

9.0

Verwaltung von IBM MQ

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 503 gelesen werden.

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

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

© **Copyright International Business Machines Corporation 2015, 2023.**

Inhaltsverzeichnis

Verwalten.....	5
Lokale und ferne Verwaltung.....	8
IBM MQ mithilfe von Steuerbefehlen verwalten.....	9
MQ mit Hilfe von MQSC-Befehlen verwalten.....	10
Scriptbefehle (MQSC).....	11
IBM MQ-Objektnamen in MQSC-Befehlen.....	13
Standardeingabe und-ausgabe.....	14
MQSC-Befehle interaktiv verwenden.....	14
MQSC-Befehle aus Textdateien ausführen.....	16
MQSC-Befehle aus Stapeldateien ausführen.....	19
Probleme mit MQSC-Befehlen beheben.....	20
IBM MQ-Verwaltung mit PCF-Befehlen automatisieren.....	21
Einführung in IBM MQ Programmable Command Formats.....	22
Verwendung von MQAI zur Vereinfachung der Verwendung von PCFs.....	35
Verwaltung über die REST API.....	71
Erste Schritte mit der administrative REST API.....	72
administrative REST API verwenden.....	78
Fernverwaltung über die REST API.....	84
REST API-Zeitmarken.....	88
REST API-Fehlerbehandlung.....	88
REST API-Erkennung.....	91
Unterstützung landessprachunterstützter Sprachen in REST API.....	92
Verwaltung über die IBM MQ Console.....	94
Erste Schritte mit der IBM MQ Console.....	95
Mit lokalen Warteschlangenmanagern arbeiten.....	98
Mit IBM MQ-Objekten arbeiten.....	100
Mit Berechtigungsdatensätzen arbeiten.....	117
Überwachen der Systemressourcennutzung.....	120
Dashboard-Layouts konfigurieren.....	133
Steuerelemente der Statusübersicht.....	134
Tastaturkurzbefehle.....	135
Verwaltung über die IBM MQ Explorer.....	135
Was Sie mit dem IBM MQ Explorer tun können.....	136
IBM MQ Explorer einrichten.....	137
IBM MQ-Taskleistanwendung verwenden (nur Windows).....	144
Die IBM MQ -Alertmonitoranwendung (nur Windows).....	144
Lokale IBM MQ-Objekte verwalten.....	145
Warteschlangenmanager starten und stoppen.....	145
MQI-Kanäle stoppen.....	148
Mit Warteschlangenmanagern arbeiten.....	149
Mit lokalen Warteschlangen arbeiten.....	151
Mit Aliaswarteschlangen arbeiten.....	156
Mit dead-letter-Warteschlangen arbeiten.....	158
Mit Modellwarteschlangen arbeiten.....	179
Mit Verwaltungsthemen arbeiten.....	180
Mit Subskriptionen arbeiten.....	183
Mit Services arbeiten.....	187
Objekte zum Auslösen verwalten.....	194
Verwenden des Dienstprogramms dmpmqmsg zwischen zwei Systemen.....	196
Ferne IBM MQ-Objekte verwalten.....	200
Kanäle und ferne Warteschlangen.....	200
Ferne Verwaltung von einem lokalen WS-Manager.....	202

