften](#)
- [Befehlseigenschaften](#)
- [Verbindungseigenschaften](#)
- [Connect:Direct-Bridgeeigenschaften](#)
- [Eigenschaften des Agenten für die Übertragung von Datei an Nachricht und von Nachricht an Datei](#)
- [Allgemeine Agenteneigenschaften](#)
- [Hochverfügbarkeitseigenschaften](#)
- [Ein-/Ausgabeeigenschaften](#)
- [Eigenschaften für das Übertragungsprotokoll](#)
- [Eigenschaften für die Unterstützung auf mehreren Kanälen](#)


- Eigenschaften für Multi-Instanz-Warteschlangen
- Eigenschaften des Prozesscontrollers
- Protokollbridge-Eigenschaften
- Protokolleigenschaften des Protokollbridgeagenten
- Warteschlangeneigenschaften
- Eigenschaften für die Ressourcenüberwachung
- Eigenschaften für das Stammverzeichnis
- Scheduler-Eigenschaft
- Sicherheitseigenschaften
- SSL/TLS-Eigenschaften
- Eigenschaften für das Zeitlimit
- Eigenschaften für das Zeitlimit für die Übertragungswiederherstellung
- Trace- und Protokollierungseigenschaften
- Eigenschaften für die Übertragungsgrenzwerte
- Eigenschaften für die Benutzerexitroutine
- Eigenschaften für die IBM MQ-Clientkomprimierung
-  z/OS-spezifische Eigenschaften
- Andere Eigenschaften

Tabelle 66. Erweiterte Agenteneigenschaften: Agentengröße

Eigenschaftsname	Beschreibung	Standardwert
agentCheckpointInterval	<p>Das Intervall in vollständigen Datenframes, in dem ein Prüfpunkt zu Wiederherstellungszwecken gesetzt wird. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>Wenn es ein Problem gibt, das eine Wiederherstellung der Übertragung verursacht, kann die Übertragung nur bis zu einer Prüfpunktgrenze wiederhergestellt werden. Das heißt, je höher dieser Wert (bei großen agentChunkSize-, agentWindowSize- und agentFrameSize-Werten), desto mehr Zeit benötigt der Agent zur Wiederherstellung von Übertragungen. Für zuverlässige Managed File Transfer-Netze, in denen Übertragungen selten in einen Wiederherstellungsstatus übergehen, kann es vorteilhaft sein, diesen Wert zu erhöhen, um die Gesamtleistung zu verbessern.</p>	1
agentChunkSize	<p>Die Größe der einzelnen Transportblöcke (Chunkgröße) für den Transport der Dateidaten. Der Wert bezeichnet daher die maximale Größe der IBM MQ-Nachrichten, die zwischen den Quellen- und Zielagenten übertragen werden. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>Dieser Wert wird zwischen dem Quellen- und dem Zielagenten verhandelt. Der größere der beiden Werte wird verwendet. Wenn Sie den Wert dieser Eigenschaft ändern möchten, ändern Sie ihn sowohl auf dem Quellen- als auch auf dem Zielagenten.</p> <p>"agentChunkSize" ist ein ganzzahliger Wert. Beispiel: "agentChunkSize=10240" setzt die Chunkgröße auf 10 KB.</p>	262144 Bytes (entspricht 256 KB)
agentFrameSize	<p>Die Anzahl Fenster für den Übertragungsframe. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>In Netzen mit einer langen Latenzzeit kann eine Erhöhung dieses Werts die Gesamtleistung verbessern, da der Agent mehr Nachrichtenblöcke gleichzeitig aktiv verwalten muss.</p> <p>Der Wert dieser Eigenschaft, multipliziert mit agentWindowSize, multipliziert mit agentChunkSize, bezeichnet die Obergrenze der Speicherbelegung des Agenten für jede einzelne Übertragung. Beispiel: 262144-Byte-Blöcke x 10 x 5 = 12,5 MB für jede Übertragung.</p> <p>Hinweis: Wenn die Größe der in einer einzelnen Übertragung gesendeten Dateien kleiner als 12,5 MB ist, hat eine Erhöhung des Werts dieser Eigenschaft keine Auswirkung auf die Leistung der Übertragung.</p>	5

Tabelle 66. Erweiterte Agenteneigenschaften: Agentengröße (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
agentWindowSize	<p>Die Anzahl Blöcke für jedes Fenster. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>In Netzen mit einer langen Latenzzeit kann eine Erhöhung dieses Werts die Gesamtleistung verbessern. Der Grund dafür ist, dass der Agent mehr Nachrichtenblöcke gleichzeitig aktiv verwalten muss und die weniger Bestätigungsnachrichten an den Quellenagenten zurückgesendet werden.</p> <p>Der Wert dieser Eigenschaft, multipliziert mit agentFrameSize, multipliziert mit agentChunkSize, bezeichnet die Obergrenze der Speicherbelegung des Agenten für jede einzelne Übertragung und damit auch die Obergrenze des Volumens an IBM MQ-Nachrichtendaten in der Datenwarteschlange des Zielagenten. Beispiel: 262144-Byte-Blöcke x 10 x 5 = Obergrenze von 12,5 MB für jede Übertragung.</p> <p>Hinweis: Wenn die Größe der in einer einzelnen Übertragung gesendeten Dateien kleiner als 12,5 MB ist, hat eine Erhöhung des Werts dieser Eigenschaft keine Auswirkung auf die Leistung der Übertragung.</p>	10

Tabelle 67. Erweiterte Agenteneigenschaften: Codepage

Eigenschaftsname	Beschreibung	Standardwert
agentCcsid	Die Codepage, mit welcher der Agent eine Verbindung zum Agentenwarteschlangenmanager herstellt. Wenn Sie einen Wert für agentCcsid angeben, müssen Sie auch einen Wert für agentCcsidName angeben. Informationen zum Anzeigen der bekannten Codepages für die JVM finden Sie unter dem Parameter <code>-hsc</code> im Befehl fteCreateBridgeAgent .	1208
agentCcsidName	Die Java-Darstellung der ID des codierten Zeichensatzes für die Koordination (agentCcsid). Wenn Sie einen Wert für agentCcsidName angeben, müssen Sie auch einen Wert für agentCcsid angeben.	UTF8

Tabelle 68. Erweiterte Agenteneigenschaften: Befehl

Eigenschaftsname	Beschreibung	Standardwert
maxCommandHandlerThreads	Steuert die Anzahl der für die erste Syntaxanalyse und die Verarbeitung von Übertragungsbefehlsnachrichten verfügbaren Threads. Aktive Threads benötigen eine Verbindung zum Warteschlangenmanager; wenn sie inaktiv sind, geben die Threads die Verbindung jedoch frei.	5
maxCommandOutput	Die Anzahl an Bytes, die maximal für die Befehlsausgabe gespeichert werden können. Diese Eigenschaft gilt für Befehle, die für einen verwalteten Aufruf angegeben werden, sowie für die Befehle 'preSource', 'postSource', 'preDestination' und 'postDestination' für eine verwaltete Übertragung. Dies begrenzt die Länge der Befehlsausgabe, die im Abschnitt SYSTEM.FTE in das Übertragungsprotokoll geschrieben wird.	10240
maxCommandRetries	Gibt an, wie oft der Agent maximal eine Befehlswiederholung zulässt. Diese Eigenschaft gilt für Befehle, die für einen verwalteten Aufruf angegeben werden, sowie für die Befehle 'preSource', 'postSource', 'preDestination' und 'postDestination' für eine verwaltete Übertragung.	9
maxCommandWait	Die vom Agenten zugelassene maximale Zeitspanne in Sekunden zwischen den Wiederholungen. Diese Eigenschaft gilt für Befehle, die für einen verwalteten Aufruf angegeben werden, sowie für die Befehle 'preSource', 'postSource', 'preDestination' und 'postDestination' für eine verwaltete Übertragung.	60
immediateShutdownTimeout	<p>Beim sofortigen Herunterfahren eines Agenten können Sie mit dieser Eigenschaft die maximale Zeit in Sekunden angeben, die gewartet wird, bis die Übertragungen des Agenten abgeschlossen sind, bevor das Herunterfahren erzwungen wird.</p> <p>Anmerkung: Setzen Sie diese Eigenschaft nicht auf einen Wert unter 10 Sekunden (Standardwert). Bei einer sofortigen Beendigung des Agenten wird ausreichend Zeit zum Beenden aller externen Prozesse benötigt. Wenn der Wert dieser Eigenschaft zu niedrig ist, bleiben einige Prozesse unter Umständen aktiv.</p> <p>Wenn diese Eigenschaft auf 0 gesetzt ist, wartet der Agent, bis alle ausstehenden Übertragungen gestoppt sind. Falls für diese Eigenschaft ein ungültiger Wert angegeben ist, wird der Standardwert verwendet.</p>	10

Eigenschaftsname	Beschreibung	Standardwert
javaLibraryPath	Wenn im Bindungsmodus eine Verbindung zu einem Warteschlangenmanager hergestellt wird, muss Managed File Transfer über Zugriff auf die IBM MQ Java-Bindungsbibliotheken verfügen. Managed File Transfer sucht die Bindungsbibliotheken standardmäßig an der von IBM MQ definierten Standardposition. Wenn sich die Bindungsbibliotheken an einer anderen Position befinden, geben Sie mit dieser Eigenschaft die Position der Bindungsbibliotheken an.	--

Eigenschaftsname	Beschreibung	Standardwert
cdNode	Diese Eigenschaft ist erforderlich, wenn Sie die Connect:Direct-Bridge verwenden möchten. Der Name des Connect:Direct-Knotens, der für die Übertragung von Nachrichten vom Connect:Direct-Bridgeagenten zu den Connect:Direct-Zielknoten verwendet werden soll. Dieser Knoten ist Teil der Connect:Direct-Bridge, also nicht der ferne Knoten, der Quelle oder Ziel der Übertragung ist. Weitere Informationen finden Sie im Abschnitt Die Connect:Direct-Bridge .	Kein Standardwert
cdNodeHost	Der Hostname bzw. die IP-Adresse des Connect:Direct-Knotens, der für die Übertragung von Dateien vom Connect:Direct-Bridgeagenten zu den Zielknoten verwendet werden soll (der Connect:Direct-Bridgeknoten). Meist befindet sich der Connect:Direct-Bridgeknoten auf dem gleichen System wie der Connect:Direct-Bridgeagent. In diesen Fällen ist der Standardwert dieser Eigenschaft, nämlich die IP-Adresse des lokalen Systems, korrekt. Wenn Ihr System über mehrere IP-Adressen verfügt oder sich Ihr Connect:Direct-Bridgeknoten auf einem anderen System als dem System Ihres Connect:Direct-Bridgeagenten befindet und diese Systeme ein Dateisystem gemeinsam nutzen, geben Sie mit dieser Eigenschaft den richtigen Hostnamen für den Connect:Direct-Bridgeknoten an. Falls Sie die Eigenschaft 'cdNode' nicht definiert haben, wird diese Eigenschaft ignoriert.	Der Hostname oder die IP-Adresse des lokalen Systems.
cdNodePort	Die Portnummer des Connect:Direct-Bridgeknotens, die von Clientanwendungen für die Kommunikation mit dem Knoten verwendet wird. In der Connect:Direct-Produktdokumentation wird dieser Port als API-Port bezeichnet. Falls Sie die Eigenschaft 'cdNode' nicht definiert haben, wird diese Eigenschaft ignoriert.	1363
cimpDir	Das Verzeichnis auf dem System, auf dem der Connect:Direct-Bridgeagent ausgeführt wird, in dem Dateien vorübergehend gespeichert werden, bevor sie an den Connect:Direct-Zielknoten übertragen werden. Diese Eigenschaft gibt den vollständigen Pfad des Verzeichnisses an, in dem Dateien vorübergehend gespeichert werden. Ist für 'cdTmpDir' beispielsweise /tmp festgelegt, werden die Dateien vorübergehend in das Verzeichnis /tmp gestellt. Der Connect:Direct-Bridgeagent und der Connect:Direct-Bridgeknoten müssen in der Lage sein, unter Verwendung desselben Pfadnamens auf das über diesen Parameter angegebene Verzeichnis zuzugreifen. Dies muss bei der Planung der Connect:Direct-Bridgeinstallation berücksichtigt werden. Erstellen Sie den Agenten nach Möglichkeit auf dem System, auf dem sich auch der Connect:Direct-Knoten für die Connect:Direct-Bridge befindet. Sind Agent und Knoten auf verschiedenen Systemen installiert, muss sich das Verzeichnis in einem gemeinsam genutzten Dateisystem befinden und es muss möglich sein, von beiden Systemen aus unter Verwendung desselben Pfadnamens auf dieses Verzeichnis zuzugreifen. Weitere Informationen zu den unterstützten Konfigurationen finden Sie im Abschnitt Die Connect:Direct-Bridge . Falls Sie die Eigenschaft 'cdNode' nicht definiert haben, wird diese Eigenschaft ignoriert. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ Verwendung von Umgebungsvariablen in MFT-Eigenschaften “ auf Seite 174.	<code>value_of_java.io.tmpdir /cdbridge-agentName</code>
cdTrace	Gibt an, ob für Daten, die zwischen dem Connect:Direct-Bridgeagenten und seinem Connect:Direct-Knoten hin- und hergesendet werden, vom Agenten ein Trace durchgeführt wird. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> .	false
cdMaxConnectionRetries	Die maximale Anzahl an Connect:Direct-Verbindungsversuchen für eine Dateiübertragung, für die noch keine erfolgreiche Verbindung hergestellt werden konnte, bevor die Übertragung fehlschlägt.	-1 (die Anzahl der Versuche ist unbegrenzt)

Tabelle 70. Erweiterte Agenteneigenschaften: Connect:Direct-Bridge (Forts.)



Eigenschaftsname	Beschreibung	Standardwert
cdMaxPartialWorkConnectionRetries	Die maximale Anzahl an Connect:Direct-Verbindungsversuchen für eine Dateiübertragung, für die bereits erfolgreich eine Verbindung hergestellt und Übertragungsarbeit erledigt wurde, bevor die Übertragung fehlschlägt.	-1 (die Anzahl der Versuche ist unbegrenzt)
cdMaxWaitForProcessEndStats	Gibt an, wie viele Millisekunden maximal gewartet werden soll, bis Informationen zur Beendigung des Connect:Direct-Prozesses nach Abschluss des Prozesses in den Connect:Direct-Knotenstatistikdaten erscheinen. Nach Ablauf dieses Zeitraums wird die Dateiübertragung als gescheitert angesehen. In der Regel sind die Informationen sofort verfügbar, unter bestimmten Fehlerbedingungen werden die Informationen jedoch nicht veröffentlicht. Unter diesen Bedingungen schlägt die Dateiübertragung nach Ablauf des durch diese Eigenschaft angegebenen Zeitraums fehl.	60000
cdAppName	Der Anwendungsname, den der Connect:Direct-Bridgeagent zur Verbindung mit dem zur Bridge gehörenden Connect:Direct-Knoten verwendet.	Managed File Transfer <i>aktuelle Version</i> . Dabei steht <i>aktuelle Version</i> für die Versionsnummer des Produkts.
cdNodeLocalPortRange	Der lokale Portbereich, der für Socketverbindungen zwischen dem Connect:Direct-Bridgeagenten und dem zur Bridge gehörenden Connect:Direct-Knoten verwendet werden kann. Das Format dieses Werts ist eine durch Kommas getrennte Liste von Werten oder Bereichen. Die lokalen Portnummern werden standardmäßig vom Betriebssystem ausgewählt.	--
cdNodeProtocol	Das Protokoll, das der Connect:Direct-Bridgeagent zur Verbindung mit dem zur Bridge gehörenden Connect:Direct-Knoten verwendet. Folgende Werte sind gültig: <ul style="list-style-type: none"> • TCPIP • SSL • TLS 	TCPIP
cdNodeKeystore	Der Pfad zum Keystore, der für die sichere Kommunikation zwischen dem Connect:Direct-Bridgeagenten und dem zur Bridge gehörenden Connect:Direct-Knoten verwendet wird. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ Verwendung von Umgebungsvariablen in MFT-Eigenschaften “ auf Seite 174.	--
cdNodeKeystoreType	Das Dateiformat des Keystores, der durch die Eigenschaft 'cdNodeKeystore' angegeben wird. Die folgenden Werte sind gültig: 'jks' und 'pkcs12'. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert.	jks
cdNodeKeystoreCredentialsFile	Der Pfad der Datei, die die cdNodeKeystore-Berechtigungsnachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ Verwendung von Umgebungsvariablen in MFT-Eigenschaften “ auf Seite 174.  Details zum Erstellen von Berechtigungsnachweisdateien finden Sie unter MFT und IBM MQ -Verbindungsauthentifizierung .  Details zum Erstellen von Berechtigungsnachweisdateien finden Sie unter MQMFTCredentials.xml unter z/OS .	Der Standardwert für diese Eigenschaft ist:    \$HOME/ MQMFTCredentials.xml  %USERPROFI- LE%/MQMFTCredentials.xml
cdNodeTruststore	Der Pfad zum Truststore, der für die sichere Kommunikation zwischen dem Connect:Direct-Bridgeagenten und dem zur Bridge gehörenden Connect:Direct-Knoten verwendet wird. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ Verwendung von Umgebungsvariablen in MFT-Eigenschaften “ auf Seite 174.	--
cdNodeTruststoreType	Das Dateiformat des Truststores, der durch die Eigenschaft 'cdNodeTruststore' angegeben wird. Die folgenden Werte sind gültig: 'jks' und 'pkcs12'. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert.	jks

Tabelle 70. Erweiterte Agenteneigenschaften: Connect:Direct-Bridge (Forts.)







Eigenschaftsname	Beschreibung	Standardwert
cdNodeTruststoreCredentialsFile	<p>Der Pfad der Datei, die die cdNodeTruststore-Berechtigungs-nachweise enthält.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.</p> <p> Details zum Erstellen von Berechtigungs-nachweisdateien finden Sie unter MFT und IBM MQ -Verbindungsauthentifizierung.</p> <p> Details zum Erstellen von Berechtigungs-nachweisdateien finden Sie unter MQMFTCredentials.xml unter z/OS.</p>	<p>Der Standardwert für diese Eigenschaft ist:</p> <p> z/OS</p> <p> Linux</p> <p> AIX \$HOME/MQMFTCredentials.xml</p> <p> Windows %USERPROFILE%/MQMFTCredentials.xml</p>
logCDProcess	Die Protokollstufe der Connect:Direct-Prozessprotokollierung, die im Ereignisprotokoll des Agenten in der Datei output0.log aufgezeichnet wird. Mögliche Werte für diese Eigenschaft sind None (Keine), Failures (Fehler) oder All (Alle).	--

Tabelle 71. Erweiterte Agenteneigenschaften: Übertragung von Datei an Nachricht und Nachricht an Datei

Eigenschaftsname	Beschreibung	Standardwert
deleteTmpFileAfterRenameFailure	Wenn Sie diese Eigenschaft auf den Wert 'false' setzen, wird dadurch sichergestellt, dass temporäre Dateien nicht aus dem Ziel gelöscht werden, wenn die Umbenennungsoperation fehlschlägt. In diesem Fall bleiben die übertragenen Daten im Ziel in einer temporären Datei (.part). Sie können diese Datei später manuell umbenennen. Standardmäßig hat diese Eigenschaft den Wert 'true'. Diese Eigenschaft ist sowohl für Übertragungen von Nachrichten an Dateien als auch von Dateien an Dateien gültig.	true
enableQueueInputOutput	Standardmäßig kann der Agent als Teil der Übertragung keine Daten aus einer Quellenwarteschlange lesen oder in eine Zielwarteschlange schreiben. Wenn für diese Eigenschaft der Wert 'true' angegeben wird, kann der Agent Übertragungen von Dateien an Nachrichten und von Nachrichten an Dateien vornehmen. Mögliche Werte für diese Eigenschaft sind true oder false.	false
enableSystemQueueInputOutput	Gibt an, ob der Agent in oder aus IBM MQ-Systemwarteschlangen schreiben kann. Systemwarteschlangen ist das Qualifikationsmerkmal SYSTEM vorangestellt. Anmerkung: Systemwarteschlangen werden von IBM MQ, Managed File Transfer und anderen Anwendungen zur Übertragung wichtiger Informationen verwendet. Durch eine Änderung dieser Eigenschaft ermöglichen Sie dem Agenten den Zugriff auf diese Warteschlangen. Wenn Sie diese Eigenschaft auf 'true' setzen, sollten Sie die Warteschlangen, auf die der Agent zugreifen kann, mittels Benutzer-Sandboxing einschränken.	false
enableClusterQueueInputOutput	Gibt an, ob der Agent für IBM MQ-Clusterwarteschlangen über Lese- oder Schreibzugriff verfügt. Anmerkung: Sie müssen zusätzlich zur Eigenschaft 'enableQueueInputOutput' die Agenteneigenschaft 'enableClusterQueueInputOutput' angeben.	false
maxDelimiterMatchLength	Die maximale Anzahl an Zeichen, die mit dem regulären Java-Ausdruck abgeglichen werden kann, mit dem eine Textdatei als Teil einer Übertragung aus einer Datei in eine Nachricht in mehrere Nachrichten aufgeteilt wird.	5
maxInputOutputMessageLength	Die maximale Länge einer Nachricht in Byte, die ein Agent aus einer Quellenwarteschlange liest bzw. in eine Zielwarteschlange schreibt. Die Eigenschaft 'maxInputOutputMessageLength' des Quellenagenten einer Übertragung bestimmt, wie viele Bytes aus einer Nachricht in der Quellenwarteschlange gelesen werden können. Die Eigenschaft 'maxInputOutputMessageLength' des Zielagenten einer Übertragung bestimmt, wie viele Bytes in eine Nachricht in der Zielwarteschlange geschrieben werden können. Wenn die Länge einer Nachricht den Wert dieser Eigenschaft überschreitet, schlägt die Übertragung mit einem Fehler fehl. Diese Eigenschaft hat keine Auswirkung auf interne Managed File Transfer-Warteschlangen. Informationen zum Ändern dieser Eigenschaft finden Sie im Abschnitt Hinweise zur Definition der MQ-Attribute und MFT-Eigenschaften für die Nachrichtengröße .	1048576

Tabelle 71. Erweiterte Agenteneigenschaften: Übertragung von Datei an Nachricht und Nachricht an Datei (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
monitorGroupRetryLimit	<p>Gibt an, wie oft ein Überwachungsprogramm maximal eine erneute Übertragung aus einer Nachricht in eine Datei auslöst, wenn die Nachrichtengruppe weiterhin in der Warteschlange verbleibt. Wie oft die Übertragung aus einer Nachricht in eine Datei ausgelöst wird, ist im MQMD-Rücksetzungszähler der ersten Nachricht in der Gruppe festgelegt.</p> <p>Bei einem Neustart des Agenten löst das Überwachungsprogramm erneut eine Übertragung aus. Dies gilt selbst dann, wenn die Anzahl der ausgelösten Übertragungen den Wert der Eigenschaft 'monitorGroupRetryLimit' überschreitet. Wenn dieses Verhalten dazu führt, dass die Anzahl der ausgelösten Übertragungen den Wert von 'monitorGroupRetryLimit' überschreitet, schreibt der Agent einen Fehler in sein Ereignisprotokoll.</p> <p>Wird für diese Eigenschaft der Wert -1 angegeben, löst das Überwachungsprogramm die Übertragung unbegrenzt oft erneut aus, solange die Auslöserbedingung nicht erfüllt ist.</p>	10

Tabelle 72. Erweiterte Agenteneigenschaften: Allgemein		
Eigenschaftsname	Beschreibung	Standardwert
agentStatusPublishRateLimit	<p>Die maximale Geschwindigkeit in Sekunden, mit der der Agent seinen Status aufgrund einer Änderung des Dateiübertragungsstatus neu veröffentlicht.</p> <p>Wenn Sie für diese Eigenschaft einen zu niedrigen Wert festlegen, kann dies die Leistung des IBM MQ-Netzes negativ beeinflussen.</p>	30
agentStatusPublishRateMin	<p>Die Mindestrate in Sekunden, in der der Agent seinen Status veröffentlicht. Dieser Wert muss größer-gleich dem Wert der Eigenschaft agentStatusPublishRateLimit sein.</p>	300
enableMemoryAllocationChecking	<p>Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>. Sie legt fest, ob der Managed File Transfer Agent überprüft, ob genügend Speicher für die Ausführung einer Übertragung verfügbar ist, bevor eine Übertragung akzeptiert wird. Die Prüfung erfolgt sowohl auf dem Quellen- als auch dem Zielagenten. Reicht der Speicher nicht aus, wird die Übertragung abgelehnt.</p> <p>Bei der Berechnung des für die Übertragung erforderlichen Speicherplatzes wird von dem maximal erforderlichen Speicherplatz ausgegangen. Deshalb kann der Wert höher sein, als der tatsächlich von der Übertragung belegte Speicherplatz. Aus diesem Grund kann die Anzahl der gleichzeitigen Übertragungen, die ausgeführt werden können, reduziert werden, wenn die Eigenschaft enableMemoryAllocationChecking auf „wahr“ gesetzt ist. Deswegen sollten Sie die Eigenschaft nur dann auf 'true' setzen, wenn Managed File Transfer häufiger mit einem Fehler aufgrund abnormaler Speicherbedingungen fehlschlägt. Bei Übertragungen aus Dateien in Nachrichten und aus Nachrichten in Dateien mit umfangreichen Nachrichtengrößen ist die Belegung großer Speichermengen wahrscheinlich.</p>	false
enableDetailedReplyMessages	<p>Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>. Wenn diese Eigenschaft auf <code>true</code> gesetzt ist, enthalten Antworten auf gesteuerte Übertragungsanforderungen Detailinformationen zu den übertragenen Dateien. Die Detailinformationen und das Format sind identisch mit den im Übertragungsprotokoll in den Fortschrittsnachrichten, d. h. dem Element <transferSet>, veröffentlichten Angaben. Weitere Informationen finden Sie unter Formate Dateiübertragungsprotokollnachricht.</p> <p>Die detaillierten Antwortinformationen sind nur enthalten, wenn die gesteuerte Übertragungsanforderung angibt, dass detaillierte Antwortinformationen erforderlich sind. Um diese Anforderung anzugeben, setzen Sie das Attribut detailed des Elements <reply> der managedTransfer-XML-Anforderungsnachricht, die an den Quellenagenten gesendet wird. Weitere Informationen finden Sie im Abschnitt Format für Dateiübertragungsanforderungsnachricht.</p> <p>Für jede Übertragungsanforderung können mehrere Antwortnachrichten generiert werden. Diese Anzahl entspricht der Anzahl der Übertragungsprotokoll-Statusnachrichten für die Übertragung plus 1 (hierbei ist die erste Antwortnachricht eine einfache ACK-Antwort). Detaillierte Informationen sind in allen Nachrichten enthalten, mit Ausnahme der ACK-Antwortnachrichten; das Gesamttransferergebnis ist jedoch nur in der aktuellsten ausführlichen Antwortnachricht enthalten.</p>	true

Tabella 72. Erweiterte Agenteneigenschaften: Allgemein (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
enableUserMetadataOptions	<p>Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>. Sie legt fest, ob Sie bekannte Schlüssel für benutzerdefinierte Metadaten in neuen Übertragungsanforderungen verwenden können, um mehr Übertragungsoptionen bereitzustellen. Diese bekannten Schlüssel beginnen immer mit dem Präfix <code>com.ibm.wmqfte.</code>. Wenn die Eigenschaft enableUserMetadataOptions auf <code>Wahr</code> gesetzt ist, werden Schlüssel, die dieses Präfix verwenden, nicht für die benutzerdefinierte Verwendung unterstützt. Wenn die Eigenschaft enableUserMetadataOptions auf <code>Falsch</code> gesetzt ist, werden derzeit die folgenden Schlüssel unterstützt:</p> <p>com.ibm.wmqfte.insertRecordLineSeparator</p> <p>Für Textübertragungen. Wenn dieser Schlüssel auf <code>true</code> gesetzt ist, werden beim Lesen von satzorientierten Dateien, z. B. z/OS Datasets, Zeilentrennzeichen zwischen Datensätzen eingefügt.</p> <p>Bei Angabe von <code>false</code> für diesen Schlüssel müssen beim Lesen von satzorientierten Dateien keine Zeilentrennzeichen zwischen Datensätzen eingefügt werden.</p> <p>com.ibm.wmqfte.newRecordOnLineSeparator</p> <p>Für Textübertragungen. Wenn dieser Schlüssel auf <code>true</code> gesetzt ist, gibt dies an, dass Zeilentrennzeichen beim Schreiben in satzorientierte Dateien, wie z. B. z/OS Datasets, einen neuen Datensatz angeben und nicht als Teil der Daten geschrieben werden.</p> <p>Bei Angabe von <code>false</code> für diesen Schlüssel müssen Zeilentrennzeichen beim Schreiben von satzorientierten Dateien wie jedes andere Zeichen (also nicht als Zeilenumbrüche) behandelt werden.</p> <p>com.ibm.wmqfte.convertLineSeparators</p> <p>Für Textübertragungen. Gibt an, ob die Zeilentrennzeichenfolgen CRLF und LF in Zeilentrennzeichenfolgen für das Ziel umgewandelt werden. Diese Umwandlung findet zurzeit nur in folgenden Fällen statt:</p> <ul style="list-style-type: none"> • Wenn der benutzerdefinierte Metadaten Schlüssel com.ibm.wmqfte.newRecordOnLineSeparator auf <code>Falsch</code> gesetzt ist und die Übertragung in eine satzorientierte Datei erfolgt. • Wenn der benutzerdefinierte Metadaten Schlüssel com.ibm.wmqfte.com.ibm.wmqfte.insertRecordLineSeparator auf <code>Falsch</code> gesetzt ist und die Übertragung aus einer datensatzorientierten Datei erfolgt. <p>Siehe auch fteCreateTransfer: Neue Dateiübertragung starten.</p>	false
failTransferOnFirstFailure	<p>Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>. Ein Agent kann so konfiguriert werden, dass eine verwaltete Übertragung fehlschlägt, sobald ein Übertragungselement innerhalb der verwalteten Übertragung fehlschlägt.</p> <p>Um diese Funktion zu aktivieren, muss APAR IT03450 sowohl für den Quellenagenten als auch für den Zielagenten angewendet werden und die Eigenschaft failTransferOnFirstFailure muss in der Datei <code>agent.properties</code> des Quellenagenten auf <code>Wahr</code> gesetzt sein. Im Zielagenten ist das Festlegen der Eigenschaft auf <code>true</code> optional.</p> <p>Wenn die Eigenschaft failTransferOnFirstFailure auf <code>Wahr</code> gesetzt ist, beginnt der Agent normal mit der Verarbeitung von verwalteten Übertragungsanforderungen. Sobald jedoch ein Übertragungselement fehlschlägt, wird die verwaltete Übertragung als fehlgeschlagen gekennzeichnet und es werden keine weiteren Übertragungselemente verarbeitet. Mit Übertragungselementen, die vor dem Fehlschlagen der verwalteten Übertragung erfolgreich verarbeitet wurden, wird wie folgt verfahren:</p> <ul style="list-style-type: none"> • Die Quellendisposition für diese Übertragungselemente wird berücksichtigt. Wenn die Quellendisposition für das Übertragungselement beispielsweise auf <code>delete</code> (Löschen) gesetzt war, wird die Quellendatei gelöscht. • Die festgeschriebenen Zieldateien bleiben im Zielsystem und werden nicht gelöscht. <p>Wenn die Eigenschaft failTransferOnFirstFailure nicht auf <code>Wahr</code> gesetzt ist und eine verwaltete Dateiübertragung mehrere Dateien enthält und eine dieser Dateien nicht übertragen werden kann, beispielsweise weil die Zieldatei bereits vorhanden ist und die Überschreibungseigenschaft auf <code>Falsch</code> gesetzt ist, wird der Quellenagent fortgesetzt und versucht, alle verbleibenden Dateien in der Anforderung zu übertragen.</p>	false

Tabelle 72. Erweiterte Agenteneigenschaften: Allgemein (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
itemsPerProgressMessage	<p>Die Anzahl Dateien, die übertragen werden, bevor ein Agent seine nächste Fortschrittsprotokollnachricht veröffentlicht. Über diese Eigenschaft steuern Sie die Rate, in der während einer Übertragung Fortschrittsprotokollnachrichten an den Koordinationswarteschlangenmanager veröffentlicht werden.</p> <p>Der maximale Wert für diese Eigenschaft ist 1000.</p> <p>Anmerkung: Zu den Fortschrittsnachrichten gehören Informationen zu jeder Datei, die seit der Veröffentlichung der letzten Fortschrittsnachricht übertragen wird. Bei einer Erhöhung dieses Werts nimmt die Größe der Fortschrittsnachrichten zu, was die Leistung beeinträchtigen kann.</p>	50
maxInlineFileSize	<p>Bei Übertragungen aus Dateien in Dateien oder aus Dateien in Nachrichten die maximale Dateigröße (in Bytes), die automatisch in die erste Übertragungsanforderungsnachricht eingeschlossen werden kann.</p> <p>Mit dieser Eigenschaft können Sie die Geschwindigkeit der Übertragungen verbessern, doch wenn der als Dateigröße angegebene Wert zu groß ist, kann die Leistung dadurch auch beeinträchtigt werden. Als Anfangswert wird für diese Eigenschaft eine Dateigröße von 100 KB vorgeschlagen, doch es wird dringend empfohlen, sorgfältig mit verschiedenen Werten zu experimentieren, bevor Sie die für Ihr System geeignetste Dateigröße festlegen.</p> <p>Diese Funktion ist standardmäßig inaktiviert, oder indem Sie die Eigenschaft maxInlineFileSize auf 0 setzen.</p>	0

Tabelle 73. Erweiterte Agenteneigenschaften: Hochverfügbarkeit

Eigenschaftsname	Beschreibung	Standardwert
highlyAvailable	<p>Diese Eigenschaft wird beim Start des Agenten gelesen. Wenn als Wert <code>true</code> angegeben ist, wird der Agent im Hochverfügbarkeitsmodus gestartet. Wenn Sie die Eigenschaft nicht angeben oder den Wert auf 'false' setzen, wird der Agent als nicht hochverfügbarer Agent gestartet.</p>	false
standbyPoll-Intervall	<p>Diese Eigenschaft wird von der Standby-Instanz verwendet, um zu versuchen, die gemeinsam genutzte Warteschlange in bestimmten Intervallen zu öffnen.</p> <p>Ab IBM MQ 9.3.0 wird diese Eigenschaft auch von allen Instanzen verwendet, um festzustellen, wie lange eine Instanz zwischen den Versuchen zur Verbindungswiederholung wartet, wenn sie von ihrem Agentenwarteschlangenmanager getrennt wird.</p> <p>Die Versuche werden wiederholt, bis entweder eine Instanz erneut eine Verbindung zu ihrem Agentenwarteschlangenmanager herstellt, die <code>SYS-TEM.FTE.HA.<agent name>-Warteschlange</code> öffnet (wenn sie sich bereits als Standby-Instanz registriert hat) oder mit dem Befehl fteStopAgent gestoppt wird.</p>	5 Sekunden
standbyStatusDiscardTime	<p>Diese Eigenschaft legt fest, wie lange die aktive Instanz auf eine Statusveröffentlichung von einer Standby-Instanz wartet.</p> <p>Wenn auch nach dieser Wartezeit keine Veröffentlichung von einer Standby-Instanz empfangen wird, entfernt die aktive Instanz die Informationen zu der Standby-Instanz aus ihrer Liste der Standby-Instanzen.</p> <p>Der Standardwert ist doppelt so hoch wie der der Eigenschaft standbyStatusPublishInterval, sodass die aktive Instanz länger wartet, bevor sie die Standby-Instanz aus ihrer Liste entfernt.</p>	600 Sekunden
standbyStatusAblauf	<p>Diese Eigenschaft legt die Ablaufzeit der Standby-Statusnachricht fest, die in die Befehlswarteschlange eines Agenten eingereicht wird. Die Nachricht läuft ab, wenn die aktive Instanz eines Agenten die Nachricht nicht verarbeitet.</p>	30 Sekunden
standbyStatusPublishInterval	<p>Diese Eigenschaft legt fest, wie oft die Standby-Instanz ihren Status veröffentlicht.</p>	300 Sekunden

Tabelle 74. Erweiterte Agenteneigenschaften: Ein-/Ausgabe

Eigenschaftsname	Beschreibung	Standardwert
doNotUseTempOutputFile	<p>Standardmäßig schreibt der Agent Daten in eine temporäre Datei am Ziel und benennt diese temporäre Datei nach Abschluss der Dateiübertragung in den erforderlichen Dateinamen um. Wird dieser Wert auf 'true' gesetzt, schreibt der Agent direkt in die endgültige Zieldatei.</p> <p>z/OS Auf z/OS-Systemen gilt dieses Verhalten nicht für sequenzielle Datasets, sondern für Member der partitionierten Datei.</p> <p>Der Wert dieser Eigenschaft für eine Übertragung wird vom Zielagenten definiert.</p>	false
enableMandatoryLocking	<p>Beim Zugriff auf normale Dateien verwendet Managed File Transfer für den Lesevorgang eine gemeinsame Sperre und für den Schreibvorgang eine exklusive Sperre.</p> <p>Windows Unter Windows stellt die Dateisperre lediglich eine Empfehlung dar. Wenn diese Eigenschaft auf 'true' gesetzt ist, erzwingt Managed File Transfer die Dateisperre. Unter Windows bedeutet dies, dass die Überwachung einer Datei, die durch eine andere Anwendung geöffnet ist, erst nach dem Schließen der Datei ausgelöst wird. Managed File Transfer-Übertragungen, an denen diese Datei beteiligt ist, schlagen fehl.</p> <p>UNIX Auf UNIX-Plattformen wird die Dateisperre prozessübergreifend ausgeführt. Auf UNIX-Plattformen hat die Einstellung dieser Eigenschaft keine Auswirkung.</p> <p>Diese Eigenschaft gilt nur für normale Managed File Transfer-Agenten. Managed File Transfer unterstützt den Mechanismus für die Dateisperre auf einem Dateiserver nicht. Diese Eigenschaft funktioniert daher nicht für einen Protokollbridgeagenten, da ein Protokollbridgeagent eine Datei beim Übertragen einer Datei nicht auf einem Dateiserver sperrt.</p> <p>Mögliche Werte für diese Eigenschaft sind true oder false.</p>	false
ioIdleThreadTimeout	<p>Gibt in Millisekunden an, wie lange eine Inaktivität bei einem Ein-/Ausgabethread des Dateisystems vorliegen muss, bevor der Thread beendet wird.</p> <p>z/OS Ab IBM MQ 9.3.0 gilt diese Eigenschaft nicht für Agenten, die unter IBM MQ für z/OS ausgeführt werden.</p>	10000
ioQueueDepth	<p>Die Anzahl an Eingabe-/Ausgabeanforderungen, die maximal in die Warteschlange eingereicht werden können.</p>	10
ioThreadPoolSize	<p>Maximale Anzahl der verfügbaren Ein-/Ausgabethreads des Dateisystems. Für gewöhnlich verwendet jede Übertragung ihren eigenen Ein-/Ausgabethread des Dateisystems; wenn die Anzahl der gleichzeitig ablaufenden Übertragungen jedoch diesen Grenzwert überschreitet, werden die Ein-/Ausgabethreads des Dateisystems von mehreren Übertragungen geteilt.</p> <p>Ist davon auszugehen, dass regelmäßig mehr gleichzeitig ablaufende Übertragungen stattfinden werden als über den Wert 'ioThreadPoolSize' angegeben, kann unter Umständen eine Leistungsverbesserung erzielt werden, wenn dieser Wert erhöht wird, sodass es für jede Übertragung einen eigenen Ein-/Ausgabethread des Dateisystems gibt.</p>	10
textReplacementCharacterSequence	<p>Wenn bei einer Übertragung im Textmodus Datenbytes nicht von der Codepage der Quelle in die Codepage des Ziels konvertiert werden können, schlägt die Dateiübertragung standardmäßig fehl.</p> <p>Wenn Sie diese Eigenschaft festlegen, kann die Übertragung erfolgreich abgeschlossen werden, indem der angegebene Zeichenwert eingefügt wird. Dieser Eigenschaftswert ist ein einzelnes Zeichen. Normalerweise wird ein Fragezeichen (?) für Zeichen verwendet, die sich nicht zuordnen lassen. Verwenden Sie beispielsweise das Format 'textReplacementCharacterSequence=?'. Dabei ist das Fragezeichen (?) das Ersatzzeichen. Die Verwendung eines Leerzeichens als Ersatzzeichen ist nicht zulässig.</p>	--

Tabelle 75. Erweiterte Agenteneigenschaften: Übertragungsprotokoll		
Eigenschaftsname	Beschreibung	Standardwert
logTransfer Beispiele für die erstellten Protokollinformationen finden Sie in „Von der Funktion 'LogTransfer' erzeugte Ausgabe“ auf Seite 223.	Mit dieser Eigenschaft aktivieren oder inaktivieren Sie die Übertragungsprotokollierung. Folgende Werte sind möglich: info Stellt allgemeine Protokollinformationen für eine Übertragung bereit. Dies ist der Standardwert. moderate Stellt Protokollinformationen für eine Übertragung bereit, deren Detaillierungsgrad zwischen 'info' und 'verbose' liegt. verbose Stellt detaillierte Protokollinformationen zu einer Übertragung bereit. off Inaktiviert die Übertragungsprotokollierung.	info
logTransferFileSize	Definiert die maximale Größe einer Übertragungsprotokolldatei in Megabyte.	20
logTransfer-Dateien	Definiert die maximale Anzahl der Übertragungsdateien, die aufbewahrt werden, bevor die älteste Datei gelöscht wird.	5

Tabelle 76. Erweiterte Agenteneigenschaften: Unterstützung auf mehreren Kanälen		
Eigenschaftsname	Beschreibung	Standardwert
agentMultipleChannelsEnabled	Wenn Sie diese Eigenschaft auf <code>true</code> setzen, kann ein Managed File Transfer Agent Übertragungsdatennachrichten über mehrere IBM MQ -Kanäle senden. In einigen Szenarios kann die Einstellung dieser Eigenschaft die Leistung verbessern. Aktivieren Sie die Mehrkanalunterstützung jedoch nur, wenn eine erkennbare Leistungsverbesserung eintritt. Es werden nur Nachrichten, die in die Warteschlange 'SYSTEM.FTE.DATA.Zielagentenname' gestellt werden, über mehrere Kanäle gesendet. Das Verhalten aller anderen Nachrichten bleibt davon unberührt. Wenn Sie diese Eigenschaft auf <code>true</code> setzen, müssen Sie zur Aktivierung der Unterstützung mehrerer Kanäle auch die IBM MQ-Konfigurationsschritte ausführen, die in einem der folgenden Abschnitte beschrieben werden: <ul style="list-style-type: none"> • MFT-Agenten für mehrere Kanäle in einem Cluster konfigurieren • MFT-Agenten für mehrere Kanäle clusterunabhängig konfigurieren Darüber hinaus müssen Sie die für einen Managed File Transfer-Agenten erforderliche IBM MQ-Standardkonfiguration ausführen (Details enthält der Abschnitt MFT für die erste Verwendung konfigurieren). Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> .	false
agentMessageBatchSize	Wenn ein Quellenagent mit mehreren Kanälen konfiguriert wurde, sendet er Datennachrichten für eine Übertragung auf Umlaufbasis über jeden einzelnen Kanal. Diese Eigenschaft steuert die Anzahl der Nachrichten, die an jeweils einen Kanal gesendet werden.	5

Tabelle 77. Erweiterte Agenteneigenschaften: Manager für Multi-Instanz-Warteschlangen		
Eigenschaftsname	Beschreibung	Standardwert
agentQMGrStandby	Der Hostname und die Portnummer für Clientverbindungen (im IBM MQ CON-NAME-Format), und zwar für die Standby-Instanz eines Agentenwarteschlangenmanagers mit mehreren Instanzen (definiert über die Eigenschaft 'connectionQMGr'). Zum Beispiel, <code>host_name(port_number)</code> The agent attempts to connect to the standby queue manager when it detects a connection broken error, for example, MQRC 2009. Sobald der Agent mit dem Standby-WS-Manager verbunden wird, bleibt der Agent verbunden, bis der Standby-WS-Manager nicht mehr verfügbar ist.	Kein Standardwert

Tabelle 78. Erweiterte Agenteneigenschaften: Prozesscontroller		
Eigenschaftsname	Beschreibung	Standardwert
agentQMGrRetryInterval	Das Zeitintervall in Sekunden, in dem der Prozesscontroller des Agenten prüft, ob der Warteschlangenmanager verfügbar ist.	30

Tabelle 78. Erweiterte Agenteneigenschaften: Prozesscontroller (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
maxRestartCount	Die maximale Anzahl Neustarts, die innerhalb des Zeitintervalls, das durch den Wert der Eigenschaft 'maxRestartInterval' angegeben ist, ausführbar sind. Wird dieser Wert überschritten, führt der Prozesscontroller des Agenten keinen Neustart des Agenten mehr durch; stattdessen führt er eine Aktion aus, die auf dem Wert der Eigenschaft 'maxRestartDelay' basiert.	4
maxRestartInterval	Der Zeitraum in Sekunden, über den der Prozesscontroller des Agenten die Agentenneustarts zählt. Überschreitet die Anzahl der Neustarts innerhalb dieses Zeitraums den Wert der Eigenschaft 'maxRestartCount', führt der Prozesscontroller des Agenten keine Agentenneustarts mehr durch. Stattdessen führt der Prozesscontroller des Agenten eine Aktion aus, die auf dem Wert der Eigenschaft 'maxRestartDelay' basiert.	120
maxRestartDelay	Gibt das Verhalten des Agentenprozesscontrollers an, wenn die Anzahl der Agentenneustarts den Wert der Eigenschaften 'maxRestartCount' und 'maxRestartInterval' übersteigt. Wenn Sie einen Wert kleiner-gleich null angeben, wird der Prozesscontroller des Agenten gestoppt. Wenn Sie einen Wert größer als null angeben, ist dies die Zeit in Sekunden, die gewartet wird, bevor die im Prozesscontroller des Agenten enthaltenen Protokolldaten zu Neustarts zurückgesetzt werden und der Agent erneut gestartet wird.	-1

Tabelle 79. Erweiterte Agenteneigenschaften: Protokollbridge		
Eigenschaftsname	Beschreibung	Standardwert
protocolBridgeCredentialConfiguration	Der Wert dieser Eigenschaft wird als Zeichenfolge an die Methode 'initialize()' der Exitklassen übergeben, die durch die Eigenschaft 'protocolBridgeCredentialExitClasses' angegeben sind.	null
protocolBridgeCredentialExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerroutine für Protokollbridgeberechtigungsanfrage implementieren. Weitere Informationen finden Sie im Abschnitt Berechtigungsanfrage für einen Dateiserver mittels Exitklassen zuordnen.	Kein Standardwert.
protocolBridgeDataTimeout	Die Zeit in Millisekunden, die der Protokollbridgeagent wartet, bis entweder eine Datenverbindung mit einem FTP-Server zustande kommt oder bis Daten von einem FTP-Server über eine bereits vorhandene Verbindung eingehen. Der Wert 0 bedeutet kein Zeitlimit. Bei Verstreichen des Zeitlimits schließt der Protokollbridgeagent alle bestehenden Datenverbindungen mit dem FTP-Server und versucht die aktuelle Übertragung über eine neue Datenverbindung fortzusetzen. Lässt sich diese neue Datenverbindung nicht herstellen, schlägt auch die aktuelle Übertragung fehl.	0
protocolBridgeLogoutBeforeDisconnect	Gibt an, ob der Protokollbridgeagent den Benutzer vom Dateiserver abmeldet, bevor er die FTP-Sitzung abschließt und die Verbindung unterbricht. Wenn Sie diese Eigenschaft auf <code>true</code> setzen, gibt der Protokollbridgeagent einen QUIT -Befehl für die FTP-Sitzung an den Dateiserver aus.	false
protocolBridgePropertiesConfiguration	Der Wert dieser Eigenschaft wird an die Methode 'initialize()' der Exitklassen, die durch die Eigenschaft 'protocolBridgeServerPropertiesExitClasses' angegeben sind, als eine der Bridgeigenschaften übergeben.	Kein Standardwert
protocolBridgePropertiesExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerroutine für Servereigenschaften der Protokollbridge implementieren. Weitere Information finden Sie im Abschnitt ProtocolBridgePropertiesExit2: Protokolldateiservereigenschaften nachschlagen .	Kein Standardwert

Tabelle 80. Erweiterte Agenteneigenschaften: Protokollierung für Protokollbridgeagenten		
Eigenschaftsname	Beschreibung	Standardwert
agentLog	Schlüssel/Wert-Paar-Komponente und Operation zur Aktivierung bzw. Inaktivierung der Protokollierung von FTP-Befehlen und -Antworten zwischen dem Protokollbridgeagenten und den FTP-/SFTP-/FTPS-Dateiservern. For example: agentLog=on Aktiviert die Protokollierung für alle Komponenten. agentLog=off Inaktiviert die Protokollierung für alle Komponenten. agentLog=ftp=on, sftp=on, ftps=off Aktiviert die Protokollierung für FTP und SFTP, inaktiviert sie für FTPS.	Kein Standardwert
agentLogFileSize	Gibt die maximale Größe einer Aufzeichnungsdatei in Megabytes an. Standard wie Standarddateigröße für regulären Trace.	20

Tabelle 80. Erweiterte Agenteneigenschaften: Protokollierung für Protokollbridgeagenten (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
agentLogFiles	<p>Gibt an, wie viele Aufzeichnungsdateien maximal aufbewahrt werden, bevor die älteste Datei gelöscht wird.</p> <ul style="list-style-type: none"> Der Standardwert der Agenteneigenschaft agentLogFiles wurde von 10 in 5 geändert. Dies bedeutet, dass ab IBM MQ 9.3.0, wenn der Standardwert festgelegt ist, maximal fünf Ereignisprotokolldateien des Protokollbridgeagenten vorhanden sein können, beginnend mit <code>agentevent0.log</code> bis <code>agentevent4.log</code>. Sie können diesen Wert jedoch bei Bedarf ändern. Falls der Agent aus einer Version vor IBM MQ 9.3.0 migriert wird, sollten Sie die Dateien <code>agentevent5.log</code> bis <code>agentevent9.log</code> manuell löschen, falls sie vorhanden sind. Die Größe der einzelnen Protokolldateien beträgt jedoch weiterhin 20 MB. 	Ab IBM MQ 9.3.0 ist der Standardwert 5.
agentLogFilter	<p>Erfasst standardmäßig die Kommunikation mit allen FTP-Servern, mit denen der Agent eine Verbindung herstellt.</p> <p>For example:</p> <ul style="list-style-type: none"> Filter für Host/IP-Adresse: <pre>host=ftpprod.ibm.com, ftp2.ibm.com host=9.182.*</pre> Filter auf Basis der Metadaten <pre>metadata="outbound files to xyz corp"</pre> 	*

Tabelle 81. Erweiterte Agenteneigenschaften: Warteschlange

Eigenschaftsname	Beschreibung	Standardwert
dynamicQueuePrefix	Diese Eigenschaft definiert das Präfix, das beim Erstellen einer temporären dynamischen Warteschlange verwendet wird.	WMQFTE.*
modelQueueName	Diese Eigenschaft definiert den Namen der Modulwarteschlange, die beim Erstellen einer temporären dynamischen Warteschlange verwendet wird.	SYSTEM.DEFAULT.MODEL.QUEUE
publicationMDUser	Die MQMD-Benutzer-ID, die den Nachrichten zugeordnet wird, welche zur Veröffentlichung durch den Koordinationswarteschlangenmanager gesendet werden. Wenn Sie diese Eigenschaft nicht festlegen, wird die MQMD-Benutzer-ID auf Basis der IBM MQ-Regeln für die Einstellung von MQMD-Benutzer-IDs festgelegt.	Kein Standardwert

Tabelle 82. Erweiterte Agenteneigenschaften: Ressourcenüberwachung

Eigenschaftsname	Beschreibung	Standardwert
monitorFilepathPlatformSeparator	Gibt an, ob in der Variablen <code>\$FILEPATH</code> plattformspezifische Pfadtrennzeichen verwendet werden sollen. Der Wert <code>true</code> bedeutet, dass plattformspezifische Pfadtrennzeichen verwendet werden. Bei Angabe von <code>false</code> wird auf allen Plattformen der UNIX-Schrägstrich (<code>/</code>) als Pfadtrennzeichen verwendet.	true
monitorMaxResourcesInPoll	<p>Gibt die maximale Anzahl an überwachten Ressourcen an, die in den einzelnen Abfrageintervallen ausgelöst werden sollen. Wenn Sie beispielsweise das Überwachungsmuster <code>*.txt</code> sowie ein Abfrageintervall von zehn Sekunden angeben und die Eigenschaft 'monitorMaxResourcesInPoll' auf '10' setzen, beschränkt die Eigenschaft 'monitorMaxResourcesInPoll' den Agenten dahingehend, dass er in jedem Abfrageintervall höchstens für zehn Übereinstimmungen eine Auslösung vornimmt. Übereinstimmende Ressourcen, die über den Grenzwert 10 hinausgehen, werden in späteren Abfrageintervallen ausgelöst.</p> <p>Sie können die Eigenschaft 'monitorMaxResourcesInPoll' außerdem mit einem entsprechenden Wert für den Parameter -bs im Befehl fteCreateMonitor kombinieren, um beispielsweise jedes Abfrageintervall auf die Auslösung von nur einer Übertragung zu beschränken.</p> <p>Ein Wert kleiner-gleich null bedeutet, dass die Anzahl der Überwachungsressourcen, die in einem Abfrageintervall ausgelöst werden, unbegrenzt ist.</p>	-1
monitorReportTriggerFail	Gibt an, ob Fehlerbedingungen in der Umgebung oder in der Konfiguration, die bei der Überwachung erkannt werden, als Protokollnachricht an den Abschnitt <code>SYSTEM.FTE</code> gemeldet werden. Bei Angabe von <code>true</code> werden Nachrichten protokolliert, bei Angabe von <code>false</code> werden keine Nachrichten protokolliert.	true

Tabelle 82. Erweiterte Agenteneigenschaften: Ressourcenüberwachung (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
monitorReportTriggerNotSatisfied	Gibt an, ob bei einem nicht erfüllten Auslöser eine Protokollnachricht an den Abschnitt SYSTEM.FTE gesendet wird, in dem sich die Details befinden. Bei Angabe von <code>true</code> werden Nachrichten protokolliert, bei Angabe von <code>false</code> werden keine Nachrichten protokolliert.	false
monitorReportTriggerSatisfied	Gibt an, ob bei einem erfüllten Auslöser eine Protokollnachricht an den Abschnitt SYSTEM.FTE gesendet wird, in dem sich die Details befinden. Bei Angabe von <code>true</code> werden Nachrichten protokolliert, bei Angabe von <code>false</code> werden keine Nachrichten protokolliert.	false
monitorSilenceOnTriggerFailure	Gibt an, nach wie vielen aufeinanderfolgenden Fehlern des Ressourcenüberwachungsauslösers die Fehler nicht mehr gemeldet werden.	5
monitorStopOnInternalFailure	Gibt an, nach wie vielen aufeinanderfolgenden internen FFDC-Bedingungen der Ressourcenüberwachung der Status der Überwachung in 'gestoppt' geändert wird.	10

Tabelle 83. Erweiterte Agenteneigenschaften: Stammverzeichnis

Eigenschaftsname	Beschreibung	Standardwert
commandPath	<p>Gibt die Pfade an, über die Befehle aufgerufen werden können:</p> <ul style="list-style-type: none"> Über die Ant-Tasks <code>fte:call Ant</code>, <code>fte:filecopy</code> oder <code>fte:filemove</code> des Agenten Unter Verwendung eines der unterstützten Managed File Transfer Agent-Befehls-XML-Schemas (beispielsweise 'managedCall' oder 'managedTransfer') in einer XML-Nachricht, die an einen Agenten übergeben wird. <p>Informationen zur gültigen Syntax des Werts der Eigenschaft 'commandPath' finden Sie im Abschnitt MFT-Eigenschaft 'commandPath'.</p> <p>Wichtig: Legen Sie diese Eigenschaft mit äußerster Vorsicht fest, da jeder Befehl in einem der angegebenen commandPathen effektiv von einem fernen Clientsystem aufgerufen werden kann, das Befehle an den Agenten senden kann. Aus diesem Grund gilt standardmäßig Folgendes, wenn Sie einen commandPath angeben:</p> <ul style="list-style-type: none"> Alle vorhandenen Agentensandboxes werden vom Agenten beim Start konfiguriert, sodass alle commandPath-Verzeichnisse automatisch zur Liste der Verzeichnisse hinzugefügt werden, die den Zugriff auf eine Übertragung verweigert haben. Alle vorhandenen Benutzersandboxes werden beim Start des Agenten aktualisiert, sodass alle commandPath-Verzeichnisse (und ihre Unterverzeichnisse) als <code><exclude></code>-Elemente zu den Elementen <code><read></code> und <code><write></code> hinzugefügt werden. Wenn der Agent nicht für die Verwendung einer Agentensandbox oder einer Benutzersandbox konfiguriert ist, wird beim Start des Agenten eine neue Agentensandbox erstellt, in der die Verzeichnisse commandPath als verweigerter Verzeichnisse angegeben sind. <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.</p> <p>Sie können die Eigenschaft addCommandPathToSandbox auf "false" setzen, um dieses Standardverhalten außer Kraft zu setzen.</p> <p>Wichtig: Es ist jedoch zu beachten, dass diese Überschreibung es einem Client ermöglicht, jeden beliebigen Befehl an das Agentensystem zu übertragen und aufzurufen. Deshalb sollte sie mit Vorsicht eingesetzt werden.</p>	Es können keine Befehle aufgerufen werden
addCommandPathToSandbox	<p>Gibt an, ob die mit der Eigenschaft commandPath angegebenen Verzeichnisse (und alle zugehörigen Unterverzeichnisse) hinzugefügt werden sollen:</p> <ul style="list-style-type: none"> den gesperrten Verzeichnissen für eine vorhandene Agentensandbox Die <code><exclude></code>-Elemente für die Elemente <code><read></code> und <code><write></code> für alle definierten Benutzersandboxes. einer neuen Agentensandbox, wenn ein Agent weder mit einer Agentensandbox noch mit einer Benutzersandbox konfiguriert wurde <p>Informationen zur gültigen Syntax des Werts der Eigenschaft commandPath finden Sie unter commandPath MFT-Eigenschaft.</p>	True

Tabelle 83. Erweiterte Agenteneigenschaften: Stammverzeichnis (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
additionalWildcardSandboxChecking	<p>Gibt an, ob für einen Agenten, der mit einer Benutzer- oder Agentensandbox konfiguriert wurde, zusätzliche Überprüfungen für Platzhalterübertragungen durchgeführt werden sollen, um die Positionen zu beschränken, an die und von denen der Agent Dateien übertragen kann.</p> <p>Ist diese Eigenschaft auf 'true' gesetzt, ist die zusätzliche Überprüfung aktiviert. Versucht eine Übertragungsanforderung, eine Position zu lesen, die außerhalb der definierten Sandbox für Dateiabgleich mit dem Platzhalter liegt, schlägt die Übertragung fehl. Wenn eine Übertragungsanforderung aus mehreren Übertragungen besteht und eine dieser Übertragungen fehlschlägt, weil sie versucht, eine Position außerhalb der Sandbox zu lesen, schlägt die gesamte Übertragung fehl. Schlägt die Überprüfung fehl, wird die Ursache dafür in einer Fehlermeldung angezeigt (siehe Zusätzliche Überprüfungen für Platzhalterübertragungen).</p> <p>Wird die Eigenschaft weggelassen oder auf 'false' gesetzt, werden keine zusätzlichen Überprüfungen für Platzhalterübertragungen durchgeführt.</p>	--
sandboxRoot	<p>Gibt eine Reihe von Stammverzeichnispfaden an, die bei Aktivierung der Sandbox-Funktion berücksichtigt bzw. nicht berücksichtigt werden sollen. Informationen zu dieser Funktion finden Sie im Abschnitt Mit Sandboxes für den MFT-Agenten arbeiten.</p> <p>Pfadnamen müssen mit einem plattformspezifischen Pfadtrennzeichen getrennt werden. Stellen Sie den Pfadangaben ein Ausrufezeichen (!) voran, um anzugeben, dass diese Pfade aus der Sandbox ausgeschlossen werden sollen. Dies ist hilfreich, wenn ein Unterverzeichnis in einem einbezogenen Stammverzeichnispfad ausgeschlossen werden soll.</p> <p>Die Eigenschaft 'sandboxRoot' wird auf Protokollbridgeagenten nicht unterstützt.</p> <p>Die Eigenschaften 'sandboxRoot' und 'userSandboxes' können nicht gemeinsam angegeben werden.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.</p>	-- - keine Sandbox
transferRoot	<p>Standardmäßiges Stammverzeichnis für relative Pfade, die dem Agenten angegeben werden.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.</p>	Das Ausgangsverzeichnis für den Benutzer, der den Agentenprozess gestartet hat.
transferRootHLQ	<p>Standardmäßiges Qualifikationsmerkmal der höheren Ebene (Benutzer-ID) für nicht vollständig qualifizierende Datengruppen, die dem Agenten angegeben werden</p>	Der Benutzername des Benutzers, der den Agentenprozess gestartet hat.
userSandboxes	<p>Schränkt den Bereich des Dateisystems, in den und aus dem Dateien übertragen werden können, auf Grundlage des MQMD-Benutzernamens des Benutzers ein, der die Übertragung anfordert. Weitere Informationen finden Sie im Abschnitt Mit Sandboxes für MFT-Benutzer arbeiten.</p> <p>Die Eigenschaft 'userSandboxes' wird auf Protokollbridgeagenten nicht unterstützt.</p> <p>Die Eigenschaften 'sandboxRoot' und 'userSandboxes' können nicht gemeinsam angegeben werden.</p>	false

Tabelle 84. Erweiterte Agenteneigenschaften: Scheduler-Eigenschaft

Eigenschaftsname	Beschreibung	Standardwert
maxSchedulerRunDelay	<p>Das maximale Intervall in Minuten, das der Agent abwartet, bis er prüft, ob geplante Übertragungen vorhanden sind. Wenn Sie diese Eigenschaft aktivieren möchten, geben Sie eine positive Ganzzahl an. Im Abschnitt Vorgehensweise, wenn die geplante Dateiübertragung nicht oder verzögert ausgeführt wird finden Sie weitere Informationen darüber, ob diese Eigenschaft für Sie sinnvoll sein könnte.</p> <p>Da der Agent bei der Fälligkeit geplanter Übertragungsausführungen möglicherweise gerade einen Befehl aus seiner Befehlswarteschlange liest, kann sich der Start der geplanten Übertragungen unter Umständen weiter verzögern. In diesem Fall wird der Scheduler unmittelbar nach Abschluss des betreffenden Befehls ausgeführt.</p>	-1







Tabelle 85. Erweiterte Agenteneigenschaften: Sicherheit		
Eigenschaftsname	Beschreibung	Standardwert
agentCredentialsKeyFile	Der Name der Datei, die den Berechtigungsnachweisschlüssel enthält, der beim Verschlüsseln von Berechtigungsnachweisen verwendet wird.	Eine Zeichenfolgeeigenschaft, die keinen Standardwert hat.
agentQMGrAuthenticationCredentialsFile	<p>Der Pfad zu der Datei, die die Berechtigungsnachweise enthält, die beim Herstellen der Verbindung zum Agentenwarteschlangenmanager verwendet werden sollen.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.</p> <p> Details zum Erstellen der Berechtigungsnachweisdateien finden Sie unter MFT und IBM MQ Verbindungsauthentifizierung.</p> <p> Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter z/OSkonfigurieren.</p>	<p>Der Standardwert für diese Eigenschaft ist:</p> <p> z/OS</p> <p> Linux</p> <p> AIX \$HOME/MQMFTCredentials.xml</p> <p> Windows %USERPROFILE%/MQMFTCredentials.xml</p>
authorityChecking	<p>Legt fest, ob die im Abschnitt <u>Benutzerberechtigungen für MFT-Agentenaktionen beschränken</u> beschriebenen Sicherheitsfunktionen aktiviert sind.</p> <p>Die Berechtigung <code>inquire</code> ist eine erforderliche Berechtigung für alle Berechtigungwarteschlangen des Agenten.</p>	false
logAuthorityChecks	Die Protokollstufe für die Berechtigungsprüfung, die im Ereignisprotokoll des Agenten in der Datei <code>output0.log</code> aufgezeichnet wird. Mögliche Werte für diese Eigenschaft sind <code>None</code> (Keine), <code>Failures</code> (Fehler) oder <code>All</code> (Alle).	--
userIdForClientConnect	Die Benutzer-ID, die über Clientverbindungen an IBM MQ weitergegeben wird. Bei Angabe von <code>java</code> wird der von der JVM gemeldete Benutzername als Teil der IBM MQ-Verbindungsanforderung weitergegeben. Mögliche Werte für diese Eigenschaft sind <code>None</code> (Keine) oder <code>java</code> .	--


Tabelle 86. Erweiterte Agenteneigenschaften: SSL/TLS		
Eigenschaftsname	Beschreibung	Standardwert
agentSslCipherSpec	<p>Gibt das Protokoll, den Hashalgorithmus und den Verschlüsselungsalgorithmus an, die verwendet werden. Gibt außerdem an, wie viele Bit im Verschlüsselungsschlüssel verwendet werden, wenn Daten zwischen dem Agenten und dem Warteschlangenmanager des Agenten ausgetauscht werden.</p> <p>Der Wert von 'agentSslCipherSpec' ist ein CipherSpec-Name. Dieser CipherSpec-Name ist mit dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen identisch. Eine Liste gültiger CipherSpec-Namen ist in den Abschnitten <u>SSL/TLS-CipherSpecs</u> und <u>-CipherSuites</u> in IBM MQ-Klassen für Java und <u>SSL/TLS-CipherSpecs</u> und <u>-CipherSuites</u> in IBM MQ-Klassen für JMS zu finden.</p> <p>'agentSslCipherSpec' entspricht weitgehend 'agentSslCipherSuite'. Wenn sowohl 'agentSslCipherSuite' als auch 'agentSslCipherSpec' angegeben ist, wird der Wert von 'agentSslCipherSpec' verwendet.</p>	--
agentSslCipherSuite	<p>Gibt SSL-Aspekte zum Austausch von Daten zwischen dem Agenten und dem Warteschlangenmanager des Agenten an.</p> <p>Beim Wert von 'agentSslCipherSuite' handelt es sich um einen CipherSuite-Namen. Dieser CipherSuite-Name ist dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen zugeordnet. Weitere Informationen hierzu finden Sie im Abschnitt <u>Namenszuordnungen von CipherSuites und CipherSpecs</u>.</p> <p>'agentSslCipherSuite' entspricht weitgehend 'agentSslCipherSpec'. Wenn sowohl 'agentSslCipherSuite' als auch 'agentSslCipherSpec' angegeben ist, wird der Wert von 'agentSslCipherSpec' verwendet.</p>	--
agentSslPeerName	Gibt den Entwurf eines definierten Namens an, der mit dem vom Warteschlangenmanager des Agenten bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Warteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.	--
agentSslTrustStore	<p>Gibt die Position der Zertifikate an, die der Agent akzeptiert. Beim Wert von 'agentSslTrustStore' handelt es sich um einen Dateipfad. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden.  Windows</p> <p>Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p>	--

Tabelle 86. Erweiterte Agenteneigenschaften: SSL/TLS (Forts.)