Lokale Definition einer fernen Warteschlange erstellen.....	208
Prüfen, ob asynchrone Befehle für verteilte Netze abgeschlossen sind.....	211
Ferne Warteschlangendefinitionen als Aliasnamen verwenden.....	213
Datenkonvertierung.....	214
MQ Telemetry verwalten.....	219
Warteschlangenmanager unter Linux und AIX für Telemetrie konfigurieren.....	219
Warteschlangenmanager unter Windows für Telemetrie konfigurieren.....	221
Verteilte Steuerung von Warteschlangen für Senden von Nachrichten an MQTT-Clients konfigurieren.....	223
MQTT-Client - Identifikation, Berechtigung und Authentifizierung.....	226
Telemetriekanal-Authentifizierung mit TLS.....	232
Datenschutz auf Telemetriekanälen.....	234
TLS-Konfiguration von MQTT-Java-Clients und Telemetriekanälen.....	235
JAAS-Konfiguration für Telemetriekanal.....	240
IBM MQ Light verwalten.....	242
Von MQ Light-Clients verwendete IBM MQ-Objekte anzeigen.....	242
MQ Light-Client - Identifikation, Berechtigung und Authentifizierung.....	244
Datenschutz bei der Veröffentlichung von Kanälen.....	246
MQ Light-Clients mit TLS konfigurieren.....	247
Verbindung zwischen MQ Light-Clients und dem Warteschlangenmanager trennen.....	248
Multicast verwalten.....	248
Erste Schritte mit Multicasting.....	248
IBM MQ Multicast-Thementopologie.....	250
Größe von Multicastnachrichten steuern.....	251
Datenkonvertierung für Multicast-Messaging aktivieren.....	253
Multicast-Anwendungsüberwachung.....	254
Multicast-Nachrichtenzuverlässigkeit.....	254
Erweiterte Multicast-Tasks.....	255
IBM MQ for IBM i verwalten.....	258
IBM MQ for IBM i mithilfe von CL-Befehlen verwalten.....	259
Alternative Methoden zur Verwaltung von IBM MQ for IBM i.....	273
Ablaufsteuerung für IBM i.....	278
Verfügbarkeit, Sicherung, Wiederherstellung und Neustart unter IBM i.....	287
IBM MQ for IBM i in den Quiescemodus versetzen.....	333
IBM MQ for z/OS verwalten.....	337
Befehle für IBM MQ for z/OS ausgeben.....	338
Dienstprogramme von IBM MQ for z/OS.....	346
IBM MQ for z/OS betreiben.....	349
Programme zur Verwaltung von IBM MQ for z/OS schreiben.....	371
IBM MQ-Ressourcen unter z/OS verwalten.....	383
Wiederherstellung und Neustart unter z/OS.....	426
IBM MQ und IMS.....	450
Advanced Message Security unter z/OS betreiben.....	463
IBM MQ for z/OS Service-Provider für z/OS Connect.....	464
Bemerkungen.....	503
Informationen zu Programmierschnittstellen.....	504
Marken.....	505



IBM MQ verwalten

Wenn Sie Ihre IBM MQ-Warteschlangenmanager und die zugehörigen Ressourcen verwalten möchten, wählen Sie Ihre bevorzugte Methode aus einer Reihe von Tasks aus, die Sie zum Aktivieren und Verwalten dieser Ressourcen verwenden können.






Sie können IBM MQ-Objekte lokal oder fern verwalten (siehe [„Lokale und ferne Verwaltung“](#) auf Seite 8).


Es gibt eine Reihe unterschiedlicher Methoden, die Sie zum Erstellen und Verwalten Ihrer Warteschlangenmanager und ihrer zugehörigen Ressourcen in IBM MQ verwenden können. Zu diesen Methoden gehören Befehlszeilenschnittstellen, grafische Benutzerschnittstellen und eine Verwaltungs-API.

Es gibt verschiedene Gruppen von Befehlen, die Sie abhängig von der Plattform für die Verwaltung von IBM MQ verwenden können:

- [„IBM MQ-Steuerbefehle“](#) auf Seite 5
- [„IBM MQ-Scriptbefehle \(MQSC\)“](#) auf Seite 6
- [„Programmierbare Befehlsformate \(PCFs\)“](#) auf Seite 6
-  [Die administrative REST API](#)
-  [„IBM i-Steuersprache“](#) auf Seite 7

Es gibt auch die folgenden anderen Optionen zum Erstellen und Verwalten von IBM MQ-Objekten:

-   [„IBM MQ Explorer“](#) auf Seite 7
-  [„IBM MQ Console“](#) auf Seite 8
-  [„Windows-Standardkonfigurationsanwendung“](#) auf Seite 8
-  [„Microsoft Cluster Service \(MSCS\)“](#) auf Seite 8

 Informationen zu den Verwaltungsschnittstellen und -optionen in IBM MQ for z/OS finden Sie im Abschnitt [„IBM MQ for z/OS verwalten“](#) auf Seite 337.

Sie können einige Verwaltungs- und Überwachungstasks sowohl für lokale als auch für ferne Warteschlangenmanager mithilfe von PCF-Befehlen automatisieren. Diese Befehle können auch durch die Verwendung der IBM MQ-Verwaltungsschnittstelle (MQAI) auf einigen Plattformen vereinfacht werden. Weitere Informationen zum Automatisieren von Verwaltungstasks finden Sie im Abschnitt [„IBM MQ-Verwaltung mit PCF-Befehlen automatisieren“](#) auf Seite 21.