Eigenschaftsname	Beschreibung	Standardwert
agentSslKeyStore	Gibt die Position des privaten Schlüssels des Agenten an. Beim Wert von 'agentSslKeyStore' handelt es sich um einen Dateipfad. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden.  Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Diese Eigenschaft muss nur angegeben werden, wenn die Clientauthentifizierung für den Warteschlangenmanager des Agenten erforderlich ist. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	--
agentSslFipsRequired	Gibt an, dass Sie die FIPS-Unterstützung auf Agentenebene aktivieren möchten. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> . Weitere Informationen finden Sie im Abschnitt FIPS-Unterstützung in MFT .	false
agentSslKeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann <code>jks</code> oder <code>pkcs12</code> angegeben werden.	jks
agentSslKeyStoreCredentialsFile	Der Pfad zu der Datei, die die Berechtigungsnachweise für den Zugriff auf den Keystore des Agenten enthält Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.  Details zum Erstellen der Berechtigungsnachweisdateien finden Sie unter MFT und IBM MQ Verbindungsauthentifizierung .  Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter z/OS konfigurieren .	Der Standardwert für diese Eigenschaft ist:  <code>z/OS</code>  <code>Linux</code>  <code>AIX</code> <code>\$HOME/MQMFTCredentials.xml</code>  <code>Windows</code> <code>%USERPROFILE%/MQMFTCredentials.xml</code>
agentSslTrustStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann <code>jks</code> oder <code>pkcs12</code> angegeben werden.	jks
agentSslTrustStoreCredentialsFile	Der Pfad zu der Datei mit den Berechtigungsnachweisen für den Zugriff auf den Truststore des Agenten. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.  Details zum Erstellen der Berechtigungsnachweisdateien finden Sie unter MFT und IBM MQ Verbindungsauthentifizierung .  Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter z/OS konfigurieren .	Der Standardwert für diese Eigenschaft ist:  <code>z/OS</code>  <code>Linux</code>  <code>AIX</code> <code>\$HOME/MQMFTCredentials.xml</code>  <code>Windows</code> <code>%USERPROFILE%/MQMFTCredentials.xml</code>

Tabelle 87. Erweiterte Agenteneigenschaften: Zeitlimit

Eigenschaftsname	Beschreibung	Standardwert
maxTransferNegotiationTime	Die maximale Zeit in Millisekunden, die eine Übertragung auf den Abschluss der Verhandlung durch den Zielagenten wartet. Wenn die Verhandlung nicht innerhalb dieser Zeit abgeschlossen wird, wird die Übertragung in einen Resynchronisationsstatus versetzt. Dadurch kann die nächste Übertragung, sofern vorhanden, ausgeführt werden. In Szenarios mit einer hohen Auslastung des Quellen- oder Zielagenten ist es möglich, dass der Standardwert zu niedrig ist, sodass der Agent nicht schnell genug auf die Verhandlungsanfrage antworten kann. Dies gilt vor allem dann, wenn für einen Quellenagenten sehr viele Ressourcenmonitore definiert sind oder wenn seine Ressourcenmonitore Verzeichnisse überwachen, die sehr viele Dateien enthalten. Es kann aber auch auftreten, wenn sehr viele Übertragungsanforderungen an einen Agenten übergeben werden. In solchen Szenarios kann es erforderlich sein, den Wert dieser Eigenschaft auf 200.000 oder mehr zu erhöhen.	30 000
recoverableTransferRetryInterval	Die Wartezeit in Millisekunden zwischen der Erkennung eines behebbaren Übertragungsfehlers und dem Versuch, die Übertragung wiederaufzunehmen.	60000
senderTransferRetryInterval	Die Zeit in Millisekunden, nach der eine Übertragung wiederholt wird, die abgelehnt wurde, weil am Ziel bereits die maximale Anzahl an Übertragungen ausgeführt wird. Der Mindestwert ist 1000.	30 000

Tabella 87. Erweiterte Agenteneigenschaften: Zeitlimit (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
transferAckTimeout	<p>Die Zeit in Millisekunden die eine Übertragung auf eine Bestätigung oder auf Daten der anderen Seite wartet, bevor ein erneuter Versuch stattfindet. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>Sobald ein vollständiges Datenfenster empfangen wird, werden vom empfangenden Agenten Bestätigungen an den sendenden Agenten gesendet. Bei Netzen mit einer Bandbreitenbeschränkung oder bei störanfälligen Netzen und hohen Einstellungen für 'agentWindowSize' und 'agentChunkSize' kann es vorkommen, dass der Standardwert nicht hoch genug ist. Dies kann zu einer unnötigen erneuten Übertragung von Daten zwischen den Agenten führen. Eine Erhöhung dieses Werts kann daher vorteilhaft sein und die Wahrscheinlichkeit verringern, dass eine Übertragung aufgrund eines langsamen Netzes in den Wiederherstellungsmodus wechselt.</p>	60000
transferAckTimeoutRetries	Maximale Anzahl an Bestätigungswiederholungen für eine Übertragung ohne eine Antwort, bevor der Agent aufgibt und die Übertragung in den Zurückschreibungsmodus versetzt.	5
xmlConfigReloadInterval	<p>Das Intervall in Sekunden, das während der Laufzeit zwischen dem erneuten Laden der XML-Konfigurationsdateien durch den Agenten liegt. Wenn Sie verhindern möchten, dass der Agent XML-Konfigurationsdateien während der Laufzeit erneut lädt, müssen Sie diese Eigenschaft auf -1 setzen. Folgende XML-Konfigurationsdateien sind von dieser Eigenschaft betroffen:</p> <ul style="list-style-type: none"> • ConnectDirectCredentials.xml • ConnectDirectNodeProperties.xml • ConnectDirectProcessDefinitions.xml • ProtocolBridgeCredentials.xml • ProtocolBridgeProperties.xml • UserSandboxes.xml 	30

Tabella 88. Erweiterte Agenteneigenschaften: Traceerstellung und Protokollierung

Eigenschaftsname	Beschreibung	Standardwert
javaCoreTriggerFile	<p>Der vollständige Pfad zu einer Datei position, die der Agent überwacht. Wenn die Datei an der angegebenen Position vorhanden ist, löst der Agentenstart eine Javacore-Datei aus. Wenn Sie nach dem Agentenstart eine Datei an dieser Position aktualisieren, löst der Agent erneut eine Javacore-Datei aus.</p> <p>Ein separater Thread führt für diese Datei alle 30 Sekunden eine Abfrage durch, um zu prüfen, ob die Datei erstellt oder aktualisiert wurde. Wenn die Datei seit dem letzten Polling erstellt oder aktualisiert wurde, generiert der Agent eine Java-Core-Dump-Datei im folgenden Verzeichnis: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code></p> <p>Wenn Sie diese Eigenschaft angeben, gibt der Agent beim Start die folgende Nachricht aus:</p> <pre>BFGAG0092I The <insert_0> file will be used to request JVM diagnostic information.</pre> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174.</p>	--

Tabella 88. Erweiterte Agenteneigenschaften: Tracerstellung und Protokollierung (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
trace	<p>Die beim Start des Agenten zu verwendende Tracespezifikation. Dies ist eine durch Kommas getrennte Liste mit Klassen und/oder Paketen, dem Gleichheitszeichen und einer Tracestufe.</p> <p>Wenn Sie beispielsweise die Klasse <code>com.ibm.wmqfte.agent.Agent</code> und die Klassen im Paket <code>com.ibm.wmqfte.commandhandler</code> ab dem Start des Agenten verfolgen möchten, fügen Sie der Datei <code>agent.properties</code> den folgenden Eintrag hinzu:</p> <pre>trace=com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.com mandhandler=all</pre> <p>Sie können auch mehrere Tracespezifikationen in einer durch Doppelpunkte getrennten Liste angeben. Beispiel:</p> <pre>tra ce=com.ibm.wmqfte.agent.Agent=all:com.ibm.wmqfte.com mandhandler=moderate</pre> <p>Die spezielle Tracespezifikation <code>=all</code> wird verwendet, um einen Trace für den Agenten und die Java Message Queuing Interface (JMQUI) zu erstellen, die die gesamte Kommunikation mit dem Agentenwarteschlangenmanager ausführen. Fügen Sie dazu den folgenden Eintrag zur Datei <code>agent.properties</code> hinzu:</p> <pre>trace==all</pre> <p>Sofern von Ihrem IBM Ansprechpartner nicht anders angegeben, verwenden Sie die Tracespezifikation <code>com.ibm.wmqfte=all</code> wie folgt:</p> <pre>trace=com.ibm.wmqfte=all</pre>	--
outputLogFiles	Die Gesamtzahl der aufzubewahrenden Dateien des Typs <code>output.log</code> . Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	5
outputLogSize	Die maximale Größe in MB, die jede Datei <code>output.log</code> erreichen kann, bevor die Ausgabe in die nächste Datei geleitet wird. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	1
outputLogEncoding	Die Zeichencodierung, die der Agent beim Schreiben von Daten in die Datei <code>output.log</code> verwendet.	Die Standardzeichencodierung der Plattform, auf welcher der Agent betrieben wird.
traceFiles	Die Gesamtanzahl der zu speichernden Tracedateien. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	5
traceSize	Die maximale Größe (in MB) der einzelnen Tracedateien; sobald diese Größe erreicht ist, wird der Trace in eine Folgedatei geschrieben. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	20
traceMaxBytes	Die maximale Menge der Nachrichtendaten, die in die Tracedatei ausgegeben wird.	4096 Byte
logTransferRecovery	Wenn diese Eigenschaft auf den Wert 'true' gesetzt wird, werden Diagnoseereignisse im Ereignisprotokoll des Agenten in der Datei <code>output0.log</code> gemeldet, sobald eine Übertragung in den Wiederherstellungsstatus versetzt wird.	Der Standardwert ist "true".
logCapture	Erfasst Übertragungsanforderungsnachrichten, die an diesen Agenten übergeben werden, und Protokollnachrichten, die vom Agenten an den Koordinationswarteschlangenmanager veröffentlicht werden. Die erfassten Nachrichten können bei der Behebung von Übertragungsfehlern hilfreich sein. Erfasste Nachrichten werden in Dateien im Agentenprotokollverzeichnis <code>capture?.log</code> gespeichert. Das Symbol ? steht für einen numerischen Wert. Die Datei mit der Nummer 0enthält die neuesten erfassten Nachrichten.	false
logCaptureFileSize	Gibt die maximale Größe einer Aufzeichnungsdatei in Megabytes an.	10
logCaptureFiles	Gibt an, wie viele Aufzeichnungsdateien maximal aufbewahrt werden, bevor die älteste Datei gelöscht wird.	10
logCaptureFilter	Ein regulärer Java-Ausdruck, den der Agent zum Abgleichen des Themennamens der Nachricht verwendet. Es werden nur die dem regulären Ausdruck entsprechenden Nachrichten erfasst.	.* (alle abgleichen)

Tabelle 88. Erweiterte Agenteneigenschaften: Tracerstellung und Protokollierung (Forts.)


Eigenschaftsname	Beschreibung	Standardwert
resourceMonitorLog	<p>Schlüssel/Wert-Paar für Ressourcenmonitor und Betrieb zum Aktivieren bzw. Inaktivieren der Protokollierung.</p> <p>Folgende Werte sind möglich:</p> <ul style="list-style-type: none"> • info • moderate • verbose • off <p>For example:</p> <ul style="list-style-type: none"> • resourceMonitorLog=MON1,MON2=info:MON3=off <p>Aktivieren Sie die Protokollierung für MON1 und MON2 und inaktivieren Sie die Protokollierung für MON3.</p> <ul style="list-style-type: none"> • resourceMonitorLog=info <p>Aktivieren Sie die Protokollierung auf info-Ebene für alle Ressourcenmonitore.</p> <p>Die Ressourcenüberwachungsprotokolle werden in eine Datei mit dem Namen resmoneventN.log geschrieben, wobei N für eine Zahl steht, z. B. resmonevent0.log.</p> <p> Achtung: Alle Ressourcenmonitore eines Agenten schreiben in dieselbe Protokolldatei.</p> <p>Weitere Informationen finden Sie im Abschnitt Protokollierung der MFT-Ressourcenüberwachung.</p>	info
resourceMonitorLogFileSize	Gibt die maximale Größe einer Aufzeichnungsdatei in Megabytes an.	20
resourceMonitorLogFiles	<p>Gibt an, wie viele Aufzeichnungsdateien maximal aufbewahrt werden, bevor die älteste Datei gelöscht wird.</p> <ul style="list-style-type: none"> • Der Standardwert der Agenteneigenschaft resourceMonitorLogFiles wurde von 10 in 5 geändert. Dies bedeutet, dass ab IBM MQ 9.3.0, wenn der Standardwert festgelegt ist, maximal fünf Ereignisprotokolldateien des Ressourcenmonitors vorhanden sein können, beginnend von resmonevent0.log bis resmonevent4.log. Sie können diesen Wert jedoch bei Bedarf ändern. • Falls der Agent aus einer Version vor IBM MQ 9.3.0 migriert wird, sollten Sie die Dateien resmonevent5.log bis resmonevent9.log manuell löschen, falls sie vorhanden sind. • Die Größe der einzelnen Protokolldateien beträgt jedoch weiterhin 20 MB. 	Ab IBM MQ 9.3.0 ist der Standardwert 5.

Tabelle 89. Erweiterte Agenteneigenschaften: Übertragungsgrenzwerte

Eigenschaftsname	Beschreibung	Standardwert
maxDestinationTransfers	<p>Die maximale Anzahl gleichzeitig ablaufender Übertragungen, die der Zielagent zu einem bestimmten Zeitpunkt verarbeitet. Jede an einen Agenten übergebene Übertragungsanforderung wird auf diese Gesamtzahl angerechnet, unabhängig von der Anzahl der Dateien, die zum Erfüllen der Anforderung übertragen werden. Das bedeutet, dass eine Übertragungsanforderung, in der eine einzelne Datei übertragen wird, ebenso gezählt wird wie eine Übertragungsanforderung, in der 10 Dateien übertragen werden.</p> <p>Wenn der Zielagent das durch die Eigenschaft 'maxDestinationTransfers' angegebene Limit erreicht hat, stellt er Übertragungen in Warteschlangen.</p> <p>Falls die Summe der Werte für die Agenteneigenschaften 'maxSourceTransfers', 'maxDestinationTransfers' und 'maxQueuedTransfers' den Wert der Einstellung MAXDEPTH der Statusspeicherwarteschlange (SYSTEM.STATE.Agentenname) überschreitet, wird der Agent nicht gestartet.</p>	<p>25 (bei allen Agenten außer Connect:Direct)</p> <p>5 (bei Connect:Direct-Bridgeagenten)</p>
maxFilesForTransfer	<p>Die maximale Anzahl der Übertragungselemente, die für eine einzelne verwaltete Übertragung zulässig sind. Wenn eine Übertragung mehr Elemente enthält, als im Wert 'maxFilesForTransfer' angegeben sind, schlägt die verwaltete Übertragung fehl und es werden keine Übertragungselemente verarbeitet.</p> <p>Das Festlegen dieser Eigenschaft verhindert, dass Sie versehentlich zu viele Dateien übertragen, weil eine fehlerhafte Übertragungsanforderung vorliegt, z. B. wenn ein Benutzer versehentlich die Übertragung des Stammverzeichnisses/im Schlüsselwort conref=" ../common/mqent.dita#mqent/unixlinuxbis " /> angibt.</p>	5000

Tabelle 89. Erweiterte Agenteneigenschaften: Übertragungsgrenzwerte (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
maxSourceTransfers	<p>Die maximale Anzahl gleichzeitig ablaufender Übertragungen, die der Quellenagent zu einem bestimmten Zeitpunkt verarbeitet. Jede an einen Agenten übergebene Übertragungsanforderung wird auf diese Gesamtzahl angerechnet, unabhängig von der Anzahl der Dateien, die zum Erfüllen der Anforderung übertragen werden. Das bedeutet, dass eine Übertragungsanforderung, in der eine einzelne Datei übertragen wird, ebenso gezählt wird wie eine Übertragungsanforderung, in der 10 Dateien übertragen werden.</p> <p>Wenn der Quellenagent das durch die Eigenschaft 'maxSourceTransfers' angegebene Limit erreicht hat, stellt er Übertragungen in Warteschlangen.</p> <p>Falls die Summe der Werte für die Agenteneigenschaften 'maxSourceTransfers', 'maxDestinationTransfers' und 'maxQueuedTransfers' den Wert der Einstellung MAXDEPTH der Statusspeicherwarteschlange (SYSTEM.FTE.STATE.Agentenname) überschreitet, wird der Agent nicht gestartet.</p>	<p>25 (bei allen Agenten außer Connect:Direct-Bridgeagenten)</p> <p>5 (bei Connect:Direct-Bridgeagenten)</p>
maxQueuedTransfers	<p>Die maximale Anzahl anstehender Übertragungen, die von einem Quellenagent in die Warteschlange gestellt werden können, bis der Agent eine neue Übertragungsanforderung zurückweist. Sie können diese Eigenschaft so einstellen, dass trotz einer Überschreitung der Grenzwerte von 'maxDestinationTransfers' und 'maxSourceTransfers' neue Übertragungsanforderungen, die Sie jetzt stellen, akzeptiert, in eine Warteschlange eingereiht und dann ausgeführt werden.</p> <p>Die Reihenfolge, in der die Übertragungsanforderungen aus der Warteschlange ausgeführt werden, wird durch deren Priorität und die Dauer bestimmt, die sie sich in der Warteschlange befinden. Ältere Anforderungen und Anforderungen mit hoher Priorität werden vorrangig verarbeitet. Übertragungen mit niedriger Priorität, die sich bereits lange Zeit in der Warteschlange befinden, werden vor neueren Übertragungen mit höherer Priorität ausgewählt.</p> <p>Falls die Summe der Werte für die Agenteneigenschaften 'maxSourceTransfers', 'maxDestinationTransfers' und 'maxQueuedTransfers' den Wert der Einstellung MAXDEPTH der Statusspeicherwarteschlange (SYSTEM.FTE.STATE.Agentenname) überschreitet, wird der Agent nicht gestartet.</p>	1000

Tabelle 90. Erweiterte Agenteneigenschaften: Zeitlimit für Übertragungswiederherstellung		
Eigenschaftsname	Beschreibung	Standardwert
transferRecoveryTimeout	<p>Geben Sie (in Sekunden) an, wie lange ein Quellenagent versuchen soll, eine blockierte Dateiübertragung wiederherzustellen.</p> <p>Wird diese Eigenschaft nicht gesetzt, wird der Agent standardmäßig den Versuch so lange wiederholen, bis die Übertragung erfolgreich wiederhergestellt werden konnte. Sie können für die Eigenschaft zum Festlegen des Zeitlimits für die Übertragungswiederherstellung die folgenden Werte angeben:</p> <p>-1 Der Agent wiederholt den Versuch, die blockierte Übertragung wiederherzustellen, bis die Übertragung abgeschlossen wurde. Diese Option entspricht dem Standardverhalten des Agenten, wenn die Eigenschaft nicht gesetzt wird.</p> <p>0 Der Agent stoppt die Dateiübertragung, sobald die Wiederherstellung einsetzt.</p> <p>>0 Der Agent wiederholt den Versuch, die blockierte Übertragung wiederherzustellen, bis der in Form eines positiven Integerwerts angegebene Zeitraum (in Sekunden) abgelaufen ist. Beispiel: <code>transferRecoveryTimeout=21600</code> gibt an, dass der Agent seit dem Eintritt in die Wiederherstellung 6 Stunden lang versucht, die Übertragung wiederherzustellen. Der Maximalwert für diesen Parameter ist 999999999.</p>	-1

Tabelle 91. Erweiterte Agenteneigenschaften: Benutzerexitroutine		
Eigenschaftsname	Beschreibung	Standardwert
agentForceConsistentPathDelimiters	Als Pfadtrennzeichen wird in den Informationen zur Quellen- und Zieldatei, die in den Übertragungsexits bereitgestellt werden, der UNIX-Schrägstrich (/) verwendet. Gültige Optionen sind <code>true</code> und <code>false</code> .	false
destinationTransferEndExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine zum Zielübertragungsende implementieren.	Kein Standardwert
destinationTransferStartExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine zum Zielübertragungsstart implementieren.	Kein Standardwert

Tabelle 91. Erweiterte Agenteneigenschaften: Benutzerexitroutine (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
exitClassPath	Gibt eine plattformspezifische, von Zeichen begrenzte Liste von Verzeichnissen an, die als Klassenpfad für Benutzerexitroutinen agieren. Das Exitverzeichnis des Agenten wird vor allen Einträgen in diesem Klassenpfad durchsucht.	Verzeichnis exits des Agenten
exitNativeLibraryPath	Gibt eine plattformspezifische, von Zeichen begrenzte Liste von Verzeichnissen an, die als Pfad der nativen Bibliothek für Benutzerexitroutinen agieren.	Verzeichnis exits des Agenten
ioMaxRecordLength	Die maximale Satzlänge in Byte, die für eine satzorientierte Datei unterstützt werden kann. Managed File Transfer kann das Schreiben in satzorientierte Dateien mit beliebiger Satzlänge unterstützen. Da lange Datensätze jedoch zu Fehlern aufgrund abnormaler Speicherbedingungen führen können, wird die maximale Satzlänge zur Vermeidung derartiger Fehler standardmäßig auf 64 K beschränkt. Beim Auslesen satzorientierter Dateien muss ein ganzer Datensatz in einen einzelnen Übertragungsblock passen. Daher ist die Satzlänge zusätzlich durch die Größe des Übertragungsblocks beschränkt. Diese Eigenschaft wird nur für satzorientierte Dateien des Ein-/Ausgabebenutzers verwendet.	64 KB
monitorExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Übertragungsexitroutine implementieren. Weitere Informationen finden Sie im Abschnitt Benutzerexits für MFT-Ressourcenüberwachung .	Kein Standardwert
protocolBridgeCredentialExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine für Protokollbrückeberechtigungsanzeige implementieren. Weitere Informationen finden Sie im Abschnitt Berechtigungsanzeige für einen Dateiserver mittels Exitklassen zuordnen .	Kein Standardwert.
sourceTransferEndExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Exitroutine zum Quellenübertragungsende implementieren.	Kein Standardwert
sourceTransferStartExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Exitroutine zum Quellenübertragungsstart implementieren.	Kein Standardwert
IOExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine für Ein-/Ausgaben implementieren. Listen Sie nur die Klassen auf, die die Schnittstelle 'IOExit' implementieren; listen Sie also keine Klassen auf, die die anderen Benutzerexits für Ein-/Ausgaben implementieren (beispielsweise IOExitResourcePath und IOExitChannel). Weitere Informationen finden Sie im Abschnitt MFT-Übertragungs-E/A-Benutzerexits verwenden .	Kein Standardwert.

Tabelle 92. Erweiterte Agenteneigenschaften: IBM MQ-Clientkomprimierung		
Eigenschaftsname	Beschreibung	Standardwert
agentDataCompression	Diese Eigenschaft wird nur für Clientverbindungen unterstützt. Eine durch Kommas getrennte Liste mit Komprimierungstypen für die Übertragung von Dateidaten, über die mit dem fernen IBM MQ-Server verhandelt wird. Informationen zu diesen Komprimierungstypen finden Sie im folgenden Abschnitt: Message data compression list (Liste zur Nachrichtendatenkomprimierung). Die Werte werden auf ihre Gültigkeit überprüft und anschließend in der angegebenen Reihenfolge als Eigenschaften an den Clientkanal des Agenten übergeben. Der IBM MQ-Client verarbeitet nun Vereinbarungen zwischen diesem Clientkanal und dem fernen Serverkanal, um den kleinsten gemeinsamen Nenner der Komprimierungseigenschaften auf den beiden Kanälen zu finden. Wenn es keine Übereinstimmung gibt, wird immer MQCOMPRESS_NONE ausgewählt.	MQCOMPRESS_NONE
agentHeaderCompression	Diese Eigenschaft wird nur für Clientverbindungen unterstützt. Eine durch Kommas getrennte Liste mit Komprimierungstypen für die Übertragung von Headerdaten, über die mit dem fernen IBM MQ-Server verhandelt wird. Die gültigen Werte lauten MQCOMPRESS_NONE bzw. MQCOMPRESS_SYSTEM. Informationen zu diesen Komprimierungstypen finden Sie im folgenden Abschnitt: HdrCompList [2] (MQLONG) . Die Werte werden auf ihre Gültigkeit überprüft und anschließend in der angegebenen Reihenfolge als Eigenschaften an den Clientkanal des Agenten übergeben. Der IBM MQ-Client verarbeitet nun Vereinbarungen zwischen diesem Clientkanal und dem fernen Serverkanal, um den kleinsten gemeinsamen Nenner der Komprimierungseigenschaften auf den beiden Kanälen zu finden. Wenn es keine Übereinstimmung gibt, wird immer MQCOMPRESS_NONE ausgewählt.	MQCOMPRESS_NONE



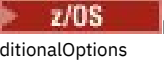
Tabelle 93. Erweiterte Agenteneigenschaften: z/OS-spezifisch		
Eigenschaftsname	Beschreibung	Standardwert
adminGroup	<p>Eine Sicherheitsmanagergruppe. Mitglieder dieser Gruppe können:</p> <ul style="list-style-type: none"> den Agenten mit dem Befehl fteStartAgent starten, den Agenten mit dem Befehl fteStopAgent stoppen, mit dem Befehl fteSetAgentTraceLevel einen Trace für den Agenten aktivieren oder inaktivieren und mit dem Befehl fteSetAgentLogLevel Protokolle für den Agenten aktivieren oder inaktivieren. <p>Zeigen Sie Einzelheiten zu einem lokalen Agenten an, indem Sie den Befehl fteShowAgentDetails mit dem Parameter -d ausführen.</p> <p>Definieren Sie eine Sicherheitsmanagergruppe, z. B. MFTADMIN, und fügen Sie dann die Benutzer-ID für gestartete Tasks und Administrator-TSO-IDs zu dieser Gruppe hinzu. Bearbeiten Sie die Agenteneigenschaftendatei und setzen Sie die Eigenschaft adminGroup auf den Namen dieser Sicherheitsmanagergruppe.</p> <pre>adminGroup=MFTADMIN</pre>	--
 bpxwdynAllocAdditionalOptions	<p>Managed File Transfer verwendet zum Erstellen und Öffnen von z/OS-Datensätzen die Textschnittstelle BPXWDYN. Wenn BPXWDYN standardmäßig für die Datasetzuordnung verwendet wird, stellt Managed File Transfer, sofern möglich, sicher, dass das Datengerät gemountet ist (dies ist für Datensätze auf Festplatten nicht nötig, für Banddatensätze hingegen schon). Da die BPXWDYN-Optionen in einigen Umgebungen nicht unterstützt werden, können Sie dieses Verhalten mit dieser Eigenschaft ändern. Bei der Übertragung an ein Dataset können Optionen für BPXWDYN auch in der Befehlszeile angegeben werden. Diese Optionen kommen zu den über diese Eigenschaft definierten Optionen hinzu.</p> <p>Bei Verwendung der Eigenschaft bpxwdynAllocAdditionalOptions in der Datei <code>agent.properties</code> dürfen einige BPXWDYN-Optionen nicht angegeben werden. Eine Liste dieser Eigenschaften finden Sie im Abschnitt BPXWDYN-Eigenschaften, die nicht mit MFT verwendet werden dürfen.</p>	<p>Die Standardwerte lauten wie folgt:</p> <ul style="list-style-type: none"> MOUNT für z/OS V1R8 und höher
armELEMTYPE	Optionale Eigenschaft. Wenn der Agent so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMTYPE, der in der zugehörigen ARM-Richtlinie festgelegt ist. Für einen Agent muss ELEMTYPE auf SYSBFGAG gesetzt werden.	Nicht festgelegt
armELEMENT	Optionale Eigenschaft. Wenn der Agent so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMENT, der in der zugehörigen ARM-Richtlinie festgelegt ist. Der Wert von ELEMENT kann dem Namen des Agenten entsprechen.	Nicht festgelegt

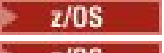
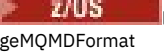
Tabelle 94. Erweiterte Agenteneigenschaften: Andere Eigenschaften		
Eigenschaftsname	Beschreibung	Standardwert
  legacyXMLMessageMQMDFormat	<p>Vom Managed File Transfer-Agenten generierte XML-Nachrichten (z. B. Protokoll- und Übertragungsstatusnachrichten) werden nun an eine Warteschlange mit einem leeren MQMD-Formatfeld gesendet. In früheren Produktversionen war das MQMD-Formatfeld auf MQSTR gesetzt (eine Textnachrichten-Zeichenfolge). Wenn diese Eigenschaft auf "true" gesetzt ist, werden die vom Managed File Transfer-Agenten generierten XML-Nachrichten an eine Warteschlange gesendet, deren MQMD-Formatfeld auf MQSTR gesetzt ist.</p> <p>Anmerkung: Das Nachrichtenformat der Antwortnachrichten des Agenten auf Befehle entspricht dem Format der Befehlsanforderung.</p> <p>Wenn das MQMD-Formatfeld auf MQSTR gesetzt ist und das MQ-Netz Kanäle mit Datenkonvertierung enthält, besteht die Gefahr, dass die XML-Nachrichten für Managed File Transfer-Befehle beschädigt werden.</p>	false

Tabella 94. Erweiterte Agenteneigenschaften: Andere Eigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
adjustScheduleTimeForDaylightSaving	<p>Wenn Ihr Unternehmen jeden Tag planmäßige Übertragungen durchführt, d. h. mit:</p> <ul style="list-style-type: none"> • Parameter -oia auf Tage gesetzt und • Parameter -tba auf Quelle gesetzt <p>im Befehl <code>fteCreateTransfer</code> eingestellt ist, setzen Sie dieses Attribut auf <code>true</code>, um die geplante Übertragungszeit um eine Stunde nach vorne zu verschieben, wenn die Uhren um eine Stunde vorgestellt werden, und um eine Stunde zurück, wenn die Uhren um eine Stunde zurückgestellt werden.</p> <p>Wenn Ihre geplante Überweisung beispielsweise um 1:00 Uhr nachts erfolgen soll, wird die Überweisung bei der Zeitumstellung um 2:00 Uhr nachts ausgeführt, und wenn die Uhren zurückgestellt werden, wird die Überweisung auf 1:00 Uhr zurückgesetzt.</p>	true

Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

[Zeitlimitoption für die Wiederherstellung von Dateiübertragungen](#)

[MFT-Sandboxes](#)

Zugehörige Tasks

[MFT-Agenten für mehrere Kanäle in einem Cluster konfigurieren](#)

[MFT-Agenten für mehrere Kanäle konfigurieren: clusterunabhängig](#)

Zugehörige Verweise

[„Java-Systemeigenschaften für MFT“ auf Seite 226](#)

Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

[SSL/TLS-Eigenschaften für MFT](#)

[„Die MFT-Datei 'command.properties'“ auf Seite 209](#)

In der Datei `command.properties` ist der Befehlswarteschlangenmanager angegeben, zu dem eine Verbindung hergestellt werden muss, wenn Befehle ausgegeben werden. Außerdem enthält die Datei Informationen, die Managed File Transfer für den Kontakt zu diesem Warteschlangenmanager benötigt.

[„Die MFT-Datei 'coordination.properties'“ auf Seite 204](#)

Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden.

[„Die MFT-Datei 'logger.properties'“ auf Seite 213](#)

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.

[fteCreateAgent](#)

[fteCreateBridgeAgent](#)

[fteCreateCDAgent](#)

[„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174](#)

Es ist möglich, dass Umgebungsvariablen in Managed File Transfer -Eigenschaften verwendet werden, die Datei- oder Verzeichnispositionen darstellen. Dadurch können die Positionen der Dateien oder Verzeichnisse, die bei der Ausführung von Teilen des Produkts verwendet werden, abhängig von der aktuellen Umgebung variieren (z. B. der Benutzer, der einen Befehl ausführt).

Die MFT-Datei 'coordination.properties'

Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager

gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden.

Die Datei `coordination.properties` wird vom Installationsprogramm oder über den Befehl **`fteSetupCoordination`** erstellt. Wenn Sie die grundlegenden Eigenschaften des Koordinationswarteschlangenmanagers in dieser Datei ändern möchten, verwenden Sie hierfür den Befehl **`fteSetupCoordination`** mit dem Flag **`-f`**. Um erweiterte Eigenschaften des Koordinationswarteschlangenmanagers zu ändern bzw. hinzuzufügen, muss die Datei in einem Texteditor bearbeitet werden.

Die Datei `coordination.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name`.

Die Datei MFT `coordination.properties` enthält die folgenden Werte:

Tabelle 95. Basiseigenschaften des Koordinierungwarteschlangenmanagers		
Eigenschaftsname	Beschreibung	Standardwert
<code>coordinationCredentialsKeyFile</code>	Der Name der Datei, die den Berechtigungsnachweisschlüssel enthält, der beim Verschlüsseln von Berechtigungsnachweisen verwendet wird.	Eine Zeichenfolgeeigenschaft, die keinen Standardwert hat.
<code>coordinationQMGr</code>	Name des Koordinationswarteschlangenmanagers.	Kein Standardwert
<code>coordinationQMGrHost</code>	Hostname oder IP-Adresse des Koordinationswarteschlangenmanagers.	Kein Standardwert
<code>coordinationQMGrPort</code>	Die für Clientverbindungen zum Koordinationswarteschlangenmanager verwendete Portnummer.	1414
<code>coordinationQMGrChannel</code>	Der SVRCONN-Kanalname, der zur Verbindung zum Koordinations-Warteschlangenmanager verwendet wird.	SYSTEM.DEF.SVRCONN

Wenn Sie keinen Wert für die Eigenschaft `'coordinationQMGrHost'` angeben, wird standardmäßig der Bindungsmodus verwendet.

Wenn Sie für die Eigenschaft `coordinationQMGrHost` einen Wert angeben, nicht jedoch für die Eigenschaften von `coordinationQMGrPort` und `coordinationQMGrChannel`, so werden standardmäßig Portnummer 1414 und Kanal `SYSTEM.DEF.SVRCONN` verwendet.

Das folgende Beispiel zeigt den Inhalt einer Datei `coordination.properties`:

```
coordinationQMGr=ERIS
coordinationQMGrHost=kuiper.example.com
coordinationQMGrPort=2005
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
```

In diesem Beispiel ist ERIS der Name eines IBM MQ-Warteschlangenmanagers auf dem System `kuiper.example.com`. ERIS ist der Warteschlangenmanager, an den Managed File Transfer Protokollinformationen sendet.

Erweiterte Eigenschaften für die Koordination

Managed File Transfer stellt auch erweiterte Eigenschaften für die Koordination bereit. Wenn Sie eine der folgenden Eigenschaften verwenden wollen, bearbeiten Sie die Datei `coordination.properties` manuell, um die erforderlichen erweiterten Eigenschaften hinzuzufügen. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen

versehen werden. **Windows** Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Weitere Informationen zur Verwendung von Escapezeichen in Java -Eigenschaftendateien finden Sie in der Oracle -Dokumentation [Javadoc für die Eigenschaftsklasse](#).