IBM MQ-Steuerbefehle



Sie können Verwaltungstasks für Warteschlangenmanager selbst ausführen, indem Sie die Steuerbefehle verwenden.

Systeme unter IBM MQ for Windows, UNIX and Linux®-Systeme stellen die *Steuerbefehle* bereit, die Sie in der Systembefehlszeile eingeben.

Eine Beschreibung der Steuerbefehle finden Sie im Abschnitt [Warteschlangenmanager auf Multiplatforms erstellen und verwalten](#). Die Befehlsreferenz für die Steuerbefehle finden Sie in [IBM MQ-Steuerbefehle](#).

IBM MQ-Scriptbefehle (MQSC)

Verwenden Sie MQSC-Befehle zum Verwalten von Warteschlangenmanagerobjekten, einschließlich des Warteschlangenmanagers selbst, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsinformationsobjekte.

MQSC-Befehle werden mit dem Befehl `runmqsc` an einen Warteschlangenmanager ausgegeben. Sie können dies interaktiv ausführen, Befehle über eine Tastatur ausgeben oder die Standardeingabeeinheit (`stdin`) umleiten, um eine Folge von Befehlen aus einer ASCII-Textdatei auszuführen. In beiden Fällen ist das Format der Befehle identisch.

Sie können den Befehl `runmqsc` in drei Modi ausführen, abhängig von den Flags, die im Befehl festgelegt wurden:

- *Prüfmodus* , bei dem die MQSC-Befehle in einem lokalen Warteschlangenmanager geprüft werden, aber nicht ausgeführt werden
- *Direktmodus* : Die MQSC-Befehle werden auf einem lokalen Warteschlangenmanager ausgeführt.
- *Indirekter Modus* , in dem die MQSC-Befehle auf einem fernen Warteschlangenmanager ausgeführt werden

MQSC-Befehle sind auf allen Plattformen verfügbar, einschließlich IBM i und z/OS. MQSC-Befehle werden in [Befehlssätzen verglichen](#) zusammengefasst.

ULW Unter UNIX, Linux, and Windows können Sie MQSC-Befehle als einzelne Befehle verwenden, die in der Systembefehlszeile ausgegeben werden. Um komplexere oder mehrere Befehle ausgeben zu können, kann der MQSC in eine Datei integriert werden, die Sie über die Befehlszeile ausführen. MQSC-Befehle können an einen fernen Warteschlangenmanager gesendet werden. Ausführliche Informationen hierzu finden Sie im Abschnitt [Befehlsscripts erstellen](#) .

IBM i Um die Befehle auf einem IBM i-Server auszugeben, müssen Sie eine Liste der Befehle in einer Scriptdatei erstellen und anschließend die Datei mit dem Befehl `STRMQMMQSC` ausführen.

Anmerkungen: **IBM i**

1. Verwenden Sie die Bibliothek `QTEMP` nicht als Eingabebibliothek für `STRMQMMQSC`, da die Verwendung der Bibliothek `QTEMP` begrenzt ist. Sie müssen eine andere Bibliothek als Eingabedatei für den Befehl verwenden.
2. Unter IBM i werden MQSC-Antworten auf Befehle, die von einer Scriptdatei ausgegeben werden, in einer Spooldatei zurückgegeben.

„[Scriptbefehle \(MQSC\)](#)“ auf Seite 11 enthält eine Beschreibung der einzelnen MQSC-Befehle und ihre Syntax.

Weitere Informationen zur Verwendung von MQSC-Befehlen finden Sie im Abschnitt [„MQ mit Hilfe von MQSC-Befehlen verwalten“](#) auf Seite 10.

Programmierbare Befehlsformate (PCFs)

Programmierbare Befehlsformate (PCFs) definieren Befehls- und Antwortnachrichten, die zwischen einem Programm und einem beliebigen WS-Manager (der PCFs unterstützt) in einem Netz ausgetauscht werden können. Sie können PCF-Befehle in einem Systemverwaltungsanwendungsprogramm für die Verwaltung von IBM MQ-Objekten verwenden: Authentifizierungsinformationsobjekte, Kanäle, Kanallistener, Namenslisten, Prozessdefinitionen, Warteschlangenmanager, Warteschlangen, Services und Speicherklassen. Die Anwendung kann von einem einzigen Punkt im Netz aus ausgeführt werden, um die Befehls- und Antwortinformationen mit jedem WS-Manager, lokal oder fern mit dem lokalen WS-Manager zu kommunizieren.

Weitere Informationen zu PCFs finden Sie im Abschnitt [„Einführung in IBM MQ Programmable Command Formats“](#) auf Seite 22.

Informationen zur Definition von PCFs und Strukturen für die Befehle und Antworten finden Sie im Abschnitt [Programmierbare Befehlsformate-Referenz](#) .

administrative REST API

V 9.0.1

Die administrative REST API stellt eine REST-konforme Schnittstelle bereit, die Sie zur Verwaltung von IBM MQ verwenden können. Wenn Sie die administrative REST API verwenden, rufen Sie eine HTTP-Methode für eine URL auf, die ein IBM MQ-Objekt darstellt. Sie können z. B. Informationen zu IBM MQ-Installationen mithilfe der HTTP-Methode GET unter der folgenden URL anfordern:

V 9.0.4

IBM MQ 9.0.4 und höher:

```
https://localhost:9443/ibmmq/rest/v1/admin/installation
```

IBM MQ 9.0.3 und früher:

```
https://localhost:9443/ibmmq/rest/v1/installation
```

Sie können die administrative REST API mit der HTTP/REST-Implementierung einer Programmiersprache oder mithilfe von Tools wie beispielsweise cURL oder eines Browser-Add-ons für REST-Clients verwenden.

Weitere Informationen finden Sie im Abschnitt [administrative REST API](#).

IBM i-Steuersprache

IBM i

Diese Sprache kann zur Ausgabe von Verwaltungsbefehlen an IBM MQ for IBM i verwendet werden. Die Befehle können entweder in der Befehlszeile oder durch das Schreiben eines CL-Programms abgesetzt werden. Diese Befehle führen ähnliche Funktionen wie PCF-Befehle aus, aber das Format ist anders. CL-Befehle sind ausschließlich für Server und CL-Antworten konzipiert. Die PCF-Befehle sind plattformunabhängig und die beiden Befehls- und Antwortformate sind für den Programmenebenbenutzung vorgesehen.

Ausführliche Informationen zur IBM i-Steuersprache (Control Language, CL) finden Sie in [IBM MQ for IBM i-CL-Befehle](#).

IBM MQ Explorer

Windows

Linux

Mit dem IBM MQ Explorer können Sie die folgenden Aktionen ausführen:

- Definieren und steuern Sie verschiedene Ressourcen, wie z. B. Warteschlangenmanager, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Cluster.
- Starten oder Stoppen eines lokalen Warteschlangenmanagers und der zugehörigen Prozesse.
- Zeigen Sie die Warteschlangenmanager und die zugehörigen Objekte auf Ihrer Workstation oder von anderen Workstations an.
- Überprüfen Sie den Status von Warteschlangenmanagern, Clustern und Kanälen.
- Überprüfen Sie, welche Anwendungen, Benutzer oder Kanäle eine bestimmte Warteschlange geöffnet haben, aus dem Warteschlangenstatus.

Auf Windows- und Linux-Systemen können Sie IBM MQ Explorer starten, indem Sie das Systemmenü, die ausführbare Datei von MQExpLorer oder den Befehl **strmqcfig** verwenden.

Linux

Damit IBM MQ Explorer unter Linux gestartet werden kann, muss es möglich sein, eine Datei in Ihr Ausgangsverzeichnis zu schreiben; außerdem muss das Ausgangsverzeichnis vorhanden sein.

Weitere Informationen finden Sie unter [„Verwaltung über die IBM MQ Explorer“](#) auf Seite 135.

Sie können IBM MQ Explorer zum Verwalten von fernen Warteschlangenmanagern auf anderen Plattformen, einschließlich z/OS, verwenden, um Details anzuzeigen und das SupportPac MSOT herunterzuladen. Weitere Informationen finden Sie unter <https://www.ibm.com/support/docview.wss?uid=swg24021041>.

IBM MQ Console

V 9.0.1

Sie können den IBM MQ Console verwenden, um IBM MQ über einen Web-Browser zu verwalten.

Weitere Informationen finden Sie unter [„Verwaltung über die IBM MQ Console“](#) auf Seite 94.

Windows-Standardkonfigurationsanwendung

Windows

Sie können das Windows-Standardkonfigurationsprogramm verwenden, um eine *Starter*-Gruppe (oder Standardgruppe) von IBM MQ-Objekten zu erstellen. Eine Zusammenfassung der erstellten Standardobjekte finden Sie in [Tabelle 1: Objekte, die von der Windows-Standardkonfigurationsanwendung erstellt wurden](#).