- [Agenteneigenschaften](#)
- [Codepageeigenschaften](#)
- [Verbindungseigenschaften](#)
- [Eigenschaften für Manager von Multi-Instanz-Warteschlangen](#)
- [Warteschlangeneigenschaften](#)

- Sicherheitseigenschaften
- SSL-Eigenschaften
- Subskriptionseigenschaften

Tabelle 96. Erweiterte Koordinationseigenschaften: Agent

Eigenschaftsname	Beschreibung	Standardwert
agentStatusJitterTolerance	<p>Die Höchstdauer, die die Veröffentlichung einer Nachricht über den Agentenstatus verzögert werden kann, bevor die Nachricht als überfällig betrachtet wird. Dieser Wert wird in Millisekunden gemessen.</p> <p>Das Alter einer Statusnachricht basiert auf dem Zeitpunkt, zu dem sie im Koordinations-Warteschlangenmanager veröffentlicht wurde. Die Nachricht wird vom Agenten jedoch einige Zeit, bevor sie am Koordinationswarteschlangenmanager empfangen wird, ausgegeben, um der Dauer der Übertragung im IBM MQ-Netz Rechnung zu tragen. Dauert dieser Durchlauf immer gleich lange, werden Nachrichten, die mit einem Abstand von 60 Sekunden erstellt wurden, ungeachtet der tatsächlichen Übertragungsdauer mit einem Abstand von 60 Sekunden veröffentlicht. Wenn die Dauer des Transit jedoch zwischen Nachrichten variiert, werden sie möglicherweise in Intervallen von 60 Sekunden erstellt, aber in Intervallen von beispielsweise 61, 59, 58 und 62 Sekunden veröffentlicht. Die maximale Abweichung von 60 (in diesem Beispiel 2 Sekunden) wird als Jitter bezeichnet. Diese Eigenschaft legt die maximale Verzögerung aufgrund des Jitters vor der Behandlung der Nachricht als überfällig fest.</p>	3000

Tabelle 97. Erweiterte Koordinationseigenschaften: Codepage

Eigenschaftsname	Beschreibung	Standardwert
coordinationCcsid	Die Codepage, mit der die Befehle eine Verbindung zum Koordinationswarteschlangenmanager herstellen. Diese Codepage wird auch für alle Veröffentlichungen des Agenten auf dem Koordinationswarteschlangenmanager verwendet. Wenn Sie einen Wert für die Eigenschaft 'coordinationCcsid' angeben, müssen Sie auch einen Wert für 'coordinationCcsidName' angeben.	1208
coordinationCcsidName	Die Java-Darstellung der ID des codierten Zeichensatzes für die Koordination (coordinationCcsid). Wenn Sie einen Wert für die Eigenschaft 'coordinationCcsidName' angeben, müssen Sie auch einen Wert für 'coordinationCcsid' angeben.	UTF8

Tabelle 98. Erweiterte Koordinationseigenschaften: Verbindung

Eigenschaftsname	Beschreibung	Standardwert
javaLibraryPath	Wenn eine Verbindung zu einem Warteschlangenmanager im Bindungsmodus hergestellt wird, muss Managed File Transfer auf die Bindungsbibliotheken von IBM MQ Java zugreifen können. Managed File Transfer sucht die Bindungsbibliotheken standardmäßig an der durch IBM MQ definierten Standardposition. Wenn sich die Bindungsbibliotheken an einer anderen Position befinden, geben Sie mit dieser Eigenschaft die Position der Bindungsbibliotheken an.	MQ_INSTALLATION_PATH/java/lib

Tabelle 99. Erweiterte Koordinationseigenschaften: Manager von Multi-Instanz-Warteschlangen

Eigenschaftsname	Beschreibung	Standardwert
coordinationQMGrStandby	Der Hostname und die Portnummer für Clientverbindungen (im IBM MQ CON-NAME-Format), und zwar für die Standby-Instanz eines Multi-Instanz-Koordinationswarteschlangenmanagers (definiert über die Eigenschaft 'coordinationQMGr'). Zum Beispiel, <i>host_name(port_number)</i>	Kein Standardwert

Tabelle 100. Erweiterte Koordinationseigenschaften: Warteschlange

Eigenschaftsname	Beschreibung	Standardwert
dynamicQueuePrefix	<p>Diese Eigenschaft definiert das IBM MQ-Präfix für den Namen der temporären Warteschlange.</p> <p>Das Format der Eigenschaft 'dynamicQueuePrefix' entspricht dem Format des Feldes DynamicQName der MQOD-Struktur von IBM MQ. Weitere Informationen finden Sie im Abschnitt Dynamische Warteschlangen erstellen.</p> <p>Diese Eigenschaft können Sie auch in der Datei <code>command.properties</code> definieren, wenn Sie ein bestimmtes IBM MQ-Präfix für temporäre Antwortwarteschlangen verwenden möchten, die durch Befehle generiert werden, von denen eine Antwort des Agenten benötigt wird.</p>	WMQFTE.*

Tabelle 100. Erweiterte Koordinationseigenschaften: Warteschlange (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
modelQueueName	<p>Diese Eigenschaft definiert die IBM MQ-Modellwarteschlange, die zum Generieren einer temporären Warteschlange verwendet werden soll.</p> <p>Sie können diese Eigenschaft auch in der Datei 'command.properties' definieren, wenn Sie eine bestimmte IBM MQ-Modellwarteschlange für temporäre Antwortwarteschlangen verwenden möchten, die über Befehle generiert werden, welche eine Antwort des Agenten erfordern. Weitere Informationen finden Sie unter „Die MFT-Datei 'command.properties'“ auf Seite 209.</p>	SYSTEM.DEFAULT.MODEL.QUEUE

Tabelle 101. Erweiterte Koordinationseigenschaften: Sicherheit		
Eigenschaftsname	Beschreibung	Standardwert
userIdForClientConnect	Die Benutzer-ID, die über Clientverbindungen an IBM MQ weitergegeben wird. Bei Angabe von <code>java</code> wird der von der JVM gemeldete Benutzername als Teil der IBM MQ-Verbindungsanforderung weitergegeben. Mögliche Werte für diese Eigenschaft sind <code>None</code> (Keine) oder <code>java</code> .	--
coordinationQMGrAuthenticationCredentialsFile	Der Pfad der Datei, die die MQ-Berechtigungsanforderung für die Verbindung mit dem Koordinationswarteschlangenmanager enthält.	<p>z/OS Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter z/OS konfigurieren.</p> <p>ALW Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt MQMFTCredentials.xml konfigurieren.</p> <p>ALW Weitere Informationen zum Erstellen der Authentifizierungsberechtigungsdatei finden Sie unter MFT- und IBM MQ-Authentifizierungsnachweis.</p>

Tabelle 102. Erweiterte Koordinationseigenschaften: SSL/TLS		
Eigenschaftsname	Beschreibung	Standardwert
coordinationSslCipherSpec	<p>Gibt beim Austausch von Daten zwischen den Befehlen und dem Koordinationswarteschlangenmanager das Protokoll, den Hashalgorithmus und den Verschlüsselungsalgorithmus sowie Informationen zur Anzahl der im Verschlüsselungsschlüssel verwendeten Bits an.</p> <p>Der Wert von 'coordinationSslCipherSpec' ist ein CipherSpec-Name. Dieser CipherSpec-Name ist mit dem im Kanal des Koordinationswarteschlangenmanagers verwendeten CipherSpec-Namen identisch. Eine Liste gültiger CipherSpec-Namen ist in den Abschnitten SSL/TLS-CipherSpecs und -CipherSuites in IBM MQ-Klassen für Java und SSL/TLS-CipherSpecs und -CipherSuites in IBM MQ-Klassen für JMS zu finden.</p> <p>'coordinationSslCipherSpec' entspricht weitgehend 'coordinationSslCipherSuite'. Wenn sowohl 'coordinationSslCipherSuite' als auch 'coordinationSslCipherSpec' angegeben ist, wird der Wert von 'coordinationSslCipherSpec' verwendet.</p>	--
coordinationSslCipherSuite	<p>Gibt SSL-Aspekte zum Austausch von Daten zwischen den Befehlen und dem Koordinationswarteschlangenmanager an.</p> <p>Beim Wert von 'coordinationSslCipherSuite' handelt es sich um einen CipherSuite-Namen. Dieser CipherSuite-Name ist dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen zugeordnet. Weitere Informationen hierzu finden Sie im Abschnitt Namenszuordnungen von CipherSuites und CipherSpecs.</p> <p>'coordinationSslCipherSuite' entspricht weitgehend 'coordinationSslCipherSpec'. Wenn sowohl 'coordinationSslCipherSuite' als auch 'coordinationSslCipherSpec' angegeben ist, wird der Wert von 'coordinationSslCipherSpec' verwendet.</p>	--
coordinationSslPeerName	Gibt den Entwurf eines definierten Namens an, der mit dem vom Koordinationswarteschlangenmanager bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Koordinationswarteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.	--

Tabelle 102. Erweiterte Koordinationseigenschaften: SSL/TLS (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
coordinationSslTrustStore	Gibt die Position der Zertifikate an, die die Befehle akzeptieren. Beim Wert von 'coordinationSslTrustStore' handelt es sich um einen Dateipfad. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden. Windows Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	--
coordinationSslTrustStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann jks oder pkcs12 angegeben werden.	jks
coordinationSslTrustStoreCredentialsFile	Der Pfad der Datei, die die coordinationSslTrustStore-Berechtigungsnachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert %USERPROFILE%/MQMFTCredentials.xml unter Windows bzw. \$HOME/MQMFTCredentials.xml auf anderen Plattformen.
coordinationSslKeyStore	Gibt die Position des privaten Schlüssels der Befehle an. Beim Wert von 'coordinationSslKeyStore' handelt es sich um einen Dateipfad. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden. Windows Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Diese Eigenschaft muss nur angegeben werden, wenn die Clientauthentifizierung für den Koordinationswarteschlangenmanager erforderlich ist. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	--
coordinationSslKeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann jks oder pkcs12 angegeben werden.	jks
coordinationSslKeyStoreCredentialsFile	Der Pfad der Datei, die die coordinationSslKeyStore-Berechtigungsnachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert %USERPROFILE%/MQMFTCredentials.xml unter Windows bzw. \$HOME/MQMFTCredentials.xml auf anderen Plattformen.
coordinationSslFipsRequired	Gibt an, dass Sie die FIPS-Unterstützung auf Ebene des Koordinationswarteschlangenmanagers aktivieren möchten. Mögliche Werte für diese Eigenschaft sind true oder false. Weitere Informationen finden Sie im Abschnitt FIPS-Unterstützung in MFT .	false

Tabelle 103. Erweiterte Koordinationseigenschaften: Subskription

Eigenschaftsname	Beschreibung	Standardwert
coordinationSubscriptionTopic	Verwenden Sie diese Eigenschaft, um ein anderes Thema als SYSTEM.FTE zu abonnieren, um Veröffentlichungen zum Status des IBM MQ -Netzes zu erhalten. Alle Tools werden weiterhin im SYSTEM.FTE, aber Sie können Ihre IBM MQ -Topologie ändern, um diese Veröffentlichungen je nach Inhalt an verschiedene Themen zu verteilen. Danach können Sie die Tools mit dieser Funktion so konfigurieren, dass sie eines dieser anderen Themen abonnieren.	SYSTEM.FTE

Zugehörige Konzepte

MFT-Konfigurationsoptionen unter Multiplattformen

Zugehörige Verweise

[fteSetupCoordination](#)

[SSL/TLS-Eigenschaften für MFT](#)

„Die MFT agent.properties-Datei“ auf Seite 180

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens agent.propertiesagent.properties, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei agent.properties angegeben sein.

„Die MFT-Datei 'command.properties'“ auf Seite 209

In der Datei `command.properties` ist der Befehlswarteschlangenmanager angegeben, zu dem eine Verbindung hergestellt werden muss, wenn Befehle ausgegeben werden. Außerdem enthält die Datei Informationen, die Managed File Transfer für den Kontakt zu diesem Warteschlangenmanager benötigt.

„Die MFT-Datei 'logger.properties'“ auf Seite 213

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.

Die MFT-Datei 'command.properties'

In der Datei `command.properties` ist der Befehlswarteschlangenmanager angegeben, zu dem eine Verbindung hergestellt werden muss, wenn Befehle ausgegeben werden. Außerdem enthält die Datei Informationen, die Managed File Transfer für den Kontakt zu diesem Warteschlangenmanager benötigt.

Die Datei `command.properties` wird vom Installationsprogramm oder über den Befehl **fteSetupCommands** erstellt. Wenn Sie die grundlegenden Eigenschaften des Befehlswarteschlangenmanagers in dieser Datei ändern möchten, verwenden Sie hierfür den Befehl **fteSetupCommands** mit dem Flag **-f**. Um erweiterte Eigenschaften des Befehlswarteschlangenmanagers zu ändern bzw. hinzuzufügen, muss die Datei in einem Texteditor bearbeitet werden.

Bei einigen Managed File Transfer-Befehlen wird eine Verbindung zum Agentenwarteschlangenmanager oder zum Koordinationswarteschlangenmanager und nicht zum Befehlswarteschlangenmanager hergestellt. Im Abschnitt [Verbindung zwischen MFT-Befehlen und Warteschlangenmanager](#) erfahren Sie, welche Befehle zu welchen Warteschlangenmanagern Verbindungen herstellen.

Die Datei `command.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name`.

Die Datei MFT `command.properties` enthält die folgenden Werte:

Eigenschaftsname	Beschreibung	Standardwert
<code>connectionCredentialsKeyFile</code>	Der Name der Datei, die den Berechtigungsnachweisschlüssel enthält, der beim Verschlüsseln von Berechtigungsnachweisen verwendet wird.	Eine Zeichenfolgeeigenschaft, die keinen Standardwert hat.
<code>connectionQMGr</code>	Der Name des für die Verbindung mit dem IBM MQ-Netz verwendeten Warteschlangenmanagers.	Kein Standardwert
<code>connectionQMGrHost</code>	Hostname oder IP-Adresse des Verbindungs-Warteschlangenmanagers.	Kein Standardwert
<code>connectionQMGrPort</code>	Die für die Verbindung zum Verbindungs-Warteschlangenmanager in Client Mode verwendete Portnummer.	1414
<code>connectionQMGrChannel</code>	Der SVRCONN-Kanalname, der zur Verbindung zum Verbindungs-Warteschlangenmanager verwendet wird.	SYSTEM.DEF.SVRCONN

Wenn Sie keinen Wert für die Eigenschaft 'connectionQMGrHost' angeben, wird standardmäßig der Bindungsmodus verwendet.


Wenn Sie einen Wert für die Eigenschaft von `connectionQMGrHost` angeben, jedoch nicht für die Eigenschaften von `connectionQMGrPort` und `connectionQMGrChannel`, werden standardmäßig die Portnummer 1414 und der Kanal `SYSTEM.DEF.SVRCONN` verwendet.

Das folgende Beispiel zeigt den Inhalt einer Datei `command.properties`:

```
connectionQMGr=PLUTO
connectionQMGrHost=kuiper.example.com
connectionQMGrPort=1930
connectionQMGrChannel=SYSTEM.DEF.SVRCONN
```

In diesem Beispiel ist `PLUTO` der Name eines IBM MQ-Warteschlangenmanagers auf dem System `kuiper.example.com`. `PLUTO` ist der Warteschlangenmanager, zu dem die Managed File Transfer-Befehle eine Verbindung herstellen.

Erweiterte Eigenschaften für Befehle

Managed File Transfer stellt auch erweiterte Eigenschaften für Befehle bereit. Wenn Sie eine der folgenden Eigenschaften verwenden wollen, bearbeiten Sie die Datei `command.properties` manuell, um die erforderlichen erweiterten Eigenschaften hinzuzufügen. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden.  Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Weitere Informationen zur Verwendung von Escapezeichen in Java -Eigenschaftendateien finden Sie in der Oracle -Dokumentation [Javadoc für die Eigenschaftsklasse](#).

- [Agenteneigenschaften](#)
- [Codepageeigenschaften](#)
- [Eigenschaften für Manager von Multi-Instanz-Warteschlangen](#)
- [Warteschlangeneigenschaften](#)
- [Sicherheitseigenschaften](#)
- [SSL-Eigenschaften](#)

Tabelle 105. Erweiterte Eigenschaften für Befehle: Agent		
Eigenschaftsname	Beschreibung	Standardwert
failCleanAgentWithNoArguments	Diese Eigenschaft hat standardmäßig den Wert 'true', was bedeutet, dass die Ausführung des Befehls fteCleanAgent fehlschlägt, wenn nur der Parameter für den Agentennamen angegeben ist. Wird die Eigenschaft auf 'false' gesetzt und nur der Parameter für den Agentennamen angegeben, verhält sich der Befehl fteCleanAgent so, als wäre der Parameter -all angegeben.	true

Tabelle 106. Erweiterte Eigenschaften für Befehle: Codepage		
Eigenschaftsname	Beschreibung	Standardwert
connectionCcsid	Die Codepage, mit der die Befehle eine Verbindung zum Befehlswarteschlangenmanager herstellen. Wenn Sie einen Wert für die Eigenschaft 'connectionCcsid' angeben, müssen Sie auch einen Wert für 'connectionCcsidName' angeben.	1208
connectionCcsidName	Die Java-Darstellung der ID des codierten Zeichensatzes für Verbindungen (connectionCcsid). Wenn Sie einen Wert für die Eigenschaft 'connectionCcsidName' angeben, müssen Sie auch einen Wert für 'connectionCcsid' angeben.	UTF8

Tabelle 107. Erweiterte Eigenschaften für Verbindungen: Manager von Multi-Instanz-Warteschlangen		
Eigenschaftsname	Beschreibung	Standardwert
connectionQMGrStandby	Der Hostname und die Portnummer für Clientverbindungen (im IBM MQ CONNAME-Format), und zwar für die Standby-Instanz eines Multi-Instanz-Befehlswarteschlangenmanagers (definiert über die Eigenschaft 'connectionQMGr'). Zum Beispiel, <code>host_name(port_number)</code>	Kein Standardwert

Tabelle 108. Erweiterte Eigenschaften für Befehle: Warteschlange		
Eigenschaftsname	Beschreibung	Standardwert
dynamicQueuePrefix	Bei Befehlen, die eine Antwort vom Agenten benötigen, definiert diese Eigenschaft das IBM MQ-Präfix, das für die Generierung des Namens der temporären Antwortwarteschlange verwendet werden soll. Das Format der Eigenschaft 'dynamicQueuePrefix' entspricht dem Format des Feldes DynamicQName der MQOD-Struktur von IBM MQ. Weitere Informationen finden Sie im Abschnitt Dynamische Warteschlangen erstellen . Sie können diese Eigenschaft auch in der Datei <code>coordination.properties</code> definieren, wenn Sie für von WMQFTE generierte temporäre Warteschlangen ein bestimmtes IBM MQ-Präfix verwenden möchten.	WMQFTE.*



Tabelle 108. Erweiterte Eigenschaften für Befehle: Warteschlange (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
modelQueueName	Bei Befehlen, die eine Antwort vom Agenten benötigen, definiert diese Eigenschaft die IBM MQ-Modellwarteschlange, die für die Generierung der temporären Antwortwarteschlange verwendet werden soll. Sie können diese Eigenschaft auch in der Datei 'coordination.properties' definieren, wenn Sie für von WMQFTE generierte temporäre Warteschlangen eine bestimmte IBM MQ-Modellwarteschlange verwenden möchten. Weitere Informationen finden Sie unter „Die MFT-Datei 'coordination.properties'“ auf Seite 204.	SYSTEM.DEFAULT.MODEL.QUEUE
Verbindungseigenschaften:		
javaLibraryPath	Wenn eine Verbindung zu einem Warteschlangenmanager im Bindungsmodus hergestellt wird, muss Managed File Transfer auf die Bindungsbibliotheken von IBM MQ Java zugreifen können. Managed File Transfer sucht die Bindungsbibliotheken standardmäßig an der durch IBM MQ definierten Standardposition. Wenn sich die Bindungsbibliotheken an einer anderen Position befinden, geben Sie mit dieser Eigenschaft die Position der Bindungsbibliotheken an.	/opt/mqm/java/lib
  legacyXMLMessageMQMDFormat	Durch Managed File Transfer-Befehle generierte XML-Nachrichten werden nun an eine Warteschlange mit einem leeren MQMD-Formatfeld gesendet. In früheren Produktversionen war das MQMD-Formatfeld auf MQSTR gesetzt (eine Textnachrichten-Zeichenfolge). Wenn diese Eigenschaft auf "true" gesetzt ist, werden die durch Managed File Transfer generierten XML-Nachrichten an eine Warteschlange gesendet, deren MQMD-Formatfeld auf MQSTR gesetzt ist. Wenn das MQMD-Formatfeld auf MQSTR gesetzt ist und das MQ-Netz Kanäle mit Datenkonvertierung enthält, besteht die Gefahr, dass die XML-Nachrichten für Managed File Transfer-Befehle beschädigt werden.	false

Tabelle 109. Erweiterte Eigenschaften für Befehle: Sicherheit		
Eigenschaftsname	Beschreibung	Standardwert
userIdForClientConnect	Die Benutzer-ID, die über Clientverbindungen an IBM MQ weitergegeben wird. Bei Angabe von <i>java</i> wird der von der JVM gemeldete Benutzername als Teil der IBM MQ-Verbindungsanforderung weitergegeben. Mögliche Werte für diese Eigenschaft sind None (Keine) oder <i>java</i> .	--
connectionQMGrAuthenticationCredentialsFile	Der Pfad der Datei, die die MQ-Berechtigungsanzeige für die Verbindung mit dem Befehlswarteschlangenmanager enthält.	Weitere Informationen finden Sie unter MFT und IBM MQ Verbindungsauthentifizierung und den zugehörigen untergeordneten Themen.

Tabelle 110. Erweiterte Eigenschaften für Befehle: SSL/TLS		
Eigenschaftsname	Beschreibung	Standardwert
connectionSslCipherSpec	Gibt beim Austausch von Daten zwischen den Befehlen und dem Befehlswarteschlangenmanager das Protokoll, den Hashalgorithmus und den Verschlüsselungsalgorithmus sowie Informationen zur Anzahl der im Verschlüsselungsschlüssel verwendeten Bits an. Beim Wert von 'connectionSslCipherSpec' handelt es sich um einen CipherSpec-Namen. Dieser CipherSpec-Name ist mit dem im Kanal des Befehlswarteschlangenmanagers verwendeten CipherSpec-Namen identisch. Eine Liste gültiger CipherSpec-Namen ist in SSL/TLS-CipherSpecs und -CipherSuites in IBM MQ-Klassen für Java und SSL/TLS-CipherSpecs und -CipherSuites in IBM MQ-Klassen für JMS zu finden. 'connectionSslCipherSpec' entspricht weitgehend 'connectionSslCipherSuite'. Wenn sowohl 'connectionSslCipherSuite' als auch 'connectionSslCipherSpec' angegeben ist, wird der Wert von 'connectionSslCipherSpec' verwendet.	--
connectionSslCipherSuite	Gibt SSL-Aspekte zum Austausch von Daten zwischen den Befehlen und dem Warteschlangenmanager für den Befehl an. Beim Wert von 'connectionSslCipherSuite' handelt es sich um einen CipherSuite-Namen. Dieser CipherSuite-Name ist dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen zugeordnet. Weitere Informationen hierzu finden Sie im Abschnitt Namenszuordnungen von CipherSuites und CipherSpecs . 'connectionSslCipherSuite' entspricht weitgehend 'connectionSslCipherSpec'. Wenn sowohl 'connectionSslCipherSuite' als auch 'connectionSslCipherSpec' angegeben ist, wird der Wert von 'connectionSslCipherSpec' verwendet.	--

Tabella 110. Erweiterte Eigenschaften für Befehle: SSL/TLS (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
connectionSslPeerName	Gibt den Entwurf eines definierten Namens an, der mit dem vom Befehlswarteschlangenmanager bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Befehlswarteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.	--
connectionSslTrustStore	Gibt die Position der Zertifikate an, die die Befehle akzeptieren. Beim Wert von 'connectionSslTrustStore' handelt es sich um einen Dateipfad. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden. Windows Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	--
connectionSslTrustStoreType	Die Art des zu verwendenden SSL-Truststores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann jks oder pkcs12 angegeben werden.	jks
connectionSslTrustStoreCredentials-File	Der Pfad der Datei, die die connectionSslTrustStore-Berechtigungs-nachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert %USERPROFILE%/MQMFTCredentials.xml unter Windows bzw. \$HOME/MQMFTCredentials.xml auf anderen Plattformen.
connectionSslKeyStore	Gibt die Position des privaten Schlüssels der Befehle an. Beim Wert von 'connectionSslKeyStore' handelt es sich um einen Dateipfad. Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden. Windows Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Diese Eigenschaft muss nur angegeben werden, wenn die Clientauthentifizierung für den Befehlswarteschlangenmanager erforderlich ist. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	--
connectionSslKeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann jks oder pkcs12 angegeben werden. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	jks
connectionSslKeyStoreCredentialsFile	Der Pfad der Datei, die die connectionSslKeyStore-Berechtigungs-nachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert %USERPROFILE%/MQMFTCredentials.xml unter Windows bzw. \$HOME/MQMFTCredentials.xml auf anderen Plattformen.
connectionSslFipsRequired	Gibt an, dass Sie die FIPS-Unterstützung auf Ebene des Befehlswarteschlangenmanagers aktivieren möchten. Mögliche Werte für diese Eigenschaft sind true oder false. Weitere Informationen finden Sie im Abschnitt <u>FIPS-Unterstützung in MFT</u> .	false

Zugehörige Konzepte

MFT-Konfigurationsoptionen unter Multiplattformen

Zugehörige Verweise

„Java-Systemeigenschaften für MFT“ auf Seite 226

Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

SSL/TLS-Eigenschaften für MFT

„Die MFT agent.properties-Datei“ auf Seite 180

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens agent.properties, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei agent.properties angegeben sein.

„Die MFT-Datei 'coordination.properties'“ auf Seite 204

Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden.

„Die MFT-Datei 'logger.properties'“ auf Seite 213

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.


[fteSetupCommands: MFT-Datei 'command.properties' erstellen](#)

[fteCleanAgent: MFT-Agenten bereinigen](#)

Die MFT-Datei 'logger.properties'

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.

In einigen Managed File Transfer-Eigenschaften, die Datei- oder Verzeichnispositionen darstellen, können Sie Umgebungsvariablen verwenden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt). Weitere Informationen finden Sie unter „[Verwendung von Umgebungsvariablen in MFT-Eigenschaften](#)“ auf Seite 174.

Anmerkung: Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden.  Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden. Weitere Informationen zur Verwendung von Escapezeichen in Java-Eigenschaftendateien in Oracle finden Sie unter [Javadoc für die Klasse 'Properties'](#).

Die Datei MFT `logger.properties` enthält die folgenden Werte:

- „[Eigenschaften von Verbindungen im Bindungsmodus](#)“ auf Seite 213
- „[Eigenschaften für SSL/TLS-Verbindungen im Clientmodus](#)“ auf Seite 221

Eigenschaften von Verbindungen im Bindungsmodus

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus		
Eigenschaftsname	Beschreibung	Standardwert
<code>wmqfte.logger.type</code>	Der verwendete Dateimodus. Setzen Sie diesen Wert auf FILE oder DATABASE.	Kein Standardwert
<code>wmqfte.max.transaction.messages</code>	Die maximale Anzahl Nachrichten, die in einer Transaktion verarbeitet werden, bevor die Transaktion festgeschrieben wird. Bei Verwendung der Umlaufprotokollierung steht einem Warteschlangenmanager ein fester Speicherbereich für unvollständige Daten zur Verfügung. Stellen Sie sicher, dass diese Eigenschaft einen möglichst niedrigen Wert erhält, damit der verfügbare Speicherplatz ausreicht.	50
<code>wmqfte.max.transaction.time</code>	Die maximale Zeit in Millisekunden zwischen Transaktionsfestschreibungen.	5000

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.max.consecutive.reject	Die maximale Anzahl Nachrichten, die nacheinander zurückgewiesen werden kann (d. h., ohne dass eine gültige Nachricht erkannt wird). Bei Überschreiten dieses Wertes geht die Protokollfunktion davon aus, dass der Fehler nicht bei den Nachrichten selbst, sondern bei der Konfiguration liegt. Wenn Sie beispielsweise die Spalte für den Agentennamen in der Datenbank so schmal machen, dass die Agentennamen dafür zu lang sind, werden alle Nachrichten, die auf Agenten verweisen, zurückgewiesen.	50
wmqfte.reject.queue.name	Der Name einer Warteschlange, in die die Protokollfunktion Nachrichten einreicht, die sie nicht verarbeiten kann. Bei Verwendung einer Datenbankprotokollfunktion finden Sie im Abschnitt <u>Fehlerbehandlung und Zurückweisung in der MFT-Protokollfunktion</u> Hinweise zu den Nachrichten, die in diese Warteschlange eingereicht werden können.	SYSTEM.FTE.LOG.RJCT.Name_der_Protokollfunktion
wmqfte.command.queue.name	Der Name einer Warteschlange, aus der die Protokollfunktion Befehlsnachrichten abrufen, über die ihr Verhalten gesteuert wird.	SYSTEM.FTE.LOG.CMD.Name_der_Protokollfunktion
wmqfte.queue.manager	Der Warteschlangenmanager, mit dem sich die Protokollfunktion verbindet. Dieser Parameter ist erforderlich und die einzige benötigte Angabe für Verbindungen im Bindungsmodus zum Warteschlangenmanager. (Die Eigenschaften für Verbindungen mit einem fernen Warteschlangenmanager finden Sie im Abschnitt <u>Tabelle 112 auf Seite 221.</u>)	Kein Standardwert
wmqfte.message.source.type	Einer der folgenden Werte: automatic subscription Dies ist der Standardwert. Die Protokollfunktion erstellt und verwendet in dem in 'SYSTEM.FTE/Log/#' definierten Warteschlangenmanager eine eigene permanente, verwaltete Subskription. Dieser Wert ist für die meisten Szenarios geeignet. administrative subscription Wenn die automatische Subskription nicht geeignet ist, können Sie eine andere Subskription definieren (beispielsweise mithilfe von IBM MQ Explorer, MQSC oder PCF) und die Protokollfunktion anweisen, diese Subskription zu verwenden. Mit dem folgenden Wert beispielsweise können Sie den Protokollbereich so aufteilen, dass von einer Protokollfunktion Agenten von A-H, von einer zweiten Protokollfunktion die von I-P und von einer dritten Protokollfunktion die von Q-Z verarbeitet werden. queue Wenn es aufgrund der IBM MQ-Topologie zu umständlich ist, eine Subskription für die Protokollfunktion zu erstellen, können Sie stattdessen eine Warteschlange verwenden. Konfigurieren Sie IBM MQ so, dass die Warteschlange die Nachrichten empfängt, die normalerweise von einer Subskription für SYSTEM.FTE/Log/# auf dem Koordinationswarteschlangenmanager.	automatic subscription
wmqfte.message.source.name	Wenn der Nachrichtenquellentyp <i>administrative subscription</i> oder <i>queue</i> lautet, ist dies der Name der zu verwendenden Subskription oder Warteschlange. Diese Eigenschaft wird ignoriert, wenn der Quellentyp <i>automatic subscription</i> lautet.	Kein Standardwert

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<p>wmqfte.database.credentials.file</p>	<p>Der Name der Datei, die den Benutzernamen und das Kennwort zum Herstellen einer Verbindung zur Datenbank enthält.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie im Abschnitt MFT-Dateiformat für Berechtigungsnachweise.</p>	<p>z/OS Informationen zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter z/OSkonfigurieren.</p> <p>ALW Informationen zur Position und den Berechtigungen dieser Datei finden Sie unter MQMFTCredentials.xml konfigurieren.</p> <p>ALW Siehe auch MFT- und IBM MQ-Verbindungsauthentifizierung.</p>
<p>wmqfte.database.driver</p>	<p>Die Position der JDBC-Treiberklassen für die Datenbank. Dies ist normalerweise der Pfad und Dateiname einer JAR-Datei.</p> <p>AIX Für den Db2-Treiber des Typs 2 auf AIX-Systemen ist beispielsweise die Datei <code>/opt/IBM/db2/V9.5/java/db2jcc.jar</code> erforderlich.</p> <p>Windows Auf Windows-Systemen wird als Pfadtrennzeichen der Schrägstrich (/) verwendet (Beispiel: <code>C:/Program Files/IBM/SQLLIB/java/db2jcc.jar</code>).</p> <p>z/OS Geben Sie unter z/OS den vollständigen Pfad der Datei <code>db2jcc.jar</code> an. Beispiel: <code>wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar</code>.</p> <p>z/OS Auf z/OS-Systemen müssen alle folgenden JAR-Dateien referenziert werden:</p> <ul style="list-style-type: none"> • <code>db2jcc.jar</code> • <code>db2jcc_license_cisuz.jar</code> • <code>db2jcc_javax.jar</code> <p>Wenn Ihr Datenbanktreiber aus mehreren JAR-Dateien besteht (für Db2 V9.1 sind beispielsweise eine JAR-Treiberdatei und eine JAR-Lizenzdatei erforderlich), schließen Sie alle diese JAR-Dateien in diese Eigenschaft ein. Bei Angabe mehrerer Dateinamen müssen diese mit dem auf Ihrer Plattform üblichen Klassenpfadtrennzeichen getrennt werden; auf Windows-Systemen ist dies das Semikolon (;), auf allen anderen Plattformen der Doppelpunkt (·).</p>	<p>Kein Standardwert</p>

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.database.exclude .duplicate.metadata	<p>Steuert, ob Einträge in der Metadatentabelle gespeichert werden. Diese Tabelle enthält übergeordnete Informationen anderer Tabellen des Schemas der Datenbankprotokollfunktion. Setzen Sie diesen Wert auf <code>true</code> oder <code>false</code>. Die Metadateneinträge werden standardmäßig nicht mehr in dieser Tabelle gespeichert, da dies eine Art der Verschwendung von Datenbankspeicher durch Datenduplizierung ist. Die folgenden Eigenschafteneinträge in den jeweils angegebenen Tabellen enthalten dieselben Daten:</p> <ul style="list-style-type: none"> • <code>com.ibm.wmqfte.SourceAgent</code>: TRANSFER_EVENT oder CALL_REQUEST • <code>com.ibm.wmqfte.DestinationAgent</code>: TRANSFER_EVENT • <code>com.ibm.wmqfte.MqmdUser</code>: TRANSFER_EVENT oder CALL_REQUEST • <code>com.ibm.wmqfte.OriginatingUser</code>: TRANSFER_EVENT oder CALL_REQUEST • <code>com.ibm.wmqfte.OriginatingHost</code>: TRANSFER_EVENT oder CALL_REQUEST • <code>com.ibm.wmqfte.TransferId</code>: TRANSFER oder CALL_REQUEST • <code>com.ibm.wmqfte.JobName</code>: TRANSFER oder CALL_REQUEST <p>Wenn der Wert dieser Eigenschaft auf <code>false</code> gesetzt ist, werden die Metadateneinträge in der Metadatentabelle gespeichert.</p>	true
wmqfte.database.host	<p>Nur für Db2:</p> <p>Der Hostname des Datenbankservers, zu dem über einen JDBC -Treiber des Typs 4 eine Verbindung hergestellt werden soll. Wenn für diese Eigenschaft ein Wert angegeben wird, muss auch für <code>wmqfte.database.port</code> ein Wert angegeben werden. Wenn keine der beiden Eigenschaften definiert ist, stellt die Datenbankprotokollfunktion eine Verbindung über den standardmäßigen Typ-2-JDBC-Treiber her.</p> <p>Wenn für diese Eigenschaft ein Wert angegeben ist, muss für diese Protokollfunktion eine Datei mit Berechtigungsnachweisen vorhanden und zugänglich sein (d. h. ein mit <code>wmqfte.database.credentials.file</code> definierter Dateipfad). Diese Datei bestimmt den Benutzernamen und das Kennwort für die Verbindung mit der Datenbank, selbst wenn sich die Datenbank auf dem lokalen System befindet.</p>	Kein Standardwert
wmqfte.database.name	Der Name der Datenbankinstanz (oder des Subsystems bei der Verwendung von Db2 for z/OS), die die Managed File Transfer-Protokolltabellen enthält.	Kein Standardwert
wmqfte.database.type	Das verwendete Datenbankmanagementsystem Db2 oder Oracle). Setzen Sie diesen Wert auf <code>db2</code> oder <code>oracle</code> .	db2

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)


Eigenschaftsname	Beschreibung	Standardwert
wmqfte.database.port	<p>Nur für Db2:</p> <p>Die Portnummer des Datenbankservers, zu dem über einen JDBC -Treiber des Typs 4 eine Verbindung hergestellt wird. Wenn für diese Eigenschaft ein Wert angegeben wird, muss auch für <code>wmqfte.database.host</code> ein Wert angegeben werden. Wenn keine der beiden Eigenschaften definiert ist, stellt die Datenbankprotokollfunktion eine Verbindung über den standardmäßigen Typ-2-JDBC-Treiber her.</p> <p>Wenn für diese Eigenschaft ein Wert angegeben ist, muss für diese Protokollfunktion eine Datei mit Berechtigungsnachweisen vorhanden und zugänglich sein (d. h. ein mit <code>wmqfte.database.credentials.file</code> definierter Dateipfad). Diese Datei bestimmt den Benutzernamen und das Kennwort für die Verbindung mit der Datenbank, selbst wenn sich die Datenbank auf dem lokalen System befindet.</p>	Kein Standardwert
wmqfte.database.schema	<p>Nur für Db2:</p> <p>Das Datenbankschema, das die Managed File Transfer-Protokolltabellen enthält. In den meisten Fällen ist der Standardwert geeignet, aber Sie können bei Bedarf abhängig von standortspezifischen Datenbankspekten einen alternativen Wert angeben.</p>	FTELOG
wmqfte.database.native.library.path	<p>Der Pfad, in dem sich die nativen Bibliotheken befinden, die der von Ihnen ausgewählte Datenbanktreiber (falls vorhanden) benötigt.</p> <p> Für den Db2-Treiber des Typs 2 auf AIX-Systemen sind beispielsweise Bibliotheken aus <code>/opt/IBM/db2/V9.5/lib32/</code> erforderlich. Alternativ zu dieser Eigenschaft können Sie mit anderen Methoden die Systemeigenschaft 'java.library.path' festlegen.</p>	Kein Standardwert
wmqfte.file.logger.fileDirectory	Das Verzeichnis, in dem sich die Protokolldateien der Dateiprotokollfunktion befinden.	<code>mqft/logs/coordination_dir/loggers/logger_name/logs</code>
wmqfte.file.logger.fileSize	Gibt die Größe an, die eine Protokolldatei maximal erreichen darf. Der Wert wird in Form einer positiven ganzen Zahl größer null angegeben, gefolgt von einer der folgenden Maßeinheiten: KB, MB, GB, m (Minuten), h (Stunden), d (Tage), w (Wochen). Beispiel: <code>wmqfte.file.logger.fileSize=5MB</code> gibt eine maximale Dateigröße von 5MB und <code>wmqfte.file.logger.fileSize=2d</code> eine maximale Dateigröße von 2 Tagen für Daten an.	10MB
wmqfte.file.logger.fileCount	Die Anzahl an Protokolldateien, die maximal erstellt werden. Liegt die Datenmenge über der, die maximal in dieser Anzahl an Dateien gespeichert werden kann, wird die älteste Datei gelöscht. Auf diese Weise überschreitet die Anzahl der Dateien nie den hier angegebenen Wert.	3