Microsoft Cluster Service (MSCS)

Windows

Microsoft Cluster Service (MSCS) ermöglicht es Ihnen, Server in einem *Cluster* zu verbinden, wodurch eine höhere Verfügbarkeit von Daten und Anwendungen möglich ist und die Verwaltung des Systems vereinfacht wird. MSCS kann Server- oder Anwendungsfehler automatisch erkennen und wiederherstellen.

Es ist wichtig, Cluster im MSCS-Sinn nicht mit IBM MQ-Clustern zu verwechseln. Die Unterscheidung ist wie folgt:

IBM MQ-Cluster

Hierbei handelt es sich um Gruppen von mindestens zwei Warteschlangenmanagern auf einem oder mehreren Computern, die eine automatische Verbindung bereitstellen, und die gemeinsame Nutzung von Warteschlangen für Lastausgleich und Redundanz.

MSCS-Cluster

Hierbei handelt es sich um Gruppen von Computern, die miteinander verbunden und so konfiguriert sind, dass MSCS bei einem Ausfall eine *Funktionsübernahme* durchführt, die Statusdaten von Anwendungen vom ausgefallenen Computer auf einen anderen Computer im Cluster überträgt und ihre Operation dort erneut einleitet.

[Unterstützung des Microsoft -Clusterservice \(MSCS\)](#) enthält detaillierte Informationen zur Konfiguration Ihres IBM MQ for Windows -Systems für die Verwendung von MSCS.

Zugehörige Informationen

[Befehlsätze, Vergleich](#)

[IBM MQ - Technische Übersicht](#)

[Planung](#)

[Konfiguration](#)

Lokale und ferne Verwaltung

Sie können IBM MQ-Objekte lokal oder über Remotezugriff verwalten.

Lokale Verwaltung

Lokale Verwaltung bedeutet, dass Verwaltungstasks auf allen Warteschlangenmanagern ausgeführt werden, die Sie auf Ihrem lokalen System definiert haben. Sie können auf andere Systeme zugreifen, z. B. über das TCP/IP-Terminal-Emulationsprogramm **telnet**, und führen die Verwaltung dort aus. In IBM MQ

können Sie dies als lokale Verwaltung in Betracht ziehen, da es keine Kanäle gibt, d. h., die Kommunikation wird vom Betriebssystem verwaltet.

Fernverwaltung

IBM MQ unterstützt die Verwaltung von einem Single Point of Contact über die Fernverwaltung. Die ferne Verwaltung ermöglicht es Ihnen, Befehle von Ihrem lokalen System auszugeben, die auf einem anderen System verarbeitet werden und auch auf den IBM MQ Explorer angewendet werden. Sie können beispielsweise einen fernen Befehl absetzen, um eine Warteschlangendefinition in einem fernen Warteschlangenmanager zu ändern. Sie müssen sich nicht bei diesem System anmelden, obwohl Sie die entsprechenden Kanäle definiert haben müssen. Der Warteschlangenmanager und der Befehlsserver auf dem Zielsystem müssen aktiv sein.

Einige Befehle können auf diese Weise nicht abgesetzt werden, insbesondere die Erstellung oder den Start von Warteschlangenmanagern und den Start von Befehlsservern. Um diesen Tasktyp ausführen zu können, müssen Sie sich entweder am fernen System anmelden und die Befehle von dort absetzen oder einen Prozess erstellen, der die Befehle für Sie absetzen kann. Diese Einschränkung gilt auch für den IBM MQ Explorer.

Eine ausführlichere Beschreibung der Fernverwaltung finden Sie im Abschnitt [„Ferne IBM MQ-Objekte verwalten“](#) auf Seite 200.

ULW IBM MQ mithilfe von Steuerbefehlen verwalten

Mit den Steuerbefehlen können Sie eine Reihe von IBM MQ-Verwaltungstasks unter UNIX, Linux, and Windows ausführen.

Wenn Sie Steuerbefehle absetzen wollen, muss Ihre Benutzer-ID für die meisten Steuerbefehle Mitglied der Gruppe 'mqm' sein. Weitere Informationen hierzu finden Sie im Abschnitt [Berechtigung zum Verwalten von IBM MQ unter UNIX, Linux, and Windows](#). Beachten Sie außerdem die umgebungsspezifischen Informationen für die Plattform oder Plattformen, die Ihr Unternehmen verwendet.