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.file.logger.mode	<p>Die Protokollfunktion arbeitet im Modus 'circular' oder 'linear'. Setzen Sie diesen Wert auf CIRCULAR oder LINEAR.</p> <p>CIRCULAR-Die Dateiprotokollfunktion schreibt Informationen in eine Datei, bis diese Datei die mit der Eigenschaft wmqfte.file.logger.fileSize definierte maximale Größe erreicht. Ist dies der Fall, beginnt die Dateiprotokollfunktion mit einer neuen Datei. Die maximale Anzahl Dateien, die in diesem Modus geschrieben werden, wird durch den Wert gesteuert, der über die Eigenschaft wmqfte.file.logger.fileCount definiert wird. Wenn die maximal zulässige Anzahl an Dateien erreicht ist, löscht die Dateiprotokollfunktion die erste Datei und erstellt sie anschließend erneut, um sie als aktive Protokolldatei einzusetzen. Wenn der in der Eigenschaft wmqfte.file.logger.fileSize definierte Wert eine Byteeinheit mit fester Größe (z. B. KB, MB oder GB) ist, entspricht die Obergrenze für den in diesem Modus verwendeten Plattenspeicherplatz dem Wert von <code>fileSize</code> multipliziert mit <code>fileCount</code>. Wenn der in der Eigenschaft wmqfte.file.logger.fileSize definierte Wert eine Zeiteinheit ist (z. B. M, H, oder W), hängt die maximale Größe vom Durchsatz der Protokollnachrichten in Ihrem System über diese Zeiträume ab. Die Namenskonvention für die Protokolldatei, die bei der Ausführung in diesem Modus verwendet wird, lautet <code>Logger_namenumber-timestamp.log</code>. Dabei gilt Folgendes:</p> <ul style="list-style-type: none"> • <i>Name_der_Protokollfunktion</i> ist der Name, der der Protokollfunktion im Befehl fteCreateLogger zugewiesen wird. • <i>Nummer</i> ist die Nummer der Datei innerhalb der Gruppe. • <i>Zeitmarke</i> gibt den Erstellungszeitpunkt der Datei an. <p>Beispiel: <code>LOGGER1-20111216123430147.log</code></p> <p>LINEAR: Die Dateiprotokollfunktion schreibt Informationen in eine Datei, bis diese Datei ihre mit der Eigenschaft wmqfte.file.logger.fileSize definierte maximale Größe erreicht. Ist dies der Fall, beginnt die Dateiprotokollfunktion mit einer neuen Datei. Die zuvor geschriebenen Dateien werden dabei nicht gelöscht, sodass die vorherigen Protokollnachrichten weiterhin vorliegen. Dateien werden bei Ausführung im Modus <code>Linear</code> nicht gelöscht. Daher wird die Eigenschaft wmqfte.file.logger.fileCount ignoriert, da es keine Obergrenze für die Anzahl der Dateien gibt, die erstellt werden können. Da es in diesem Modus keine Obergrenze gibt, muss der von den Protokolldateien belegte Speicherplatz überwacht werden, damit es zu keinen Festplattenspeicherengpässen kommt. Die Namenskonvention für die Protokolldatei, die bei der Ausführung in diesem Modus verwendet wird, lautet <code>logger_name-timestamp.log</code>. Dabei gilt Folgendes:</p> <ul style="list-style-type: none"> • <i>Name_der_Protokollfunktion</i> ist der Name, der der Protokollfunktion im Befehl fteCreateLogger zugewiesen wird. • <i>Zeitmarke</i> gibt den Erstellungszeitpunkt der Datei an. <p>Beispiel: <code>LOGGER-20111216123430147.log</code></p>	Kein Standardwert

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.max.retry.interval	<p>Die maximale Zeit in Sekunden zwischen Wiederholungen, wenn bei der Protokollfunktion ein permanenter Fehler auftritt.</p> <p>Einige Fehlerbedingungen (beispielsweise eine Unterbrechung der Datenbankverbindung) verhindern eine Fortsetzung der Protokollfunktion. Tritt eine solche Bedingung ein, setzt die Protokollfunktion die aktuelle Transaktion zurück, wartet eine bestimmte Zeit und wiederholt dann den Vorgang. Die Zeit, die die Protokollfunktion wartet, ist zu Beginn sehr kurz, sodass vorübergehende Fehler schnell überwunden werden können. Bei jedem neuen Versuch der Protokollfunktion allerdings verlängert sich die Wartezeit. Dies verhindert, dass zu viel unnötige Arbeit ausgeführt wird, wenn die Fehlerbedingung länger andauert, beispielsweise wenn eine Datenbank zur Wartung heruntergefahren wird.</p> <p>Legen Sie über diese Eigenschaft einen Grenzwert für die Wartezeit fest, sodass nach einer angemessenen Zeit, in der die Fehlerbedingung behoben sein sollte, eine Wiederholung stattfindet.</p>	600
immediateShutdownTimeout	<p>Die Zeit in Sekunden, die die Protokollfunktion wartet, bis alle ausstehenden Operationen ordnungsgemäß abgeschlossen und beendet wurden. Standardmäßig wartet die Protokollfunktion 10 Sekunden darauf, dass die Operationen beendet werden. Wenn Operationen nicht vor dem Zeitlimit abgeschlossen werden, schreibt die Protokollfunktion die folgende Ereignisnachricht in <code>output0.log</code> und wird beendet.</p> <p><code>BFGDB0082I: The logger is ending immediately.</code></p> <p>Bei Angabe des Werts 'null' wartet die Protokollfunktion unbegrenzt lange auf den Abschluss aktueller Operationen.</p> <p>Der Standardwert wird verwendet, wenn der Wert von <code>immediateShutdownTimeout</code> kleiner als null ist.</p> <p>Die Eigenschaft gilt sowohl für die eigenständige Datenbankprotokollfunktion als auch für die dateibasierte Protokollfunktion.</p>	10
loggerCredentialsKeyFile	Der Name der Datei, die den Berechtigungsnachweisschlüssel enthält, der beim Verschlüsseln von Berechtigungsnachweisen verwendet wird.	Eine Zeichenfolgeeigenschaft, die keinen Standardwert hat.
loggerQMgrRetryInterval	Das Intervall (in Sekunden), in dem der Prozesscontroller der Protokollfunktion prüft, ob der Warteschlangenmanager verfügbar ist.	30
maxRestartCount	Die Anzahl an Neustarts, die maximal innerhalb des Zeitintervalls möglich sind, das über die Eigenschaft 'maxRestartInterval' angegeben ist. Bei Überschreiten dieses Wertes versucht der Prozesscontroller der Protokollfunktion nicht mehr, die Protokollfunktion neu zu starten, sondern führt die über die Eigenschaft 'maxRestartDelay' vorgegebene Aktion aus.	4

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)




Eigenschaftsname	Beschreibung	Standardwert
maxRestartInterval	Der Zeitraum (in Sekunden), über den hinweg der Prozesscontroller der Protokollfunktion die Anzahl der bereits erfolgten Neustarts der Protokollfunktion zählt. Liegt die Anzahl der Neustart innerhalb dieses Zeitraums über dem für 'maxRestartCount' angegebenen Wert, wird die Protokollfunktion vom Prozesscontroller nicht mehr neu gestartet. Stattdessen führt der Prozesscontroller die über die Eigenschaft 'maxRestartDelay' vorgegebene Aktion aus.	120
maxRestartDelay	Gibt an, wie der Prozesscontroller der Protokollfunktion vorgeht, wenn die Anzahl der Protokollfunktionsneustarts den über die Eigenschaften 'maxRestartCount' und 'maxRestartInterval' angegebenen Wert überschreitet. Wird ein Wert kleiner oder gleich null angegeben, wird der Prozesscontroller der Protokollfunktion gestoppt. Ein Wert größer als null entspricht der Anzahl der Sekunden, nach deren Ablauf die im Prozesscontroller der Protokollfunktion gespeicherten Protokollinformationen zu den Neustarts zurückgesetzt werden und die Protokollfunktion erneut gestartet wird.	-1
wmqfte.oracle.port	Der Port, über den die Protokollfunktion eine Verbindung zur Oracle-Instanz herstellt. Dieser Port wird auch als TNS-Listener bezeichnet.	1521
wmqfte.oracle.host	Der Host, über den die Protokollfunktion eine Verbindung zur Oracle-Instanz herstellt.	localhost
armELEMTYPE	Optionale Eigenschaft. Wenn die Protokollfunktion so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMTYPE, der in der zugehörigen ARM-Richtlinie festgelegt ist. Setzen Sie ELEMTYPE für eine Protokollfunktion auf SYSBFGLG.	Nicht festgelegt
armELEMENT	Optionale Eigenschaft. Wenn die Protokollfunktion so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMENT, der in der zugehörigen ARM-Richtlinie festgelegt ist. Der Wert von ELEMENT kann dem Namen der Protokollfunktion entsprechen.	Nicht festgelegt
loggerQMGrAuthenticationCredentialsFile	Der Pfad der Datei, die die MQ-Berechtigungs-nachweise für die Verbindung mit dem Koordinationswarteschlangenmanager der Protokollfunktion enthält.	<p> Informationen zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter z/OS konfigurieren.</p> <p> Informationen zur Position und den Berechtigungen für diese Datei finden Sie unter MQMFTCredentials.xml konfigurieren.</p> <p> Siehe auch MFT- und IBM MQ-Verbindungsauthentifizierung.</p>

Tabelle 111. Eigenschaften in der Datei 'logger.properties' für Verbindungen im Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
trace	<p>Optionale Eigenschaft. Tracespezifikation, wenn die Protokollfunktion beim Start mit aktiviertem Trace ausgeführt werden soll. Die Tracespezifikation ist eine durch Kommas getrennte Liste mit Klassen, dem Gleichheitszeichen und einer Tracestufe.</p> <p>Beispiel: <code>com.ibm.wmqfte.databaselogger</code> und <code>com.ibm.wmqfte.databaselogger.operation=all</code></p> <p>Sie können auch mehrere Tracespezifikationen in einer durch Doppelpunkte getrennten Liste angeben. Beispiel: <code>com.ibm.wmqfte.databaselogger=moderate:com.ibm.wmqfte.databaselogger.operation=all</code></p>	--
traceFiles	Optionale Eigenschaft. Die Gesamtanzahl der zu speichernden Tracedateien. Dieser Wert gilt für den Prozesscontroller einer Protokollfunktion sowie für die Protokollfunktion selbst.	5
traceSize	Optionale Eigenschaft. Die maximale Größe (in MB) der einzelnen Tracedateien; sobald dieser Wert erreicht ist, wird der Trace in eine Folgedatei geschrieben. Dieser Wert gilt für den Prozesscontroller einer Protokollfunktion sowie für die Protokollfunktion selbst.	20
wmqfte.file.logger.filePermissions	<p>Optionale Eigenschaft. Verwenden Sie diese Eigenschaft, um anzugeben, welche Art von Berechtigung für die Protokolldatei der Protokollfunktion erforderlich ist.</p> <p>Die Eigenschaft gilt sowohl für lineare Protokolle als auch für Umlaufprotokolle und kann die Werte <i>UserReadWriteOnly</i> oder <i>UserReadWriteAllRead</i> annehmen.</p> <p>Der Wert <i>UserReadWriteOnly</i> verfügt über die entsprechende Berechtigung 600 und der Wert <i>UserReadWriteAllRead</i> über die entsprechende Berechtigung 644.</p> <p>Jede Änderung der Berechtigung gilt für neu erstellte Dateien der Protokollfunktion.</p> <p>Wenn Sie einen ungültigen Wert für die Eigenschaft eingeben, verwendet die Protokollfunktion den Standardwert und gibt die Nachricht BFGDB0083W im Ausgabeprotokoll aus.</p>	UserReadWriteOnly

Eigenschaften für SSL/TLS-Verbindungen im Clientmodus

Die Eigenschaften, die zur Unterstützung der Clientmodusverbindung mit einem Warteschlangenmanager für die Protokollfunktion bei Verwendung von SSL/TLS erforderlich sind.

Tabelle 112. Eigenschaften in der Datei 'logger.properties' für SSL/TLS-Verbindungen im Clientmodus

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.queue.manager.host	Hostname oder IP-Adresse des Warteschlangenmanagers für die Protokollfunktion.	Kein Standardwert
wmqfte.queue.manager.port	Der Port, an dem der Warteschlangenmanager der Protokollfunktion empfangsbereit ist.	1414
wmqfte.queue.manager.channel	Name des Serververbindungskanals für den Warteschlangenmanager der Protokollfunktion.	SYSTEM.DEF.SVRCONN

Tabelle 112. Eigenschaften in der Datei 'logger.properties' für SSL/TLS-Verbindungen im Clientmodus (Forts.)



Eigenschaftsname	Beschreibung	Standardwert
wmqfte.Ssl.CipherSuite	<p>Gibt TLS-Aspekte zum Austausch von Daten zwischen der Protokollfunktion und dem Warteschlangenmanager der Protokollfunktion an.</p> <p>Der Wert von wmqfte.Ssl.CipherSuite ist ein CipherSuite-Name. Dieser CipherSuite-Name ist mit dem CipherSpec-Namen identisch, der im Kanal des Warteschlangenmanagers der Protokollfunktion verwendet wird.</p> <p>Weitere Informationen hierzu finden Sie im Abschnitt Namenszuordnungen von CipherSuites und CipherSpecs.</p>	Kein Standardwert
wmqfte.Ssl.PeerName	<p>Gibt den Entwurf eines definierten Namens an, der mit dem vom Warteschlangenmanager der Protokollfunktion bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Warteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.</p>	Kein Standardwert
wmqfte.Ssl.TrustStore	<p>Gibt die Position der Zertifikate an, die die Protokollfunktion akzeptiert. Der Wert von wmqfte.Ssl.TrustStore ist ein Dateipfad.</p> <p>Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden.  Windows</p> <p>Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden.</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.TrustStoreCredentialsFile	<p>Der Pfad zu der Datei, die den Berechtigungsnachweis wmqfte.Ssl.TrustStore enthält</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.TrustStoreType	<p>Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Diese Eigenschaft kann den Wert 'jks' oder 'pkcs12' haben.</p>	jks
wmqfte.Ssl.KeyStore	<p>Gibt die Position des privaten Schlüssels der Protokollfunktion an. Der Wert von wmqfte.Ssl.KeyStore ist ein Dateipfad.</p> <p>Klammern, Kommas (,) und Backslashes (\) sind Sonderzeichen in MFT-Befehlen und müssen mit einem Backslash (\) als Escapezeichen versehen werden.  Windows</p> <p>Dateipfade unter Windows können entweder mit doppelten umgekehrten Schrägstrichen (\\) als Trennzeichen oder mit einfachen Schrägstrichen (/) angegeben werden.</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert

Tabelle 112. Eigenschaften in der Datei 'logger.properties' für SSL/TLS-Verbindungen im Clientmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.Ssl.KeyStore.CredentialsFile	Der Pfad zu der Datei, die den Berechtigungsnachweis wmqfte.Ssl.KeyStore enthält Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.	Kein Standardwert
wmqfte.Ssl.KeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Diese Eigenschaft kann den Wert 'jks' oder 'pkcs12' haben.	jks
wmqfte.Ssl.FipsRequired	Gibt an, dass Sie die FIPS-Unterstützung auf der Ebene der Protokollfunktion aktivieren möchten. Der Wert dieser Eigenschaft kann 'true' oder 'false' sein. Weitere Informationen finden Sie im Abschnitt <u>FIPS-Unterstützung</u> in MFT.	false

Zugehörige Konzepte

SSL/TLS-Eigenschaften für MFT

Zugehörige Verweise

„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174

Es ist möglich, dass Umgebungsvariablen in Managed File Transfer -Eigenschaften verwendet werden, die Datei- oder Verzeichnispositionen darstellen. Dadurch können die Positionen der Dateien oder Verzeichnisse, die bei der Ausführung von Teilen des Produkts verwendet werden, abhängig von der aktuellen Umgebung variieren (z. B. der Benutzer, der einen Befehl ausführt).

„Die MFT agent.properties-Datei“ auf Seite 180

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.properties`, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei `agent.properties` angegeben sein.

„Die MFT-Datei 'command.properties'“ auf Seite 209

In der Datei `command.properties` ist der Befehlswarteschlangenmanager angegeben, zu dem eine Verbindung hergestellt werden muss, wenn Befehle ausgegeben werden. Außerdem enthält die Datei Informationen, die Managed File Transfer für den Kontakt zu diesem Warteschlangenmanager benötigt.

„Die MFT-Datei 'coordination.properties'“ auf Seite 204

Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden.

Von der Funktion 'LogTransfer' erzeugte Ausgabe

In Übertragungsprotokollereignissen werden von der Übergabe bis zum Abschluss einer Übertragung die Details des Verarbeitungsfortschritts bei der Übertragung erfasst. Außerdem werden Informationen zum Übergang einer Übertragung in die Resynchronisierung erfasst, damit Sie den Verarbeitungsfortschritt einer Übertragung nachvollziehen können.

Format von Übertragungsereignissen

Übertragungsereignisse liegen im JSON-Format vor und werden in die Datei `transferlogN.json` geschrieben, die im Protokollverzeichnis des Agenten erstellt wird. Dabei ist keine Zahl, wobei 0 der Standardwert ist. Jedes Ereignis beinhaltet die folgenden allgemeinen Attribute:

- Datum und Uhrzeit (in UTC)
- Eindeutige ID

Abhängig vom Typ des Ereignisses und der Übertragungsprotokollstufe sind in den geschriebenen Ereignisinformationen weitere Attribute enthalten. Während bei der Übertragungsprotokollstufe *Info* nur minimale Informationen geschrieben werden, umfasst die Stufe *verbose* detailliertere Angaben. Im folgenden Abschnitt „Beispielereignisse“ auf Seite 224 sind einige Beispiele für Übertragungsereignisse beschrieben, die von einem Agenten protokolliert werden.

Eindeutige ID

Die eindeutige ID ist enthalten, damit Sie die verschiedenen Phasen im Verlauf einer Übertragung leichter identifizieren können, z. B. BFGTL0001. Die eindeutige ID ist Teil des Attributs **eventDescription** und besteht aus zwei Teilen:

BFGTL

Das für alle IDs verwendete Präfix, wobei BFG das Standardsuffix ist, das in Managed File Transfer verwendet wird, und TL gibt an, dass es sich um ein Übertragungsprotokoll handelt.

Zahl

Eine eindeutige Nummer, die mit 1 beginnt. For example:

```
{
  "eventDescription": "BFGTL0001: New transfer request submitted"
}
```

Beispielereignisse

In der folgenden Tabelle sind einige der Ereignisse als Beispiele für die Informationen beschrieben, die von der zusätzlichen Funktion protokolliert werden. In der Spalte der Tabelle (*Protokollstufe*) ist die Stufe angegeben, bei der das Ereignis protokolliert wird.

Wichtig: Die folgenden Attribute sind in den Ereignisinformationen enthalten, wenn die Stufe **logTransfer** auf *Ausführlich* oder *Moderat* gesetzt ist:

- **sourceAgent**
- **destinationAgent**
- **threadId**

Ereignis	Protokollstufe	Beschreibung
Liste der zu übertragenden Elemente	verbose	<pre>{ "dateTime": "<Data time in UTC>", "eventDescription": "BFGTL0002I: Generated detailed transfer item list.", "destinationAgent": "<Name of destination agent>", "sourceAgent": "<Name of source agent>", "threadId": "0000001d", "totalItemsInTransfer": <Number of items in the transfer>, "transferId": "<Transfer Identifier>", "transferItemsList": [{ "source": "source item name", "destination": "destination item name" }] }</pre> <p>Example:</p> <pre>{ "dateTime": "2022-01-14T12:56:54.219Z UTC", "eventDescription": "BFGTL0002I: Generated detailed transfer item list.", "destinationAgent": "QMBAGQ", "sourceAgent": "QMBAG1", "threadId": "0000001d", "totalItems": 1, "transferId": "414d5120514d4120202020202020202063bd17610a390040", "transferItems": [{ "destination": "/results/rts/target/destFile.txt", "source": "DESTINATIONQ@QMB" }] }</pre>

Ereignis	Protokollebene	Beschreibung
Liste der beim Start des Agenten wiederherzustellenden Übertragungen	verbose	<pre> { "dateTime": "<Date and time in UTC>", "eventDescription": "The list of transfers being recovered as part of agent recovery process.", "agentName": "<Agent name>", "transfers": [{"transferId": "<transfer state>"}] "threadId": "<Thread Id>", } Example: { "dateTime": "2022-01-14T14:42:24.902Z UTC", "eventDescription": "The list of transfers being recovered as part of agent recovery process.", "agentName": "CQMHX01AG1", "transfers": [{"414D512043514D4858303120202020B0D4176101370040": "complete Received"}, {"414D512043514D4858303120202020B0D4176101370050": "resynchronizing"}] "threadId": "0000001c", } </pre>

Zugehörige Verweise

„Java-Systemeigenschaften für MFT“ auf Seite 226

Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

[fteCreateAgent](#)

„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174

Es ist möglich, dass Umgebungsvariablen in Managed File Transfer -Eigenschaften verwendet werden, die Datei- oder Verzeichnispositionen darstellen. Dadurch können die Positionen der Dateien oder Verzeichnisse, die bei der Ausführung von Teilen des Produkts verwendet werden, abhängig von der aktuellen Umgebung variieren (z. B. der Benutzer, der einen Befehl ausführt).

Java-Systemeigenschaften für MFT



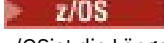
Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

Zur Einstellung der Systemeigenschaften und anderer JVM-Optionen für eine JVM, die Managed File Transfer-Befehle ausführen muss, müssen Sie die Umgebungsvariable BFG_JVM_PROPERTIES definieren. Um auf einer UNIX-Plattform beispielsweise die Eigenschaft 'com.ibm.wmqfte.maxConsoleLineLength' zu setzen, ist die Variable wie folgt zu definieren:

```
export BFG_JVM_PROPERTIES="-Dcom.ibm.wmqfte.maxConsoleLineLength=132"
```

Wenn Sie einen Agenten als Windows-Dienst ausführen, können Sie dessen Java-Systemeigenschaften mit dem Parameter `-sj` im Befehl **fteModifyAgent** ändern.

Tabelle 113. Java-Systemeigenschaften

Eigenschaftsname	Beschreibung	Wert
com.ibm.wmqfte.maxConsoleLineLength	Maximale Zeilenlänge, die in der Konsole ausgegeben werden kann. Zeilen, die diese Länge überschreiten, werden umbrochen. Der Wert wird in Byte (nicht in Zeichen) angegeben.	 Die Standardlänge für IBM i liegt bei 132 Byte.  Für AIX, Linux, and Windows ist die Länge unbegrenzt.  Für z/OS ist die Länge unbegrenzt.
com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm	(Nur Windows). Gibt die maximale Anzahl der aufzubewahrenden Protokolldateien für Windows-Dienste an. Die Protokolldateien für Windows-Dienste werden in den Protokollverzeichnissen des Agenten und der Datenbankprotokollfunktion erstellt, wenn diese Anwendungen als Windows-Dienst ausgeführt werden. Die Dateinamen der Protokolldateien für Windows-Dienste beginnen mit dem Präfix <i>Service</i> . Die Protokolldateien enthalten Nachrichten zum Start- und Stoppvorgang des Dienstes.	5

Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

[Hinweise und Tipps zur Verwendung von MFT](#)

SHA-2-CipherSpecs und -CipherSuites für MFT

Managed File Transfer unterstützt SHA-2-CipherSpecs und -CipherSuites.

Weitere Informationen zu CipherSpecs und CipherSuites, die für Verbindungen zwischen Agenten und IBM MQ -Warteschlangenmanagern verfügbar sind, finden Sie unter [TLS CipherSpecs und CipherSuites in IBM MQ classes for Java](#) und unter [SSL/TLS CipherSpecs und CipherSuites in IBM MQ Classes for JMS](#).

Weitere Informationen zum Konfigurieren von CipherSpecs und CipherSuites für die Verwendung mit den Protokollbridgeagenten (PBAs) und FTPS-Servern finden Sie unter [FTPS-Serverunterstützung durch die Protokollbridge](#) und [Format der Eigenschaftendatei der Protokollbridge](#).

Für die Konformität mit SP 800-131A wird Folgendes vorausgesetzt:

- Sie benötigen ein entsprechend konfiguriertes FTPS; SFTP wird nicht unterstützt.
- Der Remote Server darf nur SP 800-131A-konforme Cipher-Suites senden.

Zugehörige Konzepte

[SSL/TLS-Eigenschaften für MFT](#)

Konfigurationsdateien der MFT-Dateiprotokollfunktion

Neben der Datei `logger.properties` enthält das Konfigurationsverzeichnis einer eigenständigen Managed File Transfer-Dateiprotokollfunktion noch eine XML-Konfigurationsdatei (`FileLoggerFormat.xml`), in der das Format definiert ist, in dem die Dateiprotokollfunktion Nachrichten in die Protokolldatei schreibt. Der Inhalt dieser Datei muss dem in der Datei `FileLoggerFormat.xsd` definierten XML-Schema entsprechen.

Zugehörige Konzepte

[Format der eigenständigen MFT-Dateiprotokollfunktion](#)

Zugehörige Verweise

„Die MFT-Datei 'logger.properties'“ auf Seite 213

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.

„Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion“ auf Seite 228

Definition des Standardprotokolldateiformats für die eigenständige Managed File Transfer-Dateiprotokollfunktion.

„XSD-Format (eigenständiges Dateiprotokollfunktionsformat)“ auf Seite 233

Das Schemaformat einer eigenständigen Dateiprotokollfunktion.

Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion

Definition des Standardprotokolldateiformats für die eigenständige Managed File Transfer-Dateiprotokollfunktion.

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00" xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <callCompleted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false"/>/transaction/action/@time</insert>
          <insert type="user" width="48" ignoreNull="false"/>/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="3" ignoreNull="false"/>/transaction/status/@resultCode</insert>
          <insert type="user" width="0" ignoreNull="false"/>/transaction/agent/@agent</insert>
          <insert type="user" width="0" ignoreNull="false"/>/transaction/agent/@QMgr</insert>
          <insert type="user" width="0" ignoreNull="false"/>/transaction/job/name</insert>
          <insert type="user" width="0" ignoreNull="true"/>/transaction/transferSet/call/command/@ty
pe</insert>
          <insert type="user" width="0" ignoreNull="true"/>/transaction/transferSet/call/com
mand/@name</insert>
          <insert type="system" width="0" ignoreNull="true">callArguments</insert>
          <insert type="user" width="0" ignoreNull="true"/>/transaction/transferSet/call/callRe
sult/@outcome</insert>
          <insert type="user" width="0" ignoreNull="true"/>/transaction/transferSet/call/callResult/re
sult/error</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </callCompleted>
    <callStarted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false"/>/transaction/action/@time</insert>
          <insert type="user" width="48" ignoreNull="false"/>/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="0" ignoreNull="false"/>/transaction/agent/@agent</insert>
          <insert type="user" width="0" ignoreNull="false"/>/transaction/agent/@QMgr</insert>
          <insert type="user" width="0" ignoreNull="false"/>/transaction/job/name</insert>
          <insert type="user" width="0" ignoreNull="true"/>/transaction/transferSet/call/command/@ty
pe</insert>
          <insert type="user" width="0" ignoreNull="true"/>/transaction/transferSet/call/com
mand/@name</insert>
          <insert type="system" width="0" ignoreNull="true">callArguments</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </callStarted>
    <monitorAction>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false"/>/monitorLog/action/@time</insert>
          <insert type="user" width="48" ignoreNull="false"/>/monitorLog/@referenceId</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="3" ignoreNull="false"/>/monitorLog/status/@resultCode</insert>
          <insert type="user" width="0" ignoreNull="false"/>/monitorLog/@monitorName</insert>
          <insert type="user" width="0" ignoreNull="false"/>/monitorLog/monitorAgent/@agent</insert>
          <insert type="user" width="0" ignoreNull="false"/>/monitorLog/monitorAgent/@QMgr</insert>
          <insert type="user" width="0" ignoreNull="false"/>/monitorLog/action</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </monitorAction>
  </messageTypes>
</logFormatDefinition>
```

```

</format>
</monitorAction>
<monitorCreate>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
    </inserts>
    <separator>;</separator>
  </format>
</monitorCreate>
<monitorFired>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/monitorLog/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
      <insert type="user" width="48" ignoreNull="false">/monitorLog/references/taskRequest</insert>
    </inserts>
    <separator>;</separator>
  </format>
</monitorFired>
<notAuthorized>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/notAuthorized/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/notAuthorized/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/notAuthorized/status/@resultCode</insert>
      <insert type="user" width="12" ignoreNull="false">/notAuthorized/action</insert>
      <insert type="user" width="12" ignoreNull="false">/notAuthorized/authority</insert>
      <insert type="user" width="0" ignoreNull="false">/notAuthorized/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="false">/notAuthorized/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</notAuthorized>
<scheduleDelete>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
      <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
      <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</scheduleDelete>
<scheduleExpire>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
      <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
      <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</scheduleExpire>
<scheduleSkipped>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>

```

```

        <insert type="system" width="6" ignoreNull="false">type</insert>
        <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
        <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
        <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
        <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
        <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
    </inserts>
    <separator>;</separator>
</format>
</scheduleSkipped>
<scheduleSubmitInfo>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="6" ignoreNull="false">type</insert>
            <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
            <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
            <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/schedule/submit</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/schedule/submit/@timezone</in□
sert>
            <insert type="user" width="3" ignoreNull="true">/schedulelog/schedule/repeat/frequency</in□
sert>
            <insert type="user" width="12" ignoreNull="true">/schedulelog/schedule/repeat/frequency/@in□
terval</insert>
            <insert type="user" width="3" ignoreNull="true">/schedulelog/schedule/repeat/expire□
Count</insert>
            <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
    </format>
</scheduleSubmitInfo>
<scheduleSubmitTransfer>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="10" ignoreNull="false">type</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@QMgr |
/transaction/destinationWebUser/@webGatewayAgentQMgr |
/transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
        </inserts>
        <separator>;</separator>
    </format>
</scheduleSubmitTransfer>
<scheduleSubmitTransferSet>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
            <insert type="system" width="10" ignoreNull="false">type</insert>
            <insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
            <insert type="user" width="5" ignoreNull="true">source/@type</insert>
            <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
            <insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</in□
sert>
            <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
            <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
        </inserts>
        <separator>;</separator>
    </format>
</scheduleSubmitTransferSet>
<transferStarted>
    <format>
        <inserts>
            <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
            <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
            <insert type="system" width="6" ignoreNull="false">type</insert>
            <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
            <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |

```



```

</format>
</transferComplete>
<transferDelete>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
      /transaction/sourceWebUser/@webGatewayAgentName |
      /transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMgr |
      /transaction/sourceWebUser/@webGatewayAgentQMgr |
      /transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
      /transaction/sourceWebUser/@webGatewayAgentType |
      /transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
      /transaction/destinationWebUser/@webGatewayAgentName |
      /transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMgr |
      /transaction/destinationWebUser/@webGatewayAgentQMgr |
      /transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@agentType |
      /transaction/destinationWebUser/@webGatewayAgentType |
      /transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferDelete>
<transferProgress>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
      <insert type="user" width="0" ignoreNull="false">source/file/@size | source/queue/@size</in
sert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="true">source/file/@alias | source/queue/@ali
as</insert>
      <insert type="user" width="0" ignoreNull="true">source/file/@filespace | source/queue/@file
space</insert>
      <insert type="user" width="0" ignoreNull="true">source/@correlationBoolean1</insert>
      <insert type="user" width="0" ignoreNull="true">source/@correlationNum1</insert>
      <insert type="user" width="0" ignoreNull="true">source/@correlationString1</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</in
sert>
      <insert type="user" width="0" ignoreNull="false">destination/file/@size | destination/queue/
@size</insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
      <insert type="user" width="0" ignoreNull="true">destination/file/@alias | destination/queue/
@alias</insert>
      <insert type="user" width="0" ignoreNull="true">destination/file/@filespace | destinati
on/queue/@filespace</insert>
      <insert type="user" width="0" ignoreNull="true">destination/file/@truncateRecords</insert>
      <insert type="user" width="0" ignoreNull="true">destination/@correlationBoolean1</insert>
      <insert type="user" width="0" ignoreNull="true">destination/@correlationNum1</insert>
      <insert type="user" width="0" ignoreNull="true">destination/@correlationString1</insert>
      <insert type="user" width="0" ignoreNull="true">status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferProgress>
</messageTypes>
</logFormatDefinition>

```

Zugehörige Verweise

[Format der eigenständigen MFT-Dateiprotokollfunktion](#)

„XSD-Format (eigenständiges Dateiprotokollfunktionsformat)” auf Seite 233

Das Schemaformat einer eigenständigen Dateiprotokollfunktion.

XSD-Format (eigenständiges Dateiprotokollfunktionsformat)

Das Schemaformat einer eigenständigen Dateiprotokollfunktion.

Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
@start_non_restricted_prolog@
Version: %Z% %I% %W% %E% %U% [%H% %T%]

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2011, 2024. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
@end_non_restricted_prolog@
-->

<!--
This schema defines the format of the FileLoggerFormat XML file that contains the definition
of the format to use when logging FTE log messages to a file. When an XML file that conforms
to this schema is processed by a file logger it can contain definitions for one or more
message type(s) that define how log messages of those types are output to the file log.
-->

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
<xsd:include schemaLocation="fteutils.xsd"/>

  <!--
    Defines the logFileDefinition and version number
    <logFileDefinition version="1.00" ...
      <messageTypes>
        ...
      </messageTypes>
    </logFileDefinition>
  -->
  <xsd:element name="logFileDefinition">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="messageTypes" type="messageTypesType" maxOccurs="1" minOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <!--
    Defines the set of accepted message types. The definition of individual message types
    is optional. If a particular types element is present but empty then no line will be
    output for messages of that type. If a particular types element is not present then
    the default format will be used to format messages of that type.
  -->
  <xsd:complexType name="messageTypesType">
    <xsd:sequence>
      <xsd:element name="callCompleted" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="callStarted" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="monitorAction" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="monitorCreate" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="monitorFired" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="notAuthorized" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="scheduleDelete" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="scheduleExpire" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="scheduleSkipped" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
      <xsd:element name="scheduleSubmitInfo" type="messageType" maxOccurs="1" minOccurs="0"
        curs="0" />
    </xsd:sequence>
  </xsd:complexType>