Wenn Sie Steuerbefehle verwenden, die auf einem Warteschlangenmanager ausgeführt werden, müssen Sie den Befehl von der Installation aus verwenden, die dem Warteschlangenmanager zugeordnet ist, mit dem Sie arbeiten.

Bei Verwendung von Steuerbefehlen, die auf einem Warteschlangenmanager ausgeführt werden, der für die Verwendung der Verbindungsauthentifizierung mit CHCKLOCL (REQUIRED) konfiguriert ist, und ein Fehler beim Herstellen der Verbindung festgestellt wird, wird entweder

- Geben Sie eine Benutzer-ID und ein Kennwort an, wenn der Steuerbefehl dies zulässt.
- Verwenden Sie die MQSC-Entsprechungen der Steuerbefehle, sofern diese vorhanden sind.
- Starten Sie den WS-Manager mit der Option -ns, während die Steuerbefehle, die keine Verbindung herstellen können, ausgeführt werden müssen.

Eine vollständige Liste der Steuerbefehle finden Sie im Abschnitt [IBM MQ-Steuerbefehle](#).

Steuerbefehle auf Windows-Systemen verwenden

Windows

In IBM MQ for Windows werden Steuerbefehle an einer Eingabeaufforderung eingegeben.

Bei den Steuerbefehlen und den zugehörigen Flags muss die Groß-/Kleinschreibung nicht beachtet werden. Bei Argumenten für diese Befehle, wie z. B. Warteschlangennamen und Warteschlangenmanagernamen, wird die Groß-/Kleinschreibung beachtet.