```

```

    <xsd:element name="scheduleSubmitTransfer" type="messageType" maxOccurs="1" minOccurs="0" />
    <xsd:element name="scheduleSubmitTransferSet" type="messageType" maxOccurs="1" minOccurs="0" />
    <xsd:element name="transferStarted" type="messageType" maxOccurs="1" minOccurs="0" />
    <xsd:element name="transferCancelled" type="messageType" maxOccurs="1" minOccurs="0" />
    <xsd:element name="transferComplete" type="messageType" maxOccurs="1" minOccurs="0" />
    <xsd:element name="transferDelete" type="messageType" maxOccurs="1" minOccurs="0" />
    <xsd:element name="transferProgress" type="messageType" maxOccurs="1" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of a message type definition e.g.

  <callStarted>
  <format>
  ...
  </format>
  <callStarted>
-->
<xsd:complexType name="messageType">
  <xsd:sequence>
    <xsd:element name="format" type="messageFormatType" maxOccurs="1" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of a message format definition e.g.

  <format>
  <inserts>
  ...
  </inserts>
  <separator>;</separator>
  </format>
-->
<xsd:complexType name="messageFormatType">
  <xsd:sequence>
    <xsd:element name="inserts" type="insertsType" maxOccurs="1" minOccurs="1" />
    <xsd:element name="separator" type="scheduleType" maxOccurs="1" minOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of the inserts element e.g.

  <inserts>
  <insert ...>
  <insert ...>
  ...
  </inserts>
-->
<xsd:complexType name="insertsType">
  <xsd:sequence>
    <xsd:element name="insert" type="insertType" maxOccurs="unbounded" minOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of an insert definition e.g.

  <insert type="user" width="0" ignoreNull="true">/transaction/@ID</insert>
-->
<xsd:complexType name="insertType">
  <xsd:attribute name="type" type="insertTypeType" use="required" />
  <xsd:attribute name="width" type="xsd:nonNegativeInteger" use="required" />
  <xsd:attribute name="ignoreNull" type="xsd:boolean" use="required" />
</xsd:complexType>

<!--
  Defines the accepted choices for the insert type attribute.
-->
<xsd:simpleType name="insertTypeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="user" />
    <xsd:enumeration value="system" />
  </xsd:restriction>
</xsd:simpleType>

```

```

    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Zugehörige Verweise

Format der eigenständigen MFT-Dateiprotokollfunktion

„Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion“ auf Seite 228

Definition des Standardprotokolldateiformats für die eigenständige Managed File Transfer-Dateiprotokollfunktion.

z/OS Die Bibliothek SCSQFCMD

Die von IBM MQ Managed File Transfer for z/OS bereitgestellte Bibliothek SCSQFCMD enthält Member, die als Vorlagen für Jobs dienen, die zum Erstellen einer Managed File Transfer -Konfiguration sowie zum Erstellen und Verwalten eines Agenten oder einer Protokollfunktion verwendet werden können.

Die Inhalte der Bibliothek werden in der folgenden Tabelle angezeigt.

Mitglied	Beschreibung
BFGCOPY	Job, der zum Erstellen einer Kopie der Bibliothek SCSQFCMD verwendet wird.
BFGCUSTM	Job, mit dem eine Kopie der Bibliothek für einen Agenten oder eine Protokollfunktion angepasst wird
BFGXCROB	Beispielvorlage fteObfuscate
BFGXLGCR	Vorlage fteCreateLogger
BFGXMNCR	Beispielvorlage fteCreateMonitor
BFGXMNDE	Beispielvorlage fteDeleteMonitor
BFGXPRAN	Beispielvorlage fteAnt
BFGXSTDE	Beispielvorlage fteDeleteScheduledTransfer
BFGXTMCR	Beispielvorlage fteCreateTemplate
BFGXTMDE	Beispielvorlage fteDeleteTemplate
BFGXTRCA	Beispielvorlage fteCancelTransfer
BFGXTRCR	Beispielvorlage fteCreateTransfer
BFGYAGST	Vorlage für eine gestartete Taskprozedur zum Starten eines Agenten
BFGYLGST	Vorlage für eine gestartete Taskprozedur zum Starten einer Protokollfunktion
BFGZAGCL	Beispielvorlage fteCleanAgent
BFGZAGCR	Beispielvorlage fteCreateAgent
BFGZAGDE	Beispielvorlage fteDeleteAgent
BFGZAGLG	Beispielvorlage fteSetAgentLogLevel
BFGZAGLI	Beispielvorlage fteListAgents
BFGZAGPI	Beispielvorlage ftePingAgent
BFGZAGSH	Beispielvorlage fteShowAgentDetails
BFGZAGSP	Beispielvorlage fteStopAgent
BFGZAGST	Beispielvorlage fteStartAgent
BFGZAGTC	Beispielvorlage fteSetAgentTraceLevel

Mitglied	Beschreibung
BFGZCFCR	Beispielvorlage fteSetupCoordination
BFGZCFDF	Beispielvorlage fteChangeDefaultConfigurationOptions
BFGZCMCR	Beispielvorlage fteSetupCommands
BFGZCMD	Schablone für REXX-Script, das von anderen Mitgliedern in der Datei verwendet wird.
BFGZLGDE	Beispielvorlage fteDeleteLogger
BFGZLGSH	Beispielvorlage fteShowLoggerDetails
BFGZLGSP	Beispielvorlage fteStopLogger
BFGZLGST	Beispielvorlage fteStartLogger
BFGZLGTC	Beispielvorlage fteSetLoggerTraceLevel
BFGZMNL	Beispielvorlage fteListMonitors
BFGZPID	Beispielvorlage fteSetProductId
BFGZPROF	Schablone für Shell-Script, das von anderen Mitgliedern in der Datei verwendet wird.
BFGZPRSH	Beispielvorlage fteDisplayVersion
BFGZRAS	Beispielvorlage fteRas
BFGZSTLI	Beispielvorlage fteListScheduledTransfers
BFGZTMLI	Beispielvorlage fteListTemplates

Details zur Verwendung der Bibliothek SCSQFCMD zum Generieren einer neuen Bibliothek für die Erstellung einer Managed File Transfer -Konfiguration und zum Erstellen und Verwalten eines Agenten oder einer Protokollfunktion finden Sie im Abschnitt [Managed File Transfer for z/OS konfigurieren](#).

Zugehörige Verweise

„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 174

Es ist möglich, dass Umgebungsvariablen in Managed File Transfer -Eigenschaften verwendet werden, die Datei- oder Verzeichnispositionen darstellen. Dadurch können die Positionen der Dateien oder Verzeichnisse, die bei der Ausführung von Teilen des Produkts verwendet werden, abhängig von der aktuellen Umgebung variieren (z. B. der Benutzer, der einen Befehl ausführt).

Thema 'SYSTEM.FTE'

Das SYSTEM.FTE ist ein Thema im Koordinationswarteschlangenmanager, das von Managed File Transfer verwendet wird, um Übertragungen zu protokollieren und Informationen zu Agenten, Überwachungen, Zeitplänen und Vorlagen zu speichern.

Themenstruktur

```

SYSTEM.FTE
  /Agents
    /agent_name
  /monitors
    /agent_name
  /Scheduler
    /agent_name
  /Templates
    /template_ID
  /Transfers
    /agent_name
    /transfer_ID

```

```
/Log
  /agent_name
    /Monitors
    /schedule_ID
    /transfer_ID
```

SYSTEM.FTE/Agents/Agentenname

Dieser Abschnitt enthält eine ständige Veröffentlichung mit der Beschreibung eines Agenten im Managed File Transfer-Netz und dessen Eigenschaften. Die Nachricht zu diesem Thema wird regelmäßig entsprechend dem Agentenstatus aktualisiert. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Agentenstatusnachricht](#).

SYSTEM.FTE/monitors/Agentenname

Dieses Thema enthält eine ständige Veröffentlichung mit der Beschreibung der Ressourcenüberwachungen, die dem Agenten *Agentenname* zugeordnet sind. Die XML-Struktur der ständigen Veröffentlichung entspricht dem Schema `MonitorList.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Überwachungslistennachricht](#).

SYSTEM.FTE/Scheduler/Agentenname

Dieses Thema enthält eine ständige Veröffentlichung mit der Beschreibung aller aktiven Zeitpläne, die dem Agenten *Agentenname* zugeordnet sind. Die XML-Struktur der ständigen Veröffentlichung entspricht dem Schema `ScheduleList.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Planungslistennachricht](#).

SYSTEM.FTE/Templates

Dieses Thema enthält eine ständige Veröffentlichung mit einer Beschreibung aller Vorlagen, die in Ihrer Managed File Transfer-Topologie definiert sind.

- Die Veröffentlichung, die jeder Vorlage zugeordnet ist, wird in einem Unterthema mit dem Namen `SYSTEM.FTE/Templates/template_ID` veröffentlicht.

Ein Beispiel für den Inhalt dieser ständigen Veröffentlichung finden Sie im Abschnitt [MFT-Beispielvorlage XML-Nachricht](#).

SYSTEM.FTE/Transfers/Agentenname

Dieses Thema enthält Veröffentlichungen mit der Beschreibung des Status von Übertragungen, deren Ausgangspunkt der Agent *Agentenname* ist. Die Veröffentlichungen, die jeder Übertragung zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Transfers/agent_name/transfer_ID` veröffentlicht. Mit diesen Veröffentlichungen stellt das IBM MQ Explorer-Plug-in Informationen zum Fortschritt der einzelnen Übertragungen bereit. Die XML-Struktur der Veröffentlichung entspricht dem Schema `TransferStatus.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für Dateiübertragungsstatusnachricht](#).

SYSTEM.FTE/Log/Agentenname

Dieses Thema enthält Veröffentlichungen, in denen Informationen zu Übertragungen, Überwachungen und Zeitplänen protokolliert sind, deren Ausgangspunkt der Agent *Agentenname* ist. Diese Veröffentlichungen können von der Datenbankprotokollfunktion aufgezeichnet werden; damit stehen Überwachungsdatensätze zu den im Managed File Transfer-Netz auftretenden Ereignissen zur Verfügung.

- Die Veröffentlichungen, die jeder Übertragung zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Log/agent_name/transfer_ID` veröffentlicht und die XML der Veröffentlichung entspricht dem Schema `TransferLog.xsd`. Weitere Informationen finden Sie unter [Formate Dateiübertragungsprotokollnachricht](#).
- Die Veröffentlichungen, die jeder geplanten Übertragung zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Log/agent_name/schedule_ID` veröffentlicht und die XML der Veröffentlichung entspricht dem Schema `ScheduleLog.xsd`. Weitere Informationen finden Sie im Abschnitt [Formate für geplante Dateiübertragungsprotokollnachricht](#).
- Die Veröffentlichungen, die den einzelnen Überwachungen zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Log/agent_name/Monitors/monitor_name/monitor_ID` veröffentlicht und die XML der Veröffentlichung entspricht dem Schema `MonitorLog.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Überwachungsprotokollnachricht](#).

Einstellungen von MFT-Agentenwarteschlangen

Die über den Befehl **fteCreateAgent** generierten MQSC-Befehlsscripts erstellen die Agentenwarteschlangen, wobei die Parameter auf die folgenden Werte gesetzt sind. Wenn Sie die zur Erstellung der Warteschlangen bereitgestellten MQSC-Scripts nicht verwenden, sondern die Warteschlangen manuell erstellen, müssen die folgenden Parameter unbedingt auf die angegebenen Werte gesetzt werden.

Betriebswarteschlangen des Agenten

Die Betriebswarteschlangen des Agenten haben folgende Namen:

- SYSTEM.FTE.COMMAND.*Agentenname*
- SYSTEM.FTE.DATA.*Agentenname*
- SYSTEM.FTE.EVENT.*Agentenname*
- SYSTEM.FTE.REPLY.*Agentenname*
- SYSTEM.FTE.STATE.*Agentenname*

Parameter	Wert (falls anwendbar)
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
NUTZUNG	NORMAL
REPLACE	

Berechtigungswarteschlangen des Agenten

Die Berechtigungswarteschlangen des Agenten haben folgende Namen:

- SYSTEM.FTE.AUTHADM1.*Agentenname*
- SYSTEM.FTE.AUTHAGT1.*Agentenname*
- SYSTEM.FTE.AUTHMON1.*Agentenname*
- SYSTEM.FTE.AUTHOPS1.*Agentenname*
- SYSTEM.FTE.AUTHSCH1.*Agentenname*
- SYSTEM.FTE.AUTHTRN1.*Agentenname*

Parameter	Wert (falls anwendbar)
DEFPRTY	0
DEFSOPT	SHARED

<i>Tabelle 115. Parameter für die Berechtigungswarteschlangen des Agenten (Forts.)</i>	
Parameter	Wert (falls anwendbar)
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
NUTZUNG	NORMAL
REPLACE	

Zugehörige Verweise

[fteCreateAgent \(MFT-Agenten erstellen\)](#)

MFT-Systemwarteschlangen und der Systemabschnitt

Managed File Transfer weist zahlreiche Systemwarteschlangen und einen Systemabschnitt auf, die nur für die interne Verwendung vorgesehen sind.

Alle Warteschlangen, deren Name mit SYSTEM.FTE beginnt, sind interne Systemwarteschlangen für Managed File Transfer (MFT). Löschen Sie diese Warteschlangen nicht, da dies die ordnungsgemäße Funktion von IBM MQ MFT verhindert. [Tabelle 116 auf Seite 239](#) zeigt, welcher Nachrichtentyp sich in jeder Warteschlange befindet:

<i>Tabelle 116. Warteschlangennamen, -typ und -verwendung</i>		
Warteschlangename	Warteschlangentyp	Verwendung
SYSTEM.FTE.AU-THAGT1.agent_name	Berechtigung	Warteschlange zum Konfigurieren der Berechtigung zum Senden und Empfangen von Übertragungsanforderungen.
SYSTEM.FTE.AUTHTRN1.agent_name	Berechtigung	Warteschlange zum Konfigurieren der Berechtigung zum Starten und Abbrechen verwalteter Übertragungen. Auch zum Starten verwalteter Aufrufe.
SYSTEM.FTE.AUTHMON1.agent_name	Berechtigung	Warteschlange für die Konfiguration der Berechtigung, mit der ein Benutzer Ressourcenüberwachungen erstellen oder löschen kann, die von demselben Benutzer erstellt wurden.

Tabelle 116. Warteschlangennamen, -typ und -verwendung (Forts.)

Warteschlangenname	Warteschlangentyp	Verwendung
SYSTEM.FTE.AU-THOPS1.agent_name	Berechtigung	Warteschlange zum Konfigurieren der Berechtigung zum Löschen von Ressourcenüberwachungen und geplanten Übertragungen, die von einem anderen Benutzer erstellt wurden.
SYSTEM.FTE.AUTHSCH1.agent_name	Berechtigung	Warteschlange für die Konfiguration der Berechtigung zum Erstellen oder Löschen geplanter Übertragungen, die von demselben Benutzer erstellt wurden
SYSTEM.FTE.AUTHADM1.agent_name	Berechtigung	Warteschlange für die Konfiguration der Berechtigung zum Beenden des Agenten mit der Option -m im Befehl fteStopAgent .
SYSTEM.FTE.COMMAND.agent_name	Operation	Warteschlange zum Senden von Befehlsanforderungen an einen Agenten.
SYSTEM.FTE.DATA.agent_name	Operation	Warteschlange, die von einem Zielagenten zum Halten von Daten verwendet wird, die von einem Quellenagenten gesendet wurden.
SYSTEM.FTE.REPLY.agent_name	Operation	Warteschlange für den Empfang von Antworten von einem Zielagenten.
SYSTEM.FTE.STATE.agent_name	Operation	Warteschlange zum Halten des Status einer Übertragungsanforderung.
SYSTEM.FTE.EVENT.agent_name	Operation	Warteschlange für das Ressourcenüberwachungsprotokoll.
SYSTEM.FTE.HA.agent_name	Operation	Warteschlange, die von hoch verfügbaren Agenteninstanzen als Sperre verwendet wird.

Wenn ein Agent an Übertragungen aus Nachrichten in Dateien oder aus Dateien in Nachrichten teilnimmt, muss möglicherweise die Definition der `SYSTEM.FTE.STATE.agent_name`-Warteschlange geändert werden, damit diese verwalteten Übertragungen stattfinden können. Weitere Informationen hierzu finden Sie in [Anleitung zum Festlegen von MQ-Attributen und MFT-Eigenschaften, die der Nachrichtengröße zugeordnet sind](#).



Achtung: Die Definitionen der anderen Systemwarteschlangen dürfen nicht geändert werden.

Außerdem darf das Thema `SYSTEM.FTE` nicht geändert oder gelöscht werden, da es auch nur zur internen Verwendung bestimmt ist.

Temporäre Warteschlangen

Managed File Transfer erstellt zu mehreren Zwecken temporäre Warteschlangen. Der Name jeder Warteschlange beginnt mit `WMQFTE`. Standardmäßig. (Der Punkt ist Teil des Standardpräfixes.) Wenn Sie dieses Präfix ändern wollen, können Sie die Eigenschaft `dynamicQueuePrefix` in der Datei `command.pro`

properties und/oder in der Datei `coordination.properties` verwenden. Mit der Eigenschaft in der Datei `command.properties` wird das Präfix der temporären Warteschlangen festgelegt, die für Antworten auf Befehle erstellt werden, bei denen eine Antwort vom Agenten erforderlich ist. Die Eigenschaft in der Datei `coordination.properties` wird verwendet, um das Präfix von temporären Warteschlangen festzulegen, die für andere Zwecke erstellt werden. Beispiel: `WMQFTE.FTE.TIMECHCK.QUEUE`, wobei `WMQFTE` ist der Wert, der durch die Eigenschaft **dynamicQueuePrefix** definiert wird.

Zugehörige Verweise


Benutzerberechtigungen für MFT-Agentenaktionen beschränken

Konventionen zum Benennen von MFT-Objekten

Verwenden Sie zum Benennen Ihrer Managed File Transfer-Objekte folgende Konventionen:

- Agenten- und Protokollnamen:
 - Dürfen maximal 28 Zeichen lang sein, wobei die Groß-/Kleinschreibung keine Rolle spielt.
 - Namen, die in Kleinschreibung oder in gemischter Groß-/Kleinschreibung eingegeben werden, werden in Großbuchstaben umgewandelt.
 - Müssen den Standardkonventionen zum Benennen von IBM MQ-Objekten entsprechen.
Diese Konventionen werden im Abschnitt Regeln für die Benennung von IBM MQ-Objekten näher beschrieben.
- Zusätzlich zu den IBM MQ-Konventionen für die Benennung von Objekten muss auch beachtet werden, dass:
 - in Agenten- und Protokollnamen keine Schrägstriche (/) verwendet werden dürfen.
 - in Agenten- und Protokollnamen keine Prozentzeichen (%) verwendet werden dürfen.
- Bei den Namen der Eigenschaften in den Eigenschaftendateien muss die Groß-/Kleinschreibung beachtet werden.
- Bei den Warteschlangenmanagernamen muss die Groß-/Kleinschreibung beachtet werden.
- Auf einigen Plattformen muss bei den Dateinamen die Groß-/Kleinschreibung beachtet werden.
- Ressourcenmonitor- und Übertragungsschablonennamen:
 - Groß-/Kleinschreibung muss nicht beachtet werden.
 - Namen, die in Kleinschreibung oder in gemischter Groß-/Kleinschreibung eingegeben werden, werden in Großbuchstaben umgewandelt.
 - Die Namen dürfen keinen Stern (*), keine Prozentzeichen (%) oder keine Fragezeichen (?) enthalten.
- Protokolldateiservernamen müssen:
 - mindestens zwei Zeichen enthalten und sind in der Länge nicht begrenzt.
 - Groß-/Kleinschreibung muss nicht beachtet werden.
 - Müssen den Standardkonventionen zum Benennen von IBM MQ-Objekten entsprechen.
Diese Konventionen werden im Abschnitt Regeln für die Benennung von IBM MQ-Objekten näher beschrieben.

Dateien im integrierten Dateisystem (Integrated File System, IFS) von IBM i

 Dateinamen im IFS dürfen keine der folgenden Zeichen enthalten:

- Umgekehrter Schrägstrich (\)
- Schrägstrich (/)
- Doppelpunkt (:)
- Stern (*)
- Fragezeichen (?)
- Anführungszeichen (")

- Kleiner-als-Zeichen (<)
- Größer-als-Zeichen (>)
- Vertikaler Balken (|)

Wenn Sie versuchen, Dateien in das integrierte Dateisystem von IBM i zu übertragen, deren Namen diese Zeichen enthalten, schlägt die Übertragung fehl.

Namen von Datasets

z/OS Die Benennung von Datasets unterliegt einigen Beschränkungen, die sich auf die maximale Länge des Namens und die Zeichen beziehen, die für die Namen von Datasets verwendet werden können. Die Namen von Members partitionierter Dateien dürfen maximal acht Zeichen umfassen, das Punktzeichen (.) ist nicht zulässig. Bei der Übertragung in ein Dataset muss der Name explizit angegeben werden; diese Namenseinschränkungen stellen also kein Problem dar. Wenn Sie jedoch eine Übertragung aus Dateien in Mitglieder der partitionierten Datei vornehmen, wird der Dateipfad möglicherweise keinem Mitgliedsnamen der partitionierten Datei zugeordnet. Bei einer Übertragung in eine partitionierte Datei (PDS-Dataset) wird jede Quelldatei zu einem Member der partitionierten Datei und die Namen der einzelnen Members werden auf der Basis des Quellennamens generiert.

Bei den Membernamen der partitionierten Datei handelt es sich um nicht qualifizierte z/OS-Namen, die durch den folgenden regulären Ausdruck definiert werden:

```
[a-zA-Z$#@] [a-zA-Z0-9$#@]{0-7}
```

Mit dem folgenden Schema wird ein Quellendataset oder ein Quellendateiname in einen gültigen Mitgliedsnamen der partitionierten Datei konvertiert. Hierbei gelten diese Aspekte in der folgenden Reihenfolge:

1. Es werden nur die Zeichen im Namen verwendet, die auf den letzten Schrägstrich (/), den letzten Backslash (\) oder den letzten Doppelpunkt (:) folgen. Es wird also nur der Namensbereich eines Dateipfads verwendet.
2. Bei Quellendateien (nicht bei Datasets oder Members partitionierter Dateien) werden alle Zeichen ab dem letzten Punktzeichen (.) ignoriert, und zwar einschließlich des Punkts.
3. Umfassen Namen mehr als acht Zeichen, werden nur die letzten acht Zeichen verwendet.
4. Punkte werden durch kommerzielle A-Zeichen (@) ersetzt.
5. Ungültige Zeichen werden durch kommerzielle A-Zeichen (@) ersetzt.
6. Wenn die Konvertierung keine Zeichen ergibt, lautet das Member der partitionierten Datei @.

Statusnachrichten von MFT-Agenten

Hochverfügbarkeitsagenten veröffentlichen Statusinformationen im XML-Format.

Beispiel-XML mit Informationen zu drei Standby-Instanzen

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentStandbyStatus version="6.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" xsi:noNamespa
ceSchemaLocation="AgentStandbyStatus.xsd">
  <instance host="9.122.123.124" agentVersion="9.1.4.0" />
  <instance host="agenthost.ibm.com" agentVersion="9.1.4.0" />
  <instance host="10.11.12.14" agentVersion="9.1.4.0" />
</AgentStandby>
```