Geben Sie beispielsweise im folgenden Befehl Folgendes ein:

```
crtmqm /u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- Der Befehlsname kann in Groß- oder Kleinbuchstaben eingegeben werden, oder es kann eine Mischung aus beiden eingegeben werden. Diese Werte sind alle gültig: `crtmqm`, `CRTMQM` und `CRTmqm`.
- Die Markierung kann als `-u`, `-U`, `/u` oder `/U` eingegeben werden.
- `SYSTEM.DEAD.LETTER.QUEUE` und `jupiter.queue.manager` müssen genau wie gezeigt eingegeben werden.

Steuerbefehle auf UNIX- und Linux-Systemen verwenden



auf IBM MQ für UNIX- und Linux-Systemen werden Steuerbefehle in einem Shellfenster eingegeben.

In UNIX-Umgebungen muss bei den Steuerbefehlen, einschließlich Befehlsnamen, den Flags und allen Argumenten die Groß-/Kleinschreibung beachtet werden. Geben Sie beispielsweise im folgenden Befehl Folgendes ein:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- Der Befehlsname muss `crtmqm` sein, nicht `CRTMQM`.
- Die Markierung muss `-u` sein, nicht `-U`.
- Die Warteschlange für nicht zustellbare Nachrichten wird als `SYSTEM.DEAD.LETTER.QUEUE` bezeichnet.
- Das Argument wird als `jupiter.queue.manager` angegeben, das sich nicht von `JUPITER.queue.manager` unterscheidet.

Geben Sie die Befehle genau so ein, wie sie in den Beispielen angezeigt werden.

Zugehörige Informationen

[Referenz zu IBM MQ-Steuerbefehlen](#)

MQ mit Hilfe von MQSC-Befehlen verwalten

Verwendung von MQSC-Befehlen zum Ausführen allgemeiner Tasks.

MQSC-Befehle sind auf allen Plattformen verfügbar, einschließlich IBM i und z/OS.

Sie können MQSC-Befehle zum Verwalten von WS-Manager-Objekten verwenden, einschließlich des Warteschlangenmanagers selbst, Warteschlangen, Prozessdefinitionen, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services, Namenslisten, Cluster und Authentifizierungsinformationsobjekte. Dieser Abschnitt befasst sich mit Warteschlangenmanagern, Warteschlangen und Prozessdefinitionen. Eine Übersicht über Kanal-, Clientverbindungskanal- und Listener-Objekte finden Sie unter [Objekte](#). Informationen zu allen MQSC-Befehlen zum Verwalten von WS-Manager-Objekten finden Sie in [„Scriptbefehle \(MQSC\)“](#) auf Seite 11.

MQSC-Befehle werden mit dem Befehl `runmqsc` an einen Warteschlangenmanager ausgegeben. (Weitere Informationen zu diesem Befehl finden Sie in `runmqsc`.) Sie können diese Option interaktiv ausführen, Befehle über eine Tastatur ausgeben oder die Standardeingabeeinheit (`stdin`) umleiten, um eine Folge von Befehlen aus einer ASCII-Textdatei auszuführen. In beiden Fällen ist das Format der Befehle identisch. (Informationen zum Ausführen der Befehle in einer Textdatei finden Sie im Abschnitt [„MQSC-Befehle aus Textdateien ausführen“](#) auf Seite 16.)

Sie können den Befehl `runmqsc` auf drei Arten ausführen, je nachdem, welche Flags im Befehl festgelegt sind:

- Überprüfen Sie einen Befehl, ohne ihn auszuführen. Dabei werden die MQSC-Befehle in einem lokalen WS-Manager geprüft, aber nicht ausgeführt.
- Führen Sie einen Befehl auf einem lokalen WS-Manager aus, auf dem die MQSC-Befehle auf einem lokalen Warteschlangenmanager ausgeführt werden.

- Führen Sie einen Befehl auf einem fernen WS-Manager aus, auf dem die MQSC-Befehle auf einem fernen Warteschlangenmanager ausgeführt werden.

Sie können den Befehl auch gefolgt von einem Fragezeichen ausführen, um die Syntax anzuzeigen.

Objektattribute, die in MQSC-Befehlen angegeben werden, werden in diesem Abschnitt in Großbuchstaben (z. B. QMNAME) angezeigt, obwohl die Groß-/Kleinschreibung nicht beachtet werden muss. Die Namen von MQSC-Befehlsattributen sind auf acht Zeichen begrenzt.

V 9.0.1 Für Continuous Delivery können Sie ab IBM MQ 9.0.1 durch Verwendung der Umgebungsvariablen **MQPROMPT** eine Eingabeaufforderung Ihrer Wahl festlegen. Neben Klartext ermöglicht die Variable **MQPROMPT** auch die Einfügung von Umgebungsvariablen unter Verwendung von +VARIABLE+ notation in derselben Weise wie die Serviceobjektdefinitionen von IBM MQ (siehe „Serviceobjekt definieren“ auf Seite 187). Beispiel:

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
sh> runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2023.
Starting MQSC for queue manager MY.QMGR.
username @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

V 9.0.0.1 Ab IBM MQ 9.0.0 Fix Pack 1 ist die Umgebungsvariable **MQPROMPT** auch im Release von Long Term Support verfügbar.

MQSC-Befehle werden im Abschnitt [MQSC-Befehle](#) detailliert beschrieben.

Windows **Linux** Auf Windows- oder Linux-Plattformen (x86- und x86-64-Plattformen) können Sie die in diesem Abschnitt beschriebenen Operationen auch mit dem IBM MQ Explorer ausführen. Weitere Informationen finden Sie unter „Verwaltung über die IBM MQ Explorer“ auf Seite 135.

Zugehörige Informationen

[runmqsc \(MQSC-Befehle ausführen\)](#)

[Referenzinformationen zu MQSC](#)

Scriptbefehle (MQSC)

MQSC-Befehle stellen eine einheitliche Methode zum Absetzen von lesbaren Befehlen auf IBM MQ-Plattformen bereit.

Das allgemeine Format der Befehle wird im Abschnitt [MQSC-Befehle](#) beschrieben.

Bei Verwendung von MQSC-Befehlen sollten die folgenden Regeln beachtet werden:

- Jeder Befehl beginnt mit einem Primärparameter (ein Verb), und es folgt ein sekundärer Parameter (ein Substantiv). Darauf folgt dann der Name oder der generische Name des Objekts (in runden Klammern), wenn es einen Namen gibt, der in den meisten Befehlen vorhanden ist. Danach können die Parameter in der Regel in beliebiger Reihenfolge auftreten; wenn ein Parameter einen entsprechenden Wert hat, muss der Wert direkt nach dem Parameter, auf den er sich bezieht, auftreten.

Anmerkung: **z/OS** Unter z/OS darf der sekundäre Parameter nicht der zweite sein.


- Schlüsselwörter, runde Klammern und Werte können durch eine beliebige Anzahl von Leerzeichen und Kommas voneinander getrennt werden. Ein Komma, das in den Syntaxdiagrammen angezeigt wird, kann immer durch ein oder mehrere Leerzeichen ersetzt werden. Es muss mindestens ein Leerzeichen vor jedem Parameter (hinter dem primären Parameter) stehen, außer unter z/OS.
- Eine beliebige Anzahl von Leerzeichen kann am Anfang oder Ende des Befehls und zwischen Parametern, Interpunktionszeichen und Werten auftreten. Der folgende Befehl ist z. B. gültig:

```
ALTER QLOCAL ('Account' ) TRIGDPTH ( 1)
```


Leerzeichen in einem Paar von Anführungszeichen sind signifikant.

- Zusätzliche Kommas können überall dort angezeigt werden, wo Leerzeichen zulässig sind, und sie werden so behandelt, als wären sie leer (es sei denn, es handelt sich um Zeichenfolgen, die in Anführungszeichen eingeschlossen sind).
- Wiederholte Parameter sind nicht zulässig. Die Wiederholung eines Parameters mit seiner "NO" -Version, wie in REPLACE NOREPLACE, ist ebenfalls nicht zulässig.
- Zeichenfolgen, die Leerzeichen, Kleinbuchstaben oder andere Sonderzeichen enthalten als:
 - Punkt (.)
 - Schrägstrich (/)
 - Unterstrich (_)
 - Prozentzeichen (%)

müssen in einfache Anführungszeichen eingeschlossen werden, es sei denn, sie sind:


-  Ausgabe erfolgt von den IBM MQ for z/OS-Operationen und -Steuerkonsolen
- Generische Werte, die mit einem Stern enden (diese müssen unter IBM i in einfache Anführungszeichen gesetzt werden)
- Ein einzelner Stern, z. B. TRACE(*) (in IBM i müssen diese in einfache Anführungszeichen eingeschlossen werden)
- Eine Bereichsspezifikation, die einen Doppelpunkt enthält, z. B. CLASS (01:03)

Wenn die Zeichenfolge selbst ein Hochkomma enthält, wird das Hochkomma durch zwei einfache Anführungszeichen dargestellt. Kleinbuchstaben, die nicht in Anführungszeichen enthalten sind, werden in Großbuchstaben gefaltet.

-  Unter Multiplattform wird eine Zeichenfolge, die keine Zeichen enthält (d. h. zwei einfache Anführungszeichen ohne Leerzeichen dazwischen), als Leerzeichen in einfachen Anführungszeichen interpretiert, d. h. auf dieselbe Weise wie (") interpretiert. Eine Ausnahme ist, wenn das verwendete Attribut eines der folgenden Attribute ist:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

dann werden zwei einfache Anführungszeichen ohne Leerzeichen als Zeichenfolge mit Nulllänge interpretiert.

 Wenn in z/OS ein Leerzeichen in einfache Anführungszeichen eingeschlossen werden soll, müssen Sie das Leerzeichen als solches eingeben ("). Eine Zeichenfolge, die keine Zeichen (") enthält, ist identisch mit der Eingabe ().

- In IBM WebSphere MQ 7.0 werden alle abschließenden Leerzeichen in den Zeichenfolgeattributen, die auf MQCHARV-Typen basieren, wie z. B. SELECTOR, untergeordnete Benutzerdaten, als bedeutsam behandelt, was bedeutet, dass 'abc' nicht gleich 'abc ' ist.
- Eine linke runde Klammer, gefolgt von einer rechten runden Klammer ohne signifikante Informationen dazwischen, z. B.

NAME ()

ist nicht gültig, es sei denn, es ist ausdrücklich vermerkt.

- Bei Schlüsselwörtern muss die Groß-/Kleinschreibung nicht beachtet werden: AltER, alter und ALTER sind alle akzeptabel. Alles, was in Anführungszeichen nicht enthalten ist, wird in Großbuchstaben geschrieben.

- Synonyme werden für einige Parameter definiert. DEF ist z. B. immer ein Synonym für DEFINE, daher ist DEF QLOCAL gültig. Synonyme sind jedoch nicht nur Mindestzeichenfolgen; DEFI ist kein gültiges Synonym für DEFINE.

Anmerkung: Es ist kein Synonym für den Parameter DELETE vorhanden. Dies soll verhindern, dass Objekte versehentlich gelöscht werden, wenn DEF, das Synonym für DEFINE, verwendet wird.

Eine Übersicht über die Verwendung von MQSC-Befehlen für die Verwaltung von IBM MQ finden Sie im Abschnitt [„MQ mit Hilfe von MQSC-Befehlen verwalten“](#) auf Seite 10.