Veröffentlichung des Agentenstatus mit integrierter Standby-Status-XML

Die Standby-Status-XML wird in Fettdruck angezeigt.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<properties version="1.0">
  <entry key="SourceTransferStates"/>
  <entry key="queueManagerPort">1414</entry>
  <entry key="agentStandbyInstances">&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;AgentStandbyStatus
version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="AgentStandbyStatus.xsd"&gt;&lt;Instances&gt;&lt;instance
host="9.122.123.124"
  agentVersion="9.1.4.0" /&gt;&lt;instance host="agenthost.ibm.com" agentVersi
on="9.1.4.0" /&gt;&lt;/Instances&gt;&lt;/AgentStandbyStatus&gt;&lt;/entry>
  <entry key="agentType">STANDARD</entry>
  <entry key="agentDeclaredHostName">MFTHA1</entry>
  <entry key="agentDescription"/>
  <entry key="maxQueuedTransfers">1000</entry>
  <entry key="agentTimeZone">America/Los_Angeles</entry>
  <entry key="agentOsName">Windows Server 2012 R2</entry>
  <entry key="PublishTimeUTC">2019-05-22T06:02:50Z</entry>
  <entry key="queueManagerHost">localhost</entry>
  <entry key="AgentStartTimeUTC">2019-05-22T04:13:02Z</entry>
  <entry key="agentTraceLevel">&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;
agentTraceStatus version="6.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="AgentTraceStatus.xsd"&gt;&lt;trace
level="all"&gt;com.ibm.wmqfte&lt;/trace&gt;&lt;/agentTraceStatus&gt;&lt;/entry>
  <entry key="DestinationTransferStates"/>
  <entry key="queueManager">MFTHAQM</entry>
  <entry key="agentProductVersion">9.1.4.0</entry>
  <entry key="AgentStatusPublishRate">300</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="AgentStatus">STARTED</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="agentName">SRC</entry>
  <entry key="CommandTimeUTC">2019-05-22T06:02:50Z</entry>
  <entry key="queueManagerChannel">MFT_HA_CHN</entry>
  <entry key="agentInterfaceVersion">6.00</entry>
  <entry key="agentVersion">p914-L191119</entry>
</properties>

```

Zugehörige Verweise

[fteCreateAgent](#)

[Agent, GET](#)

IBM MQ Internet Pass-Thru -Konfigurationsreferenz

IBM MQ Internet Pass-Thru (MQIPT) verwendet eine Konfigurationsdatei namens `mqipt.conf`, um Routen zu definieren und die Aktionen des MQIPT-Servers zu steuern. Ab IBM MQ 9.2 können die Konfigurationseigenschaften für den Befehl `mqiptAdmin` auch in einer Eigenschaftendatei angegeben werden.

Die MQIPT-Konfigurationsdatei

Die MQIPT-Konfigurationsdatei enthält eine Reihe von Abschnitten. Es gibt einen Abschnitt `[global]` und einen zusätzlichen Abschnitt `[route]` für jede Route, die über MQIPT definiert wurde.

Jeder Abschnitt enthält Name/Wert-Eigenschaftspaare. Einige Eigenschaften können nur im Abschnitt `[global]` angezeigt werden, einige können nur in den `[route]`-Abschnitten angezeigt werden, und einige können sowohl im Abschnitt `[route]` als auch in `[global]` angezeigt werden. Wenn eine Eigenschaft sowohl im Abschnitt "route" als auch im Abschnitt `[global]` angezeigt wird, überschreibt der Wert der Eigenschaft im Abschnitt `[route]` den globalen Wert, aber nur für die betreffende Route. Auf diese Weise kann der Abschnitt `[global]` verwendet werden, um die Standardwerte festzulegen, die für die Eigenschaften verwendet werden sollen, die nicht in den einzelnen `[route]`-Abschnitten festgelegt sind.

Der Abschnitt `[global]` beginnt mit einer Zeile, in der die Zeichenfolge `[global]` steht, und endet am Beginn des ersten `[route]`-Abschnitts. Der Abschnitt `[global]` muss in der Datei vor allen Abschnitten `[route]` angegeben sein.

Jeder `[route]`-Abschnitt beginnt mit einer Zeile, die die Zeichenfolge `[route]` enthält, und endet, wenn der nächste `[route]`-Abschnitt beginnt oder wenn das Ende der Konfigurationsdatei erreicht ist.

Nicht erkannte Eigenschaftsnamen werden ignoriert. Wenn eine Eigenschaft in einem `[route]`-Abschnitt einen anerkannten Namen, aber keinen gültigen Wert hat (z. B. `MinConnectionThreads=x` oder

HTTP=unsure), ist diese Route inaktiviert (d. h. sie ist nicht empfangsbereit für eingehenden Verbindungen).



Achtung: Der Maximalwert für die Anzahl der Routen, die in der Datei `mqipt.conf` hinzugefügt werden können, ist 100.

Ungültige Werte für Eigenschaften im Abschnitt `[global]` verhindern unter Umständen, dass MQIPT oder der Befehlsserver gestartet wird. Wenn der Befehlsserver nicht gestartet wird, ist MQIPT nicht empfangsbereit für Verwaltungsbefehle, die mit dem Befehl **mqiptAdmin** an den betroffenen Befehlssport gesendet werden. Wenn Eigenschaften mit ungültigen Werten im Abschnitt `[global]` vorhanden sind, wenn MQIPT aktualisiert wird, wird eine Warnung ausgegeben, und der effektive Wert der Eigenschaft bleibt unverändert. Dadurch wird verhindert, dass ungültige Eigenschaftswerte die Beendigung einer aktiven Instanz von MQIPT verursachen, wenn sie aktualisiert wird.

Wenn für eine Eigenschaft die Werte `true` oder `false` aufgelistet sind, kann eine beliebige Mischung aus Groß- und Kleinbuchstaben im Eigenschaftswert verwendet werden.

Sie können den Wert einer Eigenschaft ändern, indem Sie die Datei `mqipt.conf` bearbeiten. Wenn Sie Änderungen übernehmen wollen, aktualisieren Sie MQIPT über den Befehl **mqiptAdmin** mit dem Schlüsselwort **-refresh**.

Um Kommentare in die Konfigurationsdatei aufzunehmen, beginnen Sie eine Zeile mit dem Zeichen "#".

Änderungen an bestimmten Eigenschaften bewirken nur dann den Neustart einer Route, wenn andere Eigenschaften bereits aktiviert sind. Alle Änderungen an den HTTP-Eigenschaften haben z. B. nur einen Effekt, wenn die Eigenschaft **HTTP** auch aktiviert ist.

Wenn eine Route erneut gestartet wird, werden bestehende Verbindungen beendet. Um dieses Verhalten außer Kraft zu setzen, legen Sie für die Eigenschaft **RouteRestart** den Wert `false` fest. Dies verhindert den Neustart der Route, sodass vorhandene Verbindungen aktiv bleiben können, bis die Eigenschaft **RouteRestart** erneut aktiviert wird.

Informationen zum Einrichten einiger einfacher Konfigurationen finden Sie unter [Erste Schritte mit MQIPT](#). Eine Beispielformatierung finden Sie in der Datei `mqiptSample.conf` im MQIPT-Installationsverzeichnis.

Die mqiptAdmin-Eigenschaftendatei

Konfigurationseigenschaften für den Befehl **mqiptAdmin** können in einer separaten Eigenschaftendatei angegeben werden. Diese Konfigurationseigenschaften werden benötigt, wenn **mqiptAdmin** eine Verbindung zum TLS-Befehlssport von MQIPT herstellt.

Die Liste der Eigenschaften, die in der Eigenschaftendatei **mqiptAdmin** angegeben werden können, finden Sie unter „[mqiptAdminEigenschaften](#)“ auf Seite 276. Bei Eigenschaftsnamen muss die Groß-/Kleinschreibung beachtet werden. Alle nicht erkannten Eigenschaften werden ignoriert.

Kommentare können in die Eigenschaftendatei eingeschlossen werden, indem Sie eine Zeile mit einem "#"-Zeichen beginnen.

Zusammenfassung der MQIPT-Eigenschaften

Diese Tabelle zeigt eine Zusammenfassung der MQIPT-Konfigurationseigenschaften und enthält folgende Informationen:

- Eine alphabetische Liste der MQIPT-Eigenschaften mit Links zu weiteren Informationen im Abschnitt `[route]` oder im Abschnitt `[global]`, wenn der Abschnitt `[route]` nicht zutrifft.
- Die Eigenschaft, die für einen Wert auf `true` gesetzt werden muss, damit ein Wert wirksam wird.
- Gibt an, ob die Eigenschaft für den Abschnitt `[global]` und/oder den Abschnitt `[route]` gilt.
- Standardwerte, die verwendet werden, wenn eine Eigenschaft sowohl im Abschnitt `[route]` als auch im Abschnitt `[global]` fehlt. Beim Angeben der Werte `true` und `false` kann eine beliebige Mischung aus Groß- und Kleinbuchstaben verwendet werden.



Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
AccessPW		ja	nein	null
Aktiv		ja	ja	wahr
V 9.4.0 V 9.4.0 „[MQ 9.4.0 Juni 2024][MQ 9.4.0 Juni 2024]AllowedProtocols“ auf Seite 254		ja	ja	mq
ClientAccess		ja	ja	false
CommandPort		ja	nein	null
CommandPortListenerAddress		ja	nein	null
ConnectionLog		ja	nein	wahr
Destination		nein	ja	null
DestinationPort		nein	ja	1414
„EnableAdvancedCapabilities“ auf Seite 252		ja	nein	false
HTTP		ja	ja	false
V 9.4.0 V 9.4.0 „[MQ 9.4.0 Juni 2024][MQ 9.4.0 Juni 2024]HTTPConnectionTimeout“ auf Seite 255		ja	ja	5000
HTTPProxy	HTTP	ja	ja	null
HTTPProxyPort	HTTP	ja	ja	8080
HTTPS	HTTP	ja	ja	false
HTTPServer	HTTP	ja	ja	null
HTTPServerPort	HTTP	ja	ja	null
IdleTimeout		ja	ja	0
IgnoreExpiredCRLs		ja	ja	false
LDAP		ja	ja	false
LDAPIgnoreErrors	LDAP	ja	ja	false
LDAPCacheTimeout	LDAP	ja	ja	24
LDAPServer1	LDAP	ja	ja	null
LDAPServer1Port	LDAP	ja	ja	389
LDAPServer1Userid	LDAP	ja	ja	null
LDAPServer1Password	LDAP	ja	ja	null
LDAPServer1Timeout	LDAP	ja	ja	0
LDAPServer2	LDAP	ja	ja	null
LDAPServer2Port	LDAP	ja	ja	389
LDAPServer2Userid	LDAP	ja	ja	null

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
LDAPServer2Password	LDAP	ja	ja	null
LDAPServer2Timeout	LDAP	ja	ja	0
ListenerAddress		ja	ja	null
ListenerPort		nein	ja	null
LocalAddress		ja	ja	null
LocalAdmin		ja	nein	wahr
MaxConnectionThreads		ja	ja	100
MaxLogFileSize		ja	nein	50
MinConnectionThreads		ja	ja	5
Name		nein	ja	null
OutgoingPort		nein	ja	0
V9.4.0 PasswordProtection		ja	ja	erforderlich
QMgrAccess		ja	ja	wahr
RemoteCommandAuthentication		ja	nein	Ohne
RemoteShutdown		ja	nein	false
RouteRestart		ja	ja	wahr
SecurityExit		ja	ja	false
SecurityExitName	SecurityExit	ja	ja	null
SecurityExitPath	SecurityExit	ja	ja	<i>mqipt_home\exits</i>
SecurityExitTimeout	SecurityExit	ja	ja	30
SecurityManager (Anmerkung 3)		ja	nein	false
RichtlinieSecurityManager (Anmerkung 3)		ja	nein	null
SocksClient		ja	ja	false
SocksProxyHost	SocksClient	ja	ja	null
SocksProxyPort	SocksClient	ja	ja	1080
SocksServer		ja	ja	false
SSLClient		ja	ja	false
SSLClientCAKeyRing	SSLClient	ja	ja	null
SSLClientCAKeyRingPW	SSLClient	ja	ja	null
„ SSLClientCAKeyRingUseCryptoHardware “ auf Seite 262	SSLClient	ja	ja	false
SSLClientCipherSuites	SSLClient	ja	ja	null


Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
SSLClientConnectTimeout	SSLClient	ja	ja	30
SSLClientCustomOutboundSNI	SSLClient	ja	ja	null
SSLClientDN_C	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_CN	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_DC	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_DNQ	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_L	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_O	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_OU	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_PC	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_ST	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_Street	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_T	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientDN_UID	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientExit		ja	ja	false
SSLClientKeyRing	SSLClient	ja	ja	null
SSLClientKeyRingPW	SSLClient	ja	ja	null
„SSLClientKeyRingUseCryptoHardware“ auf Seite 265	SSLClient	ja	ja	false
„SSLClientOutboundSNI“ auf Seite 265	SSLClient	ja	ja	Hostname
SSLClientProtocols	SSLClient	ja	ja	TLSv1.2 TLSv1.3
SSLClientSiteDN_C	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_CN	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_DC	SSLClient	ja	ja	* (Anmerkung 1)

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
SSLClientSiteDN_DNQ	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_L	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_O	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_OU	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_PC	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_ST	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_Street	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_T	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteDN_UID	SSLClient	ja	ja	* (Anmerkung 1)
SSLClientSiteLabel	SSLClient	ja	ja	null
SSLCommandPort		ja	nein	null
SSLCommandPortCipherSuites		ja	nein	null
SSLCommandPortListenerAddress		ja	nein	null
SSLCommandPortKeyRing		ja	nein	null
SSLCommandPortKeyRingPW		ja	nein	null
SSLCommandPortKeyRingUseCryptoHardware		ja	nein	false
SSLCommandPortProtocols		ja	nein	TLSv1.2 TLSv1.3
SSLCommandPortSiteLabel		ja	nein	null
SSLExitData	SSLServerExit	ja	ja	null
SSLExitName	SSLServerExit	ja	ja	null
SSLExitPath	SSLServerExit	ja	ja	<i>mqi</i> pt_home\exits
SSLExitTimeout	SSLServerExit	ja	ja	30
SSLProxyMode		ja	ja	false
SSLPlainConnections	entweder SSLServer oder SSLProxyMode	ja	ja	false
SSLServer		ja	ja	false

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
SSLServerAskClientAuth	SSLServer	ja	ja	false
SSLServerCAKeyRing	SSLServer	ja	ja	null
SSLServerCAKeyRingPW	SSLServer	ja	ja	null
„ SSLServerCAKeyRingUseCryptoHardware “ auf Seite 270	SSLServer	ja	ja	false
SSLServerCipherSuites	SSLServer	ja	ja	null
SSLServerDN_C	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_CN	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_DC	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_DNQ	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_L	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_O	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_OU	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_PC	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_ST	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_Street	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_T	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerDN_UID	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerExit		ja	ja	false
SSLServerKeyRing	SSLServer	ja	ja	null
SSLServerKeyRingPW	SSLServer	ja	ja	null
„ SSLServerKeyRingUseCryptoHardware “ auf Seite 273	SSLServer	ja	ja	false
SSLServerProtocols	SSLServer	ja	ja	TLSv1.2 TLSv1.3
SSLServerSiteDN_C	SSLServer	ja	ja	* (Anmerkung 1)

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
SSLServerSiteDN_CN	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_DC	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_DNQ	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_L	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_O	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_OU	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_PC	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_ST	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_Street	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_T	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteDN_UID	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerSiteLabel	SSLServer	ja	ja	null
StoredCredentialsFormat		ja	ja	null
TCPKeepAlive		ja	ja	false
Trace		ja	ja	0
 TraceFileAnzahl		ja	nein	25
 TraceFileGröße		ja	nein	200
„TraceUser-Daten“ auf Seite 276		ja	ja	64
UriName	HTTP	ja	ja	(Anmerkung 2)

Anmerkungen:

1. Der Stern (*) steht für ein Platzhalterzeichen.
2. Ausführliche Informationen zu den Standardeinstellungen finden Sie unter [UriName](#) im Abschnitt „MQIPT-Routeneigenschaften“ auf Seite 254.
3.  Diese Eigenschaft ist veraltet und wird in einem zukünftigen Release entfernt.

Zugehörige Verweise

„IBM MQ Internet Pass-Thru -Konfigurationsreferenz“ auf Seite 243

IBM MQ Internet Pass-Thru (MQIPT) verwendet eine Konfigurationsdatei namens `mqipt.conf`, um Routen zu definieren und die Aktionen des MQIPT-Servers zu steuern. Ab IBM MQ 9.2 können die Konfigurationseigenschaften für den Befehl **mqiptAdmin** auch in einer Eigenschaftendatei angegeben werden.

„Globale MQIPT-Eigenschaften“ auf Seite 251

Die Konfigurationsdatei `mqipt.conf` kann eine Reihe globaler Eigenschaften enthalten.

„MQIPT-Routeneigenschaften“ auf Seite 254

Die Konfigurationsdatei `mqipt.conf` kann Eigenschaften für einzelne Routen enthalten.

Globale MQIPT-Eigenschaften

Die Konfigurationsdatei `mqipt.conf` kann eine Reihe globaler Eigenschaften enthalten.

Die folgenden Eigenschaften können nur im Abschnitt `[global]` von `mqipt.conf` angezeigt werden. Alle Routeneigenschaften außer **ListenerPort**, **Destination**, **DestinationPort**, **Name** und **OutgoingPort** können auch im Abschnitt `[global]` angezeigt werden. Wenn eine Eigenschaft sowohl im Abschnitt `"route"` als auch im Abschnitt `[global]` angezeigt wird, überschreibt der Wert der Eigenschaft im Abschnitt `[route]` den globalen Wert, aber nur für die betreffende Route. Auf diese Weise kann der Abschnitt `[global]` verwendet werden, um die Standardwerte festzulegen, die für die Eigenschaften verwendet werden sollen, die nicht in den einzelnen `[route]`-Abschnitten festgelegt sind.

AccessPW

Das Kennwort, das zur Authentifizierung von Befehlen verwendet wird, die an den MQIPT-Befehlsport mit dem Befehl **mqiptAdmin** gesendet werden.

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um ein Klartextkennwort. Klartextkennwörter können nur alphanumerische Zeichen enthalten. Es wird dringend empfohlen, die in der MQIPT-Konfiguration gespeicherten Kennwörter zu verschlüsseln. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Die Authentifizierung wird für Verwaltungsbefehle ausgeführt, die vom Befehlsport empfangen werden, wenn beide der folgenden Bedingungen zutreffen:

- Die Eigenschaft **AccessPW** wird angegeben und auf einen Wert gesetzt, der nicht leer ist.
- Die Eigenschaft **RemoteCommandAuthentication** wird angegeben und auf einen anderen Wert als **Kein** gesetzt.

CommandPort

Die TCP/IP-Portnummer des nicht gesicherten Befehlsport. MQIPT akzeptiert Verwaltungsbefehle, die mit dem Befehl **mqiptAdmin** an diesen Befehlsport gesendet werden.

Verbindungen zum nicht gesicherten Befehlsport werden nicht mit TLS gesichert. Daten, die an den Befehlsport gesendet werden, einschließlich des Zugriffskennworts, können von anderen Benutzern des Netzes aufgerufen werden. Wenn Sie einen mit TLS gesicherten Befehlsport konfigurieren möchten, legen Sie stattdessen die Eigenschaft **SSLCommandPort** fest.

Wenn die Eigenschaft **CommandPort** nicht angegeben ist, ist MQIPT nicht für Verwaltungsbefehle am nicht gesicherten Befehlsport empfangsbereit. Wenn Sie die Standardportnummer 1881 verwenden möchten, die standardmäßig vom Befehl **mqiptAdmin** verwendet wird, setzen Sie **CommandPort** auf 1881.

CommandPortListenerAddress

Die lokale Listener-Adresse, die vom nicht gesicherten Befehlsport verwendet werden soll. Wenn Sie die lokale Listener-Adresse festlegen, können Sie eingehende Verbindungen auf den nicht gesicherten Befehlsport auf die von einer bestimmten Netzschnittstelle beschränken. Der Standardwert ist, auf allen Netzschnittstellen empfangsbereit zu sein.

ConnectionLog

Zulässig sind die Werte `true` oder `false`. Bei Angabe von `'true'` protokolliert MQIPT alle (erfolgreichen oder erfolglosen) Verbindungsversuche im Unterverzeichnis `logs` sowie Ereignisse für Verbindungsbeendigungen in der Datei `mqiptYYYYMMDDHHmmSS.log` (die Zeichen `YYYYMMDDHHmmSS` geben dabei das aktuelle Datum und die aktuelle Uhrzeit an). Der Standardwert von **ConnectionLog** ist `Wahr`. Wenn diese Eigenschaft von `true` in `false` geändert wird, schließt MQIPT das vorhandene

Verbindungsprotokoll und erstellt ein neues. Das neue Protokoll wird verwendet, wenn die Eigenschaft auf `true` zurückgesetzt wird.

EnableAdvancedCapabilities

Setzen Sie diese Eigenschaft auf `true`, um zu bestätigen, dass erweiterte Funktionen, für die eine Berechtigung für IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, oder IBM MQ Appliance erforderlich ist, von MQIPT verwendet werden können. Mit der entsprechenden Berechtigung können Sie die erweiterten Funktionen in MQIPT verwenden. Wenn erweiterte Funktionen für eine Route aktiviert sind, muss auch der lokale Warteschlangenmanager, der über die MQIPT -Route verbunden ist, über eine Berechtigung für IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, oder IBM MQ Appliance verfügen. Routen, die erweiterte Funktionen verwenden, können nur gestartet werden, wenn diese Eigenschaft auf `true` gesetzt ist. Wenn diese Eigenschaft von `true` in `false` geändert wird, werden Routen, die erweiterte Funktionen verwenden, gestoppt.

LocalAdmin

Gibt an, ob eine lokale Verwaltung ohne Befehlsport zulässig ist. Verwaltungsbefehle, die vom Befehl **mqiptAdmin** unter Verwendung der lokalen Verwaltung anstelle des Befehlsports gesendet werden, werden nicht akzeptiert, wenn diese Eigenschaft auf `false` gesetzt ist.

Die gültigen Werte für diese Eigenschaft sind `true` und `false`. Der Standardwert ist `true`.

MaxLogFileSize

Die maximale Größe der Verbindungsprotokolldatei (in KB). Wenn die Größe dieser Datei den maximalen Wert überschreitet, wird eine Sicherungskopie (`mqipt001.log`) erstellt und eine neue Datei gestartet. Es werden nur zwei Sicherungsdateien gespeichert (`mqipt001.log` und `mqipt002.log`); frühere Sicherungsdateien werden gelöscht, sobald die Hauptprotokolldatei voll ist. Der Standardwert von **MaxLogFileSize** ist 50; der zulässige Mindestwert ist 5.

RemoteCommandAuthentication

Gibt an, ob Verwaltungsbefehle, die vom nicht gesicherten Befehlsport oder vom TLS-Befehlsport empfangen werden, authentifiziert werden sollen. Befehle werden authentifiziert, indem überprüft wird, ob das angegebene Kennwort dem in der Eigenschaft `AccessPW` angegebenen Kennwort entspricht. Dieser kann einen der folgenden Werte annehmen:

none

Für Befehle, die an einen der beiden Befehlsports ausgegeben werden, wird keine Authentifizierung ausgeführt. Benutzer des Befehls **mqiptAdmin** müssen kein Kennwort eingeben. Dies ist der Standardwert.

optional

Benutzer des Befehls **mqiptAdmin** müssen nicht zwingend ein Kennwort bereitstellen. Wenn ein Kennwort bereitgestellt wird, muss es jedoch gültig sein.

erforderlich

Benutzer des Befehls **mqiptAdmin** müssen ein gültiges Kennwort für jeden Befehl bereitstellen, der an die Befehlsports ausgegeben wird.

Die Eigenschaft **AccessPW** muss ebenfalls angegeben werden, um die Authentifizierung für die Befehlsports zu aktivieren.

RemoteShutDown

Gibt an, ob MQIPT durch einen Stoppbefehl beendet werden kann, der mit dem Befehl **mqiptAdmin** an den nicht gesicherten Befehlsport oder an den TLS-Befehlsport gesendet wird. Diese Eigenschaft muss auf `true` gesetzt sein, damit Stoppbefehle, die von einem der zu verarbeitenden Befehlsports empfangen werden, verarbeitet werden.

Die gültigen Werte für diese Eigenschaft sind `true` und `false`. Der Standardwert ist `false`.

SecurityManager

Setzen Sie diese Eigenschaft auf `true`, um den Java security manager für diese Instanz von MQIPT zu aktivieren. Sie müssen sicherstellen, dass die korrekten Berechtigungen erteilt sind. Weitere Informationen finden Sie in [Java security manager](#). Der Standardwert für diese Eigenschaft ist `false`.

Diese Eigenschaft ist veraltet und wird in einem zukünftigen Release entfernt.

SecurityManagerPolicy

Der vollständig qualifizierte Dateiname einer Java security manager -Richtliniendatei. Wenn diese Eigenschaft nicht festgelegt ist, werden nur die Dateien für das Standardsystem und die Benutzer-richtlinien verwendet. Wenn der Java security manager bereits aktiviert ist, haben Änderungen an dieser Eigenschaft erst Auswirkungen, wenn der Java security manager inaktiviert und erneut aktiviert wurde.

Deprecated Diese Eigenschaft ist veraltet und wird in einem zukünftigen Release entfernt.

SSLCommandPort

Die TCP/IP-Portnummer des TLS-Befehlsports. MQIPT akzeptiert Verwaltungsbefehle, die mit dem Befehl `mqiptAdmin` an diesen Befehlsport gesendet werden. Dieser Port akzeptiert nur TLS-Verbindungen. Diese Eigenschaft muss angegeben werden, damit der TLS-Befehlsport aktiviert werden kann.

SSLCommandPortCipherSuites

Der Name der Cipher-Suites, die auf dem TLS-Befehlsport aktiviert werden sollen. Bei Angabe von mehr als einer Cipher-Suite müssen die Werte durch Kommas getrennt werden. Nur Cipher-Suites des Typs TLS 1.2 und TLS 1.3, die standardmäßig in der mit MQIPT bereitgestellten Java runtime environment (JRE) aktiviert sind, können angegeben werden. Wenn diese Eigenschaft nicht angegeben ist, werden alle Cipher-Suites, die in der JRE aktiviert sind, im TLS-Befehlsport aktiviert.

SSLCommandPortListenerAddress

Die lokale Listener-Adresse, die vom TLS-Befehlsport verwendet werden soll. Wenn Sie die Adresse des lokalen Listener festlegen, können Sie eingehende Verbindungen zum TLS-Befehlsport auf die von einer bestimmten Netzschchnittstelle aus beschränken. Der Standardwert ist, auf allen Netzschchnittstellen empfangsbereit zu sein.

SSLCommandPortKeyRing

Der Name der PKCS#12-Schlüsselringdatei, die das TLS-Befehlsportserverzertifikat enthält.

Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden.

SSLCommandPortKeyRingPW

Das verschlüsselte Kennwort für den Zugriff auf die TLS-Befehlsportschlüsselringdatei oder den PKCS#11-Schlüsselspeicher. Das Kennwort muss mit dem Befehl `mqiptPW` verschlüsselt werden und der Wert dieser Eigenschaft muss auf die von `mqiptPW` ausgegebene Zeichenfolge gesetzt werden.

SSLCommandPortKeyRingUseCryptoHardware

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für das Serverzertifikat des TLS-Befehlsports verwendet wird. Die gültigen Werte für diese Eigenschaft sind `true` und `false`. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLCommandPortKeyRing** nicht ebenfalls angegeben werden.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktion von IBM MQ Advanced. Die Eigenschaft `EnableAdvancedCapabilities` muss auf `true` gesetzt werden, um zu bestätigen, dass Sie über IBM MQ Advanced-Berechtigung verfügen.

SSLCommandPortProtocols

Eine durch Kommas getrennte Liste der Protokolle, die auf dem TLS-Befehlsport aktiviert werden sollen. Einer oder mehrere der folgenden Werte können angegeben werden.

Wert	Protokoll
TLSv1.2	TLS 1.2
TLSv1.3	TLS 1.3

Wenn Sie diese Eigenschaft nicht angeben, sind TLS 1.2 und TLS 1.3 standardmäßig aktiviert.

SSLCommandPortSiteLabel

Der Kennsatzname des Serverzertifikats, das vom TLS-Befehlsport verwendet wird. Wenn diese Eigenschaft nicht angegeben ist, wird jedes Zertifikat im TLS-Befehlsportschlüsselspeicher ausgewählt, das mit der Cipher Suite kompatibel ist.

Trace

Die Tracestufe für globale MQIPT-Threads, die keiner Route zugeordnet sind, und für Routen, für die die Eigenschaft **Trace** nicht festgelegt ist. Der MQIPT-Hauptsteuerungsthread und die Befehlsserverthreads sind beispielsweise keiner Route zugeordnet und werden nur verfolgt, wenn der Trace im Abschnitt [global] aktiviert ist. Der Wert der Eigenschaft **Trace** in einem Abschnitt [route] überschreibt die globale Eigenschaft **Trace** für diese Route. Informationen zum Tracing von Threads, die einer Route zugeordnet sind, finden Sie unter **Trace** im Abschnitt [route].

Diese Eigenschaft kann einen der folgenden Werte haben:

0

Trace ist nicht aktiviert

Jede beliebige positive Ganzzahl.

Trace ist aktiviert

Der Standardwert ist 0.

V 9.4.0 TraceFileAnzahl

Die Anzahl der Tracedateien in der rotierenden Dateigruppe, die von MQIPT zum Schreiben von Tracedaten verwendet werden.

Der zulässige Mindestwert ist 3. Der Standardwert ist 25.

Wenn Sie den Wert dieser Eigenschaft ändern, wird die aktuelle Tracedatei geschlossen und die nächste Datei in der rotierenden Gruppe von Tracedateien geöffnet.

V 9.4.0 Größe der TraceFile

Die maximale Größe der von MQIPT erstellten Tracedateien (in MB).

Der zulässige Mindestwert ist 1. Der Standardwert ist 200.

Wenn Sie den Wert dieser Eigenschaft ändern, wird die aktuelle Tracedatei geschlossen und die nächste Datei in der rotierenden Gruppe von Tracedateien geöffnet.

MQIPT-Routeneigenschaften

Die Konfigurationsdatei mqipt.conf kann Eigenschaften für einzelne Routen enthalten.

Der Abschnitt [route] der Konfigurationsdatei mqipt.conf kann die folgenden Eigenschaften enthalten:

Aktiv

Die Route akzeptiert eingehende Verbindungen nur, wenn der Wert von **Active** auf **Wahr** gesetzt ist. Dies bedeutet, dass Sie den Zugriff auf das Ziel vorübergehend inaktivieren können, indem Sie diesen Wert auf **Falsch** setzen, ohne den Abschnitt [route] aus der Konfigurationsdatei löschen zu müssen. Wenn Sie den Wert dieser Eigenschaft in **false** ändern, wird die Route gestoppt, wenn ein Befehl zur Aktualisierung ausgegeben wird. Alle Verbindungen zu der Route werden gestoppt.

V 9.4.0 **V 9.4.0** AllowedProtocols

Gibt die Protokolle an, die von dieser Route akzeptiert werden. Verbindungen, die ein nicht in dieser Liste angegebenes Protokoll verwenden, werden zurückgewiesen. Es können mehrere Protokolle als durch Kommas getrennte Liste angegeben werden. Folgende Protokolle können angegeben werden:

MQ

Die Route akzeptiert Verbindungen, die das Protokoll IBM MQ verwenden.

http

Die Route akzeptiert HTTP-Verbindungen von einer anderen Instanz von MQIPT.

Der Standardwert dieser Eigenschaft ist `mq`.

Wenn der Wert dieser Eigenschaft geändert wird, wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

ClientAccess

Die Route lässt eingehende Clientkanalverbindungen nur zu, wenn der Wert von **ClientAccess** auf `true` gesetzt ist. Beachten Sie, dass Sie MQIPT auch so konfigurieren können, dass nur Clientanforderungen, nur Warteschlangenmanageranforderungen oder beide Anforderungstypen akzeptiert werden. Verwenden Sie diese Eigenschaft zusammen mit der Eigenschaft **QMGrAccess**. Wenn Sie diese Eigenschaft in `false` ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

Destination

Der Hostname (oder die IP-Adresse in Schreibweise mit Trennzeichen) des Warteschlangenmanagers, oder der nachfolgenden Instanz von MQIPT, zu der diese Route eine Verbindung herstellen soll. Jeder Abschnitt `[route]` muss einen expliziten **Destination**-Wert enthalten, aber mehrere Abschnitte `[route]` können auf dasselbe Ziel verweisen. Wenn sich eine Änderung dieser Eigenschaft auf die Route auswirkt, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt. Bei Verwendung der Eigenschaft **SocksProxyHost** muss die Eigenschaft **Destination** das IPv4-Adressformat in Schreibweise mit Trennzeichen verwenden.

DestinationPort

Der Port auf dem Zielhost, zu dem diese Route eine Verbindung herstellen soll. Jeder `[route]`-Abschnitt muss einen expliziten **DestinationPort**-Wert enthalten, aber mehrere Routen können auf dieselbe Kombination von **Destination**- und **DestinationPort**-Werten verweisen. Wenn sich eine Änderung dieser Eigenschaft auf die Route auswirkt, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

HTTP

Setzen Sie **HTTP** für Routen, die für abgehende HTTP-Tunnelungsanforderungen zuständig sind, auf `true`. Die Eigenschaft **Destination** für die Route muss der Hostname einer anderen Instanz von MQIPT sein, wenn **HTTP** auf `true` gesetzt ist. Setzen Sie **HTTP** auf `false` für Routen, die mit IBM MQ-Warteschlangenmanagern verbunden sind. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt. Mindestens eine der Eigenschaften **HTTPProxy** oder **HTTPServer** muss ebenfalls angegeben werden, wenn **HTTP** auf `true` gesetzt ist. Diese Eigenschaft kann nicht gemeinsam mit der Eigenschaft **SocksClient** verwendet werden.

V 9.4.0

V 9.4.0

HTTPConnectionTimeout

Diese Eigenschaft gibt die Zeit in Millisekunden an, die MQIPT auf die erfolgreiche Herstellung einer HTTP-Verbindung wartet, bevor die Verbindung zurückgewiesen wird.

Der Standardwert ist 5000.

HTTPProxy

Der Hostname (oder die IP-Adresse in Schreibweise mit Trennzeichen) des HTTP-Proxys, der von allen Verbindungen für diese Route verwendet wird. Eine **CONNECT**-Anforderung wird an den HTTP-Proxy ausgegeben, anstelle der **POST**-Anforderung, die normalerweise verwendet wird, wenn kein HTTP-Proxy konfiguriert ist. Wenn Sie diese Eigenschaft ändern (und **HTTP** auf `true` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

HTTPProxyPort

Die Portadresse, die im HTTP-Proxy verwendet werden soll. Der Standardwert ist 8080. Wenn Sie diese Eigenschaft ändern (und **HTTP** auf `true` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

HTTPServer

Der Hostname (oder die IP-Adresse in Schreibweise mit Trennzeichen) des HTTP-Servers, der von allen Verbindungen für diese Route verwendet wird. Dies ist in der Regel der Hostname eines anderen MQIPT.

Wenn **HTTPProxy** nicht angegeben ist, stellt MQIPT eine Verbindung zu dem in **HTTPServer** angegebenen Host her und gibt HTTP-Anforderungen **POST** an den Host aus, der in der Eigenschaft **Destination** der Route angegeben ist. Wenn **HTTPProxy** angegeben ist, stellt MQIPT stattdessen eine Verbindung zu dem in **HTTPProxy** angegebenen Host her und fordert an, dass der Proxy einen Tunnel zu dem in **HTTPServer** angegebenen Host herstellt.

Wenn **HTTPProxy** angegeben wird, ist der Standardwert die Route **Destination**.

Wenn Sie diese Eigenschaft ändern (und **HTTP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

HTTPS

Setzen Sie **HTTPS** auf `Wahr`, wenn HTTPS-Anforderungen abgesetzt werden sollen. Die Eigenschaften **HTTP** und **SSLClient** müssen ebenfalls aktiviert sein und der Clientschlüsselring muss wie bei einer SSL/TLS-Operation mit der Eigenschaft **SSLClientKeyRing** oder **SSLClientKeyRingUseCryptoHardware** konfiguriert werden. Wenn Sie die Eigenschaft **HTTPS** ändern (und **HTTP** auf `true` gesetzt ist), wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

HTTPServerPort

Die Portadresse, die im HTTP-Server verwendet werden soll. Der Standardwert ist 8080, sofern **HTTPProxy** nicht angegeben ist. In diesem Fall ist der Standardwert die Route **DestinationPort**.

Wenn Sie diese Eigenschaft ändern (und **HTTP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

IdleTimeout

Die Zeit in Minuten, nach der eine inaktive Verbindung geschlossen wird. Beachten Sie, dass Warteschlangenmanager-zu-Warteschlangenmanager-Kanäle auch über die Eigenschaft **DISCINT** verfügen. Wenn Sie den Parameter **IdleTimeout** festlegen, beachten Sie **DISCINT**. Wenn **IdleTimeout** auf 0 gesetzt ist, gibt es kein Inaktivitätszeitlimit. Änderungen an dieser Eigenschaft werden nur bei einem Neustart der Route wirksam.

IgnoreExpiredCRLs

Setzen Sie **IgnoreExpiredCRLs** auf `Wahr`, um eine abgelaufene CRL zu ignorieren. Der Standardwert ist `false`. Wenn Sie **IgnoreExpiredCRLs** auf `Wahr` setzen, könnte ein widerrufenes Zertifikat verwendet werden, um eine SSL/TLS-Verbindung herzustellen.

LDAP

Setzen Sie **LDAP** auf `Wahr`, um die Verwendung eines LDAP-Servers bei Verwendung von SSL/TLS-Verbindungen zu aktivieren. MQIPT nutzt den LDAP-Server, um CRLs und ARLs abzurufen. Die Eigenschaft **SSLClient** oder **SSLServer** muss ebenfalls auf `Wahr` gesetzt sein, damit diese Eigenschaft wirksam wird.

LDAPCacheTimeout

Die Ablaufzeit des temporären Cache in Stunden, die eine von einem LDAP-Server abgerufene CRL gespeichert wird. Danach wird der gesamte CRL-Cache geleert. Wenn Sie beispielsweise einen Wert von 1 Stunde angeben, wird der Cache einmal pro Stunde geleert. Der Standardwert ist 24. Wenn Sie einen Zeitlimitwert von 0 angeben, verfallen die Einträge im Cache erst, wenn die Route erneut gestartet wird. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPIgnoreErrors

Setzen Sie **LDAPIgnoreErrors** auf `Wahr`, wenn beim Ausführen einer LDAP-Suche alle Verbindungs- oder Zeitlimitfehler ignoriert werden sollen. Wenn MQIPT keine erfolgreiche Suche durchführen kann, kann die Clientverbindung nur dann vollständig abgeschlossen werden, wenn diese Eigenschaft aktiviert wurde. Bei einer erfolgreichen Suche wurde eine CRL abgerufen oder es sind keine CRLs für die

angegebene CA verfügbar. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Anmerkung: Wenn Sie diese Eigenschaft aktivieren, kann eine SSL/TLS-Verbindung mithilfe eines widerrufenen Zertifikats hergestellt werden.

LDAPServer1

Der Hostname oder die IP-Adresse des LDAP-Hauptservers. Diese Eigenschaft muss festgelegt sein, wenn LDAP auf `true` gesetzt ist. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer1Port

Die Nummer des Empfangsports auf dem LDAP-Hauptserver. Der Standardwert ist 389. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer1Userid

Die Benutzer-ID, die für den Zugriff auf den LDAP-Hauptserver erforderlich ist. Diese Eigenschaft muss festgelegt sein, wenn eine Autorisierung für den Zugriff auf den LDAP-Hauptserver erforderlich ist. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer1Password

Das Kennwort, das für den Zugriff auf den LDAP-Hauptserver erforderlich ist. Diese Eigenschaft muss festgelegt werden, wenn **LDAPServer1Userid** auf `Wahr` gesetzt wurde. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um ein Klartextkennwort. Klartextkennwörtern können nur alphanumerische Zeichen enthalten. Es wird dringend empfohlen, die in der MQIPT-Konfiguration gespeicherten Kennwörter zu verschlüsseln. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

LDAPServer1Timeout

Die Zeit in Sekunden, die MQIPT auf eine Antwort vom LDAP-Hauptserver wartet. Der Standardwert ist 0, was bedeutet, dass das Zeitlimit der Verbindung nie überschritten wird. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer2

Der Hostname oder die IP-Adresse des LDAP-Sicherungsservers. Diese Eigenschaft ist optional. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer2Port

Die Nummer des Empfangsports für den LDAP-Sicherungsserver. Der Standardwert ist 389. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer2Userid

Die Benutzer-ID, die für den Zugriff auf den LDAP-Sicherungsserver erforderlich ist. Diese Eigenschaft muss festgelegt sein, wenn eine Autorisierung für den Zugriff auf den LDAP-Sicherungsserver erforderlich ist. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

LDAPServer2Password

Das Kennwort, das für den Zugriff auf den LDAP-Sicherungsserver erforderlich ist. Diese Eigenschaft muss festgelegt werden, wenn **LDAPServer2** auf Wahrgesetzt wurde. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um ein Klartextkennwort. Klartextkennwörtern können nur alphanumerische Zeichen enthalten. Es wird dringend empfohlen, die in der MQIPT-Konfiguration gespeicherten Kennwörter zu verschlüsseln. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

LDAPServer2Timeout

Die Zeit in Sekunden, die MQIPT auf eine Antwort vom LDAP-Sicherungsserver wartet. Der Standardwert ist 0, was bedeutet, dass das Zeitlimit der Verbindung nie überschritten wird. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

ListenerAddress

Verwenden Sie diese Eigenschaft, wenn das MQIPT-System über mehrere IP-Adressen verfügt und Sie den Listener-Port für die Route an eine bestimmte Adresse binden müssen. Dies ist hilfreich, wenn Sie eingehende Verbindungen auf die aus einer bestimmten Netzchnittstelle beschränken möchten. Der Wert dieser Eigenschaft sollte eine IP-Adresse für eine der Netzchnittstellen auf dem System sein, auf dem MQIPT ausgeführt wird. Standardmäßig werden Verbindungen aus allen Netzchnittstellen akzeptiert.

ListenerPort

Die Portnummer, an der die Route für eingehende Anforderungen empfangsbereit sein sollte. Jeder [route]-Abschnitt muss einen expliziten **ListenerPort**-Wert enthalten. Die in jedem Abschnitt festgelegten **ListenerPort**-Werte müssen eindeutig sein. Es kann jede gültige Portnummer verwendet werden, einschließlich der Ports 80 und 443, sofern die ausgewählten Ports nicht von einem anderen TCP/IP-Listener verwendet werden, der auf dem gleichen Host ausgeführt wird.

LocalAddress

Die IP-Adresse, mit der alle Verbindungen zu dieser Route auf diesem Computer gebunden werden sollen. Die ausgewählte Adresse muss eine IP-Adresse sein, die einer der Netzchnittstellen auf dem Computer zugeordnet ist, auf dem MQIPT ausgeführt wird. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

MaxConnectionThreads

Die maximale Anzahl der Verbindungsthreads und somit die maximale Anzahl gleichzeitiger Verbindungen, die von dieser Route verarbeitet werden können. Wenn dieser Grenzwert erreicht ist, gibt der Wert **MaxConnectionThreads** auch die Anzahl der Verbindungen an, die sich in der Warteschlange befinden, wenn alle Threads verwendet werden. Wenn diese Zahl erreicht wird, werden nachfolgende Verbindungsanforderungen abgelehnt.

Der zulässige Mindestwert ist der höhere Wert von 1 und dem Wert von **MinConnectionThreads**.

Wenn der Wert erhöht wird, wird der neue Wert verwendet, wenn der Aktualisierungsbefehl ausgegeben wird. Alle Verbindungen verwenden sofort den neuen Wert. Die Route wird nicht gestoppt.

Wenn der Wert verringert wird, wird der neue Wert erst nach einem Neustart der Route wirksam.

MinConnectionThreads

Die Anzahl der Verbindungsthreads, die für die Verarbeitung eingehender Verbindungen auf einer Route zugeordnet sind, wenn die Route gestartet wird. Die Anzahl der zugeordneten Threads sinkt nicht unter diesen Wert ab, solange die Route aktiv ist.

Der Wert muss zwischen 0 und dem Wert von **MaxConnectionThreads** liegen.

Änderungen an dieser Eigenschaft werden nur bei einem Neustart der Route wirksam.

Name

Ein Name, mit dem die Route angegeben wird. Diese Eigenschaft ist optional. Der Wert wird in Konsolennachrichten und Traceinformationen angezeigt. Änderungen an dieser Eigenschaft werden nur bei einem Neustart der Route wirksam.

OutgoingPort

Die Nummer des Startports, der von abgehenden Verbindungen verwendet wird. Der Bereich der Portnummern stimmt mit dem Wert **MaxConnectionThread** für diese Route überein. Bei Angabe des Standardwerts 0 wird eine vom System definierte Portnummer verwendet. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu dieser Route werden gestoppt. Wenn HTTP verwendet wird, sind für jede Kanalverbindung zwei abgehende Ports erforderlich. Weitere Informationen finden Sie im Abschnitt [Portnummernsteuerung](#).

V 9.4.0

PasswordProtection

Gibt an, ob MQIPT für MQIPT -Routen, die für das Hinzufügen oder Entfernen der TLS-Verschlüsselung konfiguriert sind, den Schutz für Berechtigungsnachweise hinzufügen oder entfernen kann, die in MQCSP-Strukturen von IBM MQ clientsgesendet werden, um die Kompatibilität zwischen dem Client und dem Warteschlangenmanager aufrechtzuerhalten.

Berechtigungsnachweise in MQCSP-Strukturen können entweder geschützt werden, indem die IBM MQ MQCSP-Kennwortschutzfunktion verwendet wird, oder mithilfe der TLS-Verschlüsselung verschlüsselt werden. Der MQCSP-Kennwortschutz ist für Test- und Entwicklungszwecke nützlich, da er einfacher ist als die Einrichtung der TLS-Verschlüsselung, aber nicht so sicher.

Weitere Informationen zum MQCSP-Kennwortschutz finden Sie unter [MQCSP-Kennwortschutz](#).

Wenn eine MQIPT -Route zum Hinzufügen oder Entfernen der TLS-Verschlüsselung konfiguriert ist, muss MQIPT möglicherweise die Berechtigungsnachweise in der MQCSP-Struktur schützen oder den MQCSP-Kennwortschutz entfernen, damit die Verbindung erfolgreich hergestellt werden kann.

Die Eigenschaft kann einen der folgenden Werte aufweisen:

erforderlich

MQIPT stellt sicher, dass Berechtigungsnachweise in der MQCSP-Struktur entweder mit TLS verschlüsselt oder mit MQCSP-Kennwortschutz geschützt werden.

Wenn Berechtigungsnachweise in der MQCSP-Struktur vom Client mithilfe der TLS-Verschlüsselung verschlüsselt gesendet werden und die MQIPT -Route die TLS-Verschlüsselung entfernt, schützt MQIPT die Berechtigungsnachweise mit MQCSP-Kennwortschutz, bevor die Berechtigungsnachweise an das Routenziel weitergeleitet werden. Dies ist der Fall, wenn die MQIPT -Route mit `SSLServer=true` und `SSLClient=false` konfiguriert ist und die ausgewählte CipherSuite keine Nullverschlüsselung verwendet.

Wenn Berechtigungsnachweise in der MQCSP-Struktur vom Client mit MQCSP-Kennwortschutz geschützt werden, entfernt MQIPT den Schutz nicht, auch wenn die Verbindung zwischen MQIPT und dem Routenziel TLS-Verschlüsselung verwendet. Wenn die Verbindung zwischen MQIPT und dem Routenziel die TLS-Verschlüsselung verwendet, schlägt die Verbindung möglicherweise mit dem Ursachencode `MQRC_PASSWORD_PROTECTION_ERROR` (2594) fehl.

Dies ist der Standardwert.

kompatibel

MQIPT wendet den MQCSP-Kennwortschutz an oder entfernt ihn, um sicherzustellen, dass die Verbindung erfolgreich hergestellt werden kann.

Wenn die Berechtigungsnachweise in der MQCSP-Struktur vom Client mit TLS-Verschlüsselung verschlüsselt gesendet werden und die MQIPT -Route die TLS-Verschlüsselung entfernt, schützt MQIPT die Berechtigungsnachweise mit MQCSP-Kennwortschutz, bevor das Kennwort an das Routenziel weitergeleitet wird. Dies ist der Fall, wenn die MQIPT -Route mit `SSLServer=true` und `SSLClient=false` konfiguriert ist und die ausgewählte CipherSuite keine Nullverschlüsselung verwendet.

Wenn Berechtigungsnachweise in der MQCSP-Struktur durch den Client mit MQCSP-Kennwortschutz geschützt werden und die MQIPT -Route TLS-Verschlüsselung hinzufügt, entfernt MQIPT

den MQCSP-Kennwortschutz, bevor die Berechtigungsnachweise an das Routenziel weitergeleitet werden. Dies ist der Fall, wenn die MQIPT -Route mit `SSLServer=false` und `SSLClient=true` konfiguriert ist und die ausgewählte CipherSuite keine Nullverschlüsselung verwendet.

Diese Option bietet die beste Kompatibilität. Sie sollte jedoch nur für Test- und Entwicklungszwecke in vertrauenswürdigen Netzen verwendet werden, da sie nicht sicherstellt, dass das Kennwort im Netz geschützt ist.

passthru

Berechtigungsnachweise in der MQCSP-Struktur werden von MQIPT an das Routenziel weitergeleitet, ohne den MQCSP-Kennwortschutz hinzuzufügen oder zu entfernen. Wenn die MQIPT -Route zum Hinzufügen oder Entfernen der TLS-Verschlüsselung konfiguriert ist, können Clientverbindungen mit dem Ursachencode `MQRC_PASSWORD_PROTECTION_ERROR` (2594) fehlschlagen.

QMGrAccess

Setzen Sie **QMGrAccess** auf `Wahr`, um eingehende Kanalverbindungen des Warteschlangenmanagers (z. B. Senderkanäle) zuzulassen. Wenn Sie den Wert dieser Eigenschaft in `false` ändern, wird die Route gestoppt, wenn ein Befehl zur Aktualisierung ausgegeben wird. Alle Verbindungen zu dieser Route werden gestoppt.

RouteRestart

Setzen Sie **RouteRestart** auf `Falsch`, damit die Route nicht erneut gestartet wird, wenn andere Routeneigenschaften geändert wurden und ein Aktualisierungsbefehl abgesetzt wurde. Der Standardwert für diese Eigenschaft ist `Wahr`.

SecurityExit

Setzen Sie **SecurityExit** auf `Wahr`, um einen benutzerdefinierten Sicherheitsexit zu aktivieren. Der Standardwert für diese Eigenschaft ist `false`.

SecurityExitName

Der Klassenname des benutzerdefinierten Sicherheitsexits. Diese Eigenschaft muss festgelegt werden, wenn **SecurityExit** auf `Wahr` gesetzt wurde. Wenn Sie diese Eigenschaft ändern (und **SecurityExit** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SecurityExitPath

Der vollständig qualifizierte Pfad mit dem benutzerdefinierten Sicherheitsexit. Wenn diese Eigenschaft nicht festgelegt wurde, wird standardmäßig das Unterverzeichnis des Exits verwendet. Diese Eigenschaft kann auch den Namen der Datei mit einem Java-Archiv (JAR-Datei) definieren, in dem sich der benutzerdefinierte Sicherheitsexit befindet. Wenn Sie diese Eigenschaft ändern (und **SecurityExit** auf `Wahr` gesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SecurityExitTimeout

Der Zeitlimitwert (in Sekunden), der von MQIPT verwendet wird, um festzulegen, wie lange auf eine Antwort gewartet wird, wenn eine Verbindungsanforderung validiert wird. Der Standardwert ist 30. Wenn Sie diese Eigenschaft ändern (und **SecurityExit** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SocksClient

Setzen Sie **SocksClient** auf `Wahr`, damit die Route als SOCKS-Client agiert und alle Verbindungen über den SOCKS-Proxy mit den Eigenschaften **SocksProxyHost** und **SocksProxyPort** definiert werden. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt. Diese Eigenschaft kann mit den folgenden Eigenschaften verwendet werden:

- **HTTP**
- **SocksServer**
- **SSLClient**
- **SSLProxyMode**

SocksProxyHost

Der Hostname (oder die IPv4-Adresse in der Schreibweise mit Trennzeichen) des SOCKS-Proxys, den alle Verbindungen für diese Route verwenden. Wenn Sie diese Eigenschaft ändern (und **SocksClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt. Bei Verwendung der Eigenschaft **SocksProxyHost** muss die Eigenschaft **Destination** das Format in der Schreibweise mit Trennzeichen verwenden.

SocksProxyPort

Die Portnummer, die in einem SOCKS-Proxy verwendet werden soll. Der Standardwert ist 1080. Wenn Sie diese Eigenschaft ändern (und **SocksClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SocksServer

Setzen Sie **SocksServer** auf `Wahr`, damit die Route als SOCKS-Proxy agiert und SOCKS-Clientverbindungen akzeptiert. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt. Diese Eigenschaft kann mit den folgenden Eigenschaften verwendet werden:

- **SocksClient**
- **SSLProxyMode**
- **SSLServer**

SSLClient

Setzen Sie **SSLClient** auf `Wahr`, damit die Route als SSL/TLS-Client agiert und abgehende SSL/TLS-Verbindungen hergestellt werden. Die Einstellung **SSLClient** auf `Wahr` impliziert, dass das Ziel entweder eine andere Instanz von MQIPT ist, die als SSL/TLS-Server fungiert, oder ein HTTP-Proxy/-Server.

Wenn Sie **SSLClient** auf `true` setzen, müssen Sie mit der Eigenschaft **SSLClientKeyRing** oder **SSLClientCAKeyRing** einen SSL/TLS-Clientschlüsselring angeben oder MQIPT für die Verwendung von Verschlüsselungshardware konfigurieren, indem Sie die Eigenschaft **SSLClientKeyRingUseCryptoHardware** bzw. **SSLClientCAKeyRingUseCryptoHardware** festlegen.

Wenn Sie **SSLClient** ändern, wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

Diese Eigenschaft kann nicht in Verbindung mit der folgenden Eigenschaft verwendet werden:

- **SSLProxyMode**

SSLClientCAKeyRing

Der vollständig qualifizierte Dateiname der Schlüsselringdatei mit CA-Zertifikaten, die für die Authentifizierung von Zertifikaten vom SSL/TLS-Server verwendet werden. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientCAKeyRingPW

Das Kennwort zum Öffnen der CA-Schlüsselringdatei des SSL/TLS-Clients, die mit der Eigenschaft **SSLClientCAKeyRing** angegeben wird, oder zum Herstellen einer Verbindung zum Schlüsselpeicher der Verschlüsselungshardware, wenn die Eigenschaft **SSLClientCAKeyRingUseCryptoHardware** auf `Wahr` gesetzt ist.

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientCAKeyRingUseCryptoHardware

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für CA-Zertifikate verwendet wird, die zur Authentifizierung von Serverzertifikaten vom SSL/TLS-Server verwendet werden, wenn MQIPT als SSL/TLS-Client agiert. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLClientCAKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Um diese Funktion nutzen zu können, muss der lokale Warteschlangenmanager, der über die MQIPT -Route verbunden ist, auch über die Berechtigung IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `true` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** wird gesetzt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

SSLClientCipherSuites

Der Name der SSL/TLS-CipherSuite, die auf der SSL/TLS-Clientseite verwendet werden soll. Dabei kann es sich um eine oder mehrere der unterstützten CipherSuites handeln. Wenn Sie diese Eigenschaft leer lassen, wird jede Cipher-Suite für die aktivierten Protokolle verwendet, die mit dem Clientzertifikat im Schlüsselring kompatibel ist. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLClientConnectTimeout

Die Zeit in Sekunden, die ein SSL/TLS-Client wartet, bis eine SSL/TLS-Verbindung akzeptiert wird. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientCustomOutboundSNI

Gibt den Wert von Angabe des Servernamens (SNI) an, wenn MQIPT eine TLS-Verbindung zum Routenziel einleitet, wenn die Route mit der Einstellung Benutzerdefiniert für **SSLClientOutboundSNI** konfiguriert ist. Legen Sie mit dieser Eigenschaft für die SNI einen bestimmten Wert fest, der nicht automatisch von MQIPT festgelegt werden kann. Dies kann beispielsweise der Fall sein, wenn Sie für die SNI einen Hostnamen festlegen wollen, das Routenziel jedoch mit einer IP-Adresse konfiguriert ist.

Der Wert muss ein gültiger IDN (Internationalized Domain Name, internationaler Domänenname) sein, der mit der Spezifikation RFC 3490 konform ist. Der Wert darf nicht mit einem abschließenden Punkt enden. Falls ein ungültiger Wert angegeben ist, wird die Route nicht gestartet.

Wenn Sie den Wert dieser Eigenschaft ändern und **SSLClientOutboundSNI** auf Benutzerdefiniert gesetzt ist, wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird.



Achtung: Sie dürfen diese Einstellung nicht verwenden, wenn Verbindungen an einen IBM MQ -Kanal weitergeleitet werden, für den im Feld **CERTLABL** des Kanals eine Zertifikatsbezeichnung konfiguriert ist. Wenn Sie einen Client so weiterleiten, wird er mit dem Rückkehrcode `MQRC_SSL_INITIALIZATION_ERROR` zurückgewiesen und in den Fehlerprotokollen des fernen Warteschlangenmanagers wird ein Fehler `AMQ9673` ausgegeben.

SSLClientDN_C

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Ländernamen übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmung

gen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Ländernamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_CN

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem allgemeinen Namen übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_DC

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Domänenkomponente übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domänennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Domänenkomponenten akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_DNQ

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Qualifikationsmerkmal für eine Domäne übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Qualifikationsmerkmale für Domänen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_L

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Position übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Positionen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_O

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Unternehmen übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate von allen Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_OU

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Organisationseinheit (Organizational Unit, OU) übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese

Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLClientDN_PC

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Postleitzahl übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Postleitzahlen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_ST

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Status übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate in jedem Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_Street

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Straßennamen übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_T

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Titel übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientDN_UID

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Benutzer-ID übereinstimmen. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Benutzer-IDs akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientExit

Mit dieser Eigenschaft können Sie die Verwendung eines Exits aktivieren oder inaktivieren, wenn die Route als SSL/TLS-Client fungiert. Auf diese Weise können Sie die Exitdetails in der Konfigurationsdatei definieren, ohne dass sie tatsächlich verwendet werden.

SSLClientKeyRing

Der vollständig qualifizierte Dateiname der Schlüsselringdatei, die das Clientzertifikat enthält. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie **SSLClientKeyRing** ändern (und **SSLClient** auf Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientKeyRingPW

Das Kennwort zum Öffnen der mit der Eigenschaft **SSLClientKeyRing** angegebenen SSL/TLS-Clientschlüsselringdatei oder zum Herstellen einer Verbindung zum Keystore der Verschlüsselungshardware, wenn die Eigenschaft **SSLClientKeyRingUseCryptoHardware** auf `Wahr` gesetzt ist.

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Wenn Sie **SSLClientKeyRingPW** ändern (und **SSLClient** auf `Wahr` gesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientKeyRingUseCryptoHardware

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher mit dem Clientzertifikat verwendet wird, wenn MQIPT als SSL/TLS-Client agiert. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLClientKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Um diese Funktion nutzen zu können, muss der lokale Warteschlangenmanager, der über die MQIPT -Route verbunden ist, auch über die Berechtigung IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `true` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** wird gesetzt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

SSLClientOutboundSNI

Gibt den Wert der Erweiterung für die Servernamensangabe (Server Name Indication, SNI) an, wenn MQIPT eine TLS-Verbindung zum Routenziel einleitet. Die SNI wird abhängig von der Konfiguration entweder von IBM MQ-Warteschlangenmanagern verwendet, um beim TLS-Handshake das richtige Zertifikat anzugeben, oder zur Weiterleitung von Verbindungen an das Ziel.

Diese Eigenschaft gilt nur für Routen, die mit `SSLClient=true` definiert sind, und kann nicht für Routen angegeben werden, die mit `HTTP=true` definiert sind. Wenn Sie den Wert dieser Eigenschaft ändern und **SSLClient** auf `Wahr` gesetzt ist, wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird.



Achtung: Wenn der Zielkanal mit einer Zertifikatsbezeichnung im Feld **CERTLABL** des Kanalobjekts konfiguriert ist, muss die Einstellung **CERTLABL** auf den Kanalwert gesetzt werden. Wenn ein Client ohne die SNI-Kanaleinstellung weitergeleitet wird, wird er mit dem Rückkehrcode `MQRC_SSL_INITIALIZATION_ERROR` und einer `AMQ9673` -Nachricht zurückgewiesen, die in den Fehlerprotokollen des fernen Warteschlangenmanagers ausgegeben wird.

Die Eigenschaft kann einen der folgenden Werte aufweisen:

Hostname

Als SNI wird der Hostnamen des Routenziels festgelegt. Verwenden Sie diese Option, wenn die Route eine Verbindung zu einer Lastausgleichsfunktion oder zu einem Router herstellt, die/der die SNI zur Weiterleitung von Anforderungen nutzt. Der Red Hat® OpenShift® Container Platform Router verwendet beispielsweise die SNI, um Anforderungen an den IBM MQ-Warteschlangenmanager weiterzuleiten.

Falls ein Warteschlangenmanager das Routenziel ist, empfangen Verbindungsanforderungen während des TLS-Handshakes das Standardzertifikat des fernen Warteschlangenmanagers. Kanalbezogene Zertifikate können somit nicht verwendet werden.

Falls das Routenziel unter Verwendung einer IP-Adresse angegeben ist und wird keine DNS-Rückwärtssuche ausgeführt werden kann, ist die SNI leer.

Dies ist der Standardwert.

Kanal

Als SNI wird der Name des IBM MQ-Kanals festgelegt. Mit dieser Option können Sie die Verwendung von kanalbezogenen Zertifikaten durch den Zielwarteschlangenmanager zulassen, falls von der Route empfangene Verbindungen aus einem der folgenden Gründe in der SNI nicht den Kanalnamen enthalten:

- Die Route ist so konfiguriert, dass Verbindungen akzeptiert werden, die nicht mit TLS mit `SSLServer=false` oder `SSLPlainConnections=true` gesichert sind.
- Die Anwendung, die eine Verbindung zur Route herstellt, kann die SNI nicht festlegen oder ist so konfiguriert, dass sie für die SNI einen anderen Wert als den IBM MQ-Kanalnamen festlegt.

passthru

Wenn die Route mit `SSLServer=true` definiert ist, wird die SNI in der abgehenden Verbindung auf den Wert der SNI gesetzt, die in der eingehenden Verbindung zur Route empfangen wurde. Akzeptiert die Route konfigurationsgemäß keine TLS-Verbindungen, wird als SNI der Zielhostname festgelegt.

custom

Die SNI wird auf den Wert gesetzt, der in der Eigenschaft `SSLClientCustomOutboundSNI` angegeben ist. Wird die Eigenschaft `SSLClientCustomOutboundSNI` nicht angegeben, wird die SNI so gesetzt, als wäre die Route mit `SSLClientOutboundSNI=hostname` konfiguriert.

none

Die SNI ist nicht festgelegt.

SSLClientProtocols

Wird verwendet, um die Gruppe aktivierter Secure Socket-Protokolle zu beschränken, die verwendet werden, um abgehende Verbindungen zum Ziel für eine Route herzustellen, wenn `SSLClient` auf `Wahr` gesetzt ist.

Sie können mehrere Werte angeben, indem Sie sie mit Kommas trennen. Wenn Sie diese Eigenschaft nicht angeben, sind TLS 1.2 und TLS 1.3 standardmäßig aktiviert. Um andere Protokolle als TLS 1.2 oder TLS 1.3 zu aktivieren, müssen Sie die zu aktivierenden Protokolle in dieser Eigenschaft angeben und außerdem Unterstützung für das Protokoll in der Java runtime environment hinzufügen, indem Sie die im Abschnitt [Veraltete Protokolle und CipherSuites](#) beschriebene Prozedur befolgen. Sie können einen oder mehrere der folgenden Werte angeben.

Wert	Protokoll
SSLv3	SSL 3.0
TLSv1	TLS 1.0
TLSv1.1	TLS 1.1
TLSv1.2	TLS 1.2
TLSv1.3	TLS 1.3

Verwenden Sie den Eintrag, der in der Spalte **Value** (Wert) der Routeneigenschaft angegeben ist. Der zugehörige Eintrag in der Spalte **Protocol** (Protokoll) dient nur zur Information.

SSLClientSiteDN_C

Verwenden Sie diese Eigenschaft zur Angabe eines Ländernamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten

muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Ländernamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_CN

Verwenden Sie diese Eigenschaft zur Angabe eines allgemeinen Namens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_DC

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für die Domänenkomponente, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domänennamen dar. Der Domänename `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Domänenkomponente akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_DNQ

Verwenden Sie diese Eigenschaft zur Angabe einer Domäne für ein Qualifikationsmerkmal, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Qualifikationsmerkmal akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_L

Verwenden Sie diese Eigenschaft zur Angabe eines Positionsnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Position akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_O

Verwenden Sie diese Eigenschaft zur Angabe eines Unternehmensnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_OU

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für eine Organisationseinheit (OU), mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf Wahrgesetzt ist), wird die Route gestoppt

und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLClientSiteDN_PC

Verwenden Sie diese Eigenschaft zur Angabe einer Postleitzahl, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Postleitzahl akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_ST

Verwenden Sie diese Eigenschaft zur Angabe eines Statusnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für den Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_Street

Verwenden Sie diese Eigenschaft zur Angabe eines Straßennamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_T

Verwenden Sie diese Eigenschaft zur Angabe eines Titels, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteDN_UID

Verwenden Sie diese Eigenschaft zur Angabe einer Benutzer-ID, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Benutzer-ID akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLClientSiteLabel

Verwenden Sie diese Eigenschaft zur Angabe einer Bezeichnung, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Bezeichnung akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLExitData

Verwenden Sie diese Eigenschaft, um eine benutzerdefinierte Zeichenfolge bereitzustellen, die an den Exit übergeben werden soll.

SSLExitName

Mit dieser Eigenschaft können Sie den Klassennamen für den Exit definieren, der aufgerufen wird, wenn eine Route als SSL/TLS-Client oder SSL/TLS-Server fungiert. Der Name muss einen beliebigen Paketnamen enthalten, z. B. `com.ibm.mq.ipc.exit.TestExit`.

SSLExitPath

Mit dieser Eigenschaft können Sie die Position des Exits definieren, der zum Laden einer Kopie des Exits verwendet wird. Der Name muss ein vollständig qualifizierter Name sein, der für die Suche nach der Klassendatei oder dem Namen einer `.jar`-Datei mit der Klassendatei verwendet werden kann, wie beispielsweise `C:\mqipt\exits` oder `C:\mqipt\exits\exits.jar`.

SSLExitTimeout

Definieren Sie mit dieser Eigenschaft, wie lange MQIPT auf den Abschluss des Exits wartet, bevor die Verbindungsanforderung beendet wird. Der Wert 0 gibt an, dass MQIPT unendlich wartet.

SSLPlainConnections

Mit dieser Eigenschaft können Sie angeben, ob SSL/TLS für Verbindungen zum MQIPT-Listener-Port einer Route verbindlich ist, die für das Akzeptieren eingehender SSL/TLS-Verbindungen konfiguriert ist. Diese Eigenschaft gilt für Routen, für die die Eigenschaft **SSLServer** oder **SSLProxyMode** auf "true" gesetzt ist. Wenn diese Eigenschaft aktiviert ist, können unverschlüsselte Verbindungen zum Listener-Port der Route hergestellt werden, wodurch MQIPT alle IBM MQ-Verbindungen zum Listener-Port des Warteschlangenmanagers weiterleiten kann, unabhängig davon, ob die Verbindung verschlüsselt ist. Wenn Sie diesen Parameter nicht festlegen oder ihn auf false setzen, sind nur eingehende SSL/TLS-Verbindungen zulässig. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

SSLProxyMode

Setzen Sie diese Eigenschaft auf true, damit die Route nur SSL/TLS-Clientverbindungsanforderungen akzeptiert und die Anforderung direkt an das Ziel übertragen wird. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu dieser Route werden gestoppt. Diese Eigenschaft kann nicht in Verbindung mit den folgenden Eigenschaften verwendet werden:

- **SocksClient**
- **SocksServer**
- **SSLClient**
- **SSLServer**

SSLServer

Setzen Sie diese Eigenschaft auf true, damit die Route als SSL/TLS-Server fungiert und eingehende SSL/TLS-Verbindungen akzeptiert. Wenn Sie **SSLServer** auf wahr setzen, bedeutet dies, dass der Aufrufende ein anderer MQIPT ist, der als SSL/TLS-Client fungiert, oder ein IBM MQ-Client oder -Warteschlangenmanager mit aktiviertem SSL/TLS.

Wenn Sie **SSLServer** auf true setzen, müssen Sie mit der Eigenschaft **SSLServerKeyRing** einen SSL/TLS-Serverschlüsselring festlegen oder MQIPT für die Verwendung von Verschlüsselungshardware konfigurieren, indem Sie die Eigenschaft **SSLServerKeyRingUseCryptoHardware** festlegen.

Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

Diese Eigenschaft kann nicht in Verbindung mit den folgenden Eigenschaften verwendet werden:

- **SocksServer**
- **SSLProxyMode**

SSLServerCAKeyRing

Der vollständig qualifizierte Dateiname der Schlüsselringdatei mit CA-Zertifikaten, die für die Authentifizierung von Zertifikaten vom SSL/TLS-Client verwendet werden. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf wahr gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLServerCAKeyRingPW

Das Kennwort zum Öffnen der CA-Schlüsselringdatei des SSL/TLS-Servers, die mit der Eigenschaft **SSLServerCAKeyRing** angegeben ist, oder zum Herstellen einer Verbindung zum Schlüssel Speicher der Verschlüsselungshardware, wenn die Eigenschaft **SSLServerCAKeyRingUseCryptoHardware** auf wahr gesetzt ist

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten

Backslash (\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptpw** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerCAKeyRingUseCryptoHardware

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für die CA-Zertifikate verwendet wird, die zur Authentifizierung von Zertifikaten vom SSL/TLS-Client verwendet werden. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLServerCAKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Um diese Funktion nutzen zu können, muss der lokale Warteschlangenmanager, der über die MQIPT -Route verbunden ist, auch über die Berechtigung IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `true` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** wird gesetzt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

SSLServerAskClientAuth

Verwenden Sie diese Eigenschaft für die Anforderung der SSL/TLS-Clientauthentifizierung durch den SSL/TLS-Server. Der SSL/TLS-Client muss über ein eigenes Zertifikat verfügen, das an den SSL/TLS-Server gesendet wird. Das Zertifikat wird aus der Schlüsselringdatei abgerufen. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLServerCipherSuites

Der Name der SSL/TLS-CipherSuite, die auf der Seite des SSL/TLS-Servers verwendet werden soll. Dabei kann es sich um eine oder mehrere der unterstützten CipherSuites handeln. Wenn Sie dieses Feld leer lassen, wird jede CipherSuite für die aktivierten Protokolle verwendet, die mit dem Serverzertifikat im Schlüsselring kompatibel ist. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLServerDN_C

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Ländernamen empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Unternehmensnamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_CN

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem allgemeinen Namen empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_DC

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dem Namen dieser Domänenkomponente empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domänennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Domänenkomponente akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_DNQ

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Qualifikationsmerkmal für die Domäne empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Qualifikationsmerkmal akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_L

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dieser Position empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Position akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_O

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Unternehmen empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_OU

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client dieser Organisationseinheit (OU) empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLServerDN_PC

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dieser Postleitzahl empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Postleitzahl akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_ST

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Status empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_Street

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Straßennamen empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_T

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Titel empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerDN_UID

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dieser Benutzer-ID empfangen werden. Der Name kann mit einem Stern (*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Benutzer-ID akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerExit

Mit dieser Eigenschaft können Sie die Verwendung eines Exits aktivieren oder inaktivieren, wenn die Route als SSL/TLS-Server fungiert. Auf diese Weise können Sie die Exitdetails in der Konfigurationsdatei definieren, ohne dass sie tatsächlich verwendet werden.

SSLServerKeyRing

Der vollständig qualifizierte Dateiname der Schlüsselringdatei, die das Serverzertifikat enthält. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerKeyRingPW

Das Kennwort zum Öffnen der mit der Eigenschaft **SSLServerKeyRing** angegebenen SSL/TLS-Serverschlüsselringdatei oder zum Herstellen einer Verbindung zum Schlüssel Speicher der Verschlüsselungshardware, wenn die Eigenschaft **SSLServerKeyRingUseCryptoHardware** auf **Wahr** gesetzt ist.

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Sie müssen **SSLServerKeyRingPW** angeben, wenn Sie **SSLServer** auf `Wahr` setzen.

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerKeyRingUseCryptoHardware

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für das Serverzertifikat verwendet wird, wenn MQIPT als SSL/TLS-Server agiert. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLServerKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Um diese Funktion nutzen zu können, muss der lokale Warteschlangenmanager, der über die MQIPT -Route verbunden ist, auch über die Berechtigung IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `true` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** wird gesetzt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

SSLServerProtocols

Wird verwendet, um die Gruppe aktivierter Secure Socket-Protokolle zu beschränken, die verwendet werden, um eingehende Verbindungen zum Route-Listener-Port für eine Route zu akzeptieren, wenn **SSLServer** auf `Wahr` gesetzt ist.

Sie können mehrere Werte angeben, indem Sie sie mit Kommas trennen. Wenn Sie diese Eigenschaft nicht angeben, sind TLS 1.2 und TLS 1.3 standardmäßig aktiviert. Um andere Protokolle als TLS 1.2 oder TLS 1.3 zu aktivieren, müssen Sie die zu aktivierenden Protokolle in dieser Eigenschaft angeben und außerdem Unterstützung für das Protokoll in der Java runtime environment hinzufügen, indem Sie die im Abschnitt [Veraltete Protokolle und CipherSuites](#) beschriebene Prozedur befolgen. Sie können einen oder mehrere der folgenden Werte angeben.

Wert	Protokoll
SSLv3	SSL 3.0
TLSv1	TLS 1.0
TLSv1.1	TLS 1.1
TLSv1.2	TLS 1.2
TLSv1.3	TLS 1.3

Verwenden Sie den Eintrag, der in der Spalte **Value** (Wert) der Routeneigenschaft angegeben ist. Der zugehörige Eintrag in der Spalte **Protocol** (Protokoll) dient nur zur Information.

SSLServerSiteDN_C

Verwenden Sie diese Eigenschaft zur Angabe eines Ländernamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Ländernamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_CN

Verwenden Sie diese Eigenschaft zur Angabe eines allgemeinen Namens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_DC

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für die Domänenkomponente, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domännennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Domänenkomponente akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_DNQ

Verwenden Sie diese Eigenschaft zur Angabe einer Domäne für ein Qualifikationsmerkmal, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Qualifikationsmerkmal akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_L

Verwenden Sie diese Eigenschaft zur Angabe eines Positionsnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Position akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_O

Verwenden Sie diese Eigenschaft zur Angabe eines Unternehmensnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_OU

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für eine Organisationseinheit (OU), mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

SSLServerSiteDN_PC

Verwenden Sie diese Eigenschaft zur Angabe einer Postleitzahl, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben,

werden Zertifikate mit einer beliebigen Postleitzahl akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_ST

Verwenden Sie diese Eigenschaft zur Angabe eines Statusnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für den Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_Street

Verwenden Sie diese Eigenschaft zur Angabe eines Straßennamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_T

Verwenden Sie diese Eigenschaft zur Angabe eines Titels, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteDN_UID

Verwenden Sie diese Eigenschaft zur Angabe einer Benutzer-ID, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Benutzer-ID akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

SSLServerSiteLabel

Verwenden Sie diese Eigenschaft zur Angabe einer Bezeichnung, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Bezeichnung akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

StoredCredentialsFormat

Mit dieser Eigenschaft können Sie angeben, ob die Werte der Kennworteigenschaften das von MQIPT ab IBM MQ 9.1.5 unterstützte Format für verschlüsselte Kennwörter verwendet. MQIPT kann fast immer erkennen, ob Kennwörter im Format für verschlüsselte Kennwörter angegeben sind. Diese Eigenschaft muss nur in dem unwahrscheinlichen Fall festgelegt werden, in dem MQIPT nicht automatisch zwischen einem verschlüsselten Kennwort und einem Klartextkennwort oder einem Dateinamen unterscheiden kann.

Dieser kann einen der folgenden Werte annehmen:

encrypted

Kennworteigenschaften enthalten ein verschlüsseltes Kennwort in dem in MQIPT ab IBM MQ 9.1.5 unterstützten Format.

compat

Kennworteigenschaften enthalten entweder ein Klartextkennwort oder bei Schlüsselringkennwörtern den Namen der Datei, die ein verschlüsseltes Kennwort enthält.

TCPKeepAlive

Setzen Sie diese Eigenschaft auf `true`, um das regelmäßige Senden von TCP/IP-Keepalive-Paketen zu aktivieren, mit denen verhindert wird, dass Verbindungen in dieser Route inaktiv werden. Dies

verringert die Wahrscheinlichkeit, dass MQIPT-Verbindungen durch eine Firewall oder einen Router getrennt werden. Das Senden von TCP/IP-Keepalive-Paketen wird durch Optimierungsparameter des Betriebssystems gesteuert. Weitere Informationen zum Einstellen von Keepalive-Paketen finden Sie in der Dokumentation zu Ihrem Betriebssystem. Wenn Sie diesen Parameter nicht festlegen oder ihn auf `false` setzen, werden keine Keepalive-Pakete gesendet.

Trace

Die für diese Route erforderliche Tracestufe. Durch das Aktivieren der Tracefunktion für eine Route wird die Tracefunktion gleichzeitig nicht für andere Routen aktiviert. Wenn Sie mehrere Routen verfolgen müssen, müssen Sie die Eigenschaft **Trace** zum Abschnitt `[route]` jeder Route hinzufügen, für die ein Trace erstellt werden soll.

Diese Eigenschaft kann einen der folgenden Werte haben:

0

Trace ist nicht aktiviert

Jede beliebige positive Ganzzahl.

Trace ist aktiviert

Der Standardwert ist 0.

Wenn der Abschnitt `[route]` keine Eigenschaft **Trace** enthält, wird die Eigenschaft **Trace** aus dem Abschnitt `[global]` verwendet. Weitere Informationen zur Tracefunktion für Threads, die keiner Route zugeordnet sind, finden Sie unter **Trace** im Abschnitt `[global]`. Wenn sich eine Änderung dieser Eigenschaft auf eine Route auswirkt, wird bei der Ausgabe eines Aktualisierungsbefehls der neue Wert verwendet. Alle Verbindungen verwenden sofort den neuen Wert. Die Route wird nicht gestoppt.

TraceUser-Daten

Die Menge der Benutzerdaten in den von dieser Route empfangenen und gesendeten Netzübertragungen, für die ein Trace durchgeführt wird, wenn der Trace für diese Route aktiviert ist. Dieser kann einen der folgenden Werte annehmen:

0

Es wird kein Trace für Benutzerdaten erstellt.

Alle

Alle Benutzerdaten werden verfolgt.

numberOfByte

Die angegebene Anzahl von Datenbyte, einschließlich des Übertragungssegmentheaders (TSH), wird verfolgt. Der angegebene Wert muss größer als 15 sein.

UriName

Mit dieser Eigenschaft kann der Name eines Uniform Resource Identifier (URI) der Ressource bei der Verwendung eines HTTP-Proxys geändert werden, obwohl der Standardwert für die meisten Konfigurationen ausreicht:

```
HTTP://destination:destination_port/mqipt
```

Wenn Sie diese Eigenschaft ändern (und **HTTP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird.

mqiptAdminEigenschaften

Der Befehl **mqiptAdmin** liest Konfigurationseigenschaften aus einer Eigenschaftendatei, die angegeben wird, wenn der Befehl gestartet wird.

Die folgenden Eigenschaften können in der Eigenschaftendatei angegeben werden, die mit dem Befehl **mqiptAdmin** verwendet wird. Bei Eigenschaftsnamen muss die Groß-/Kleinschreibung beachtet werden.

PasswordProtectionKeyFile

Der Name der Datei mit dem Verschlüsselungsschlüssel, der zum Verschlüsseln des Truststore-Kennworts verwendet wird, das in der Eigenschaft **SSLClientCAKeyRingPW** angegeben ist. Wenn diese Eigenschaft nicht angegeben wird, wird der Standardverschlüsselungsschlüssel für die Entschlüsselung des Kennworts verwendet. Der Verschlüsselungsschlüssel, der für die Verschlüsselung des **mqiptAdmin**-Trust-Store-Kennworts verwendet wird, kann vom Verschlüsselungsschlüssel für die Verschlüsselung von Kennwörtern in der `mqipt.conf`-Konfigurationsdatei abweichen.

SSLClientCAKeyRing

Der Dateiname des PKCS#12-Truststore, der für Verbindungen zum TLS-Befehlsport von MQIPT verwendet werden soll. Der Truststore sollte das CA-Zertifikat der Zertifizierungsstelle enthalten, die das Serverzertifikat signiert hat, das der TLS-Befehlsport von MQIPT laut Konfiguration verwenden soll. Backslash-Zeichen(\) im Dateinamen müssen Escapezeichen vorangestellt und als doppelter Backslash (\\) angegeben werden.

SSLClientCAKeyRingPW

Das verschlüsselte Kennwort für den Zugriff auf den mit der Eigenschaft **SSLClientCAKeyRing** angegebenen Truststore. Das Kennwort muss mit dem Befehl **mqiptPW** verschlüsselt werden, und der Wert dieser Eigenschaft ist auf die Zeichenfolgeausgabe von **mqiptPW** gesetzt.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in diesem Dokument beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf Produkte, Programme oder Services von IBM bedeuten nicht, dass nur Produkte, Programme oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East & Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmieretechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos ohne Zahlung an IBM in jeder Form kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben sind. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Informationen zu Programmierschnittstellen

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

Dieses Handbuch enthält Informationen zu geplanten Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zum Abrufen der Services von IBM 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 eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (<https://www.eclipse.org/>) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.



Teilenummer:

(1P) P/N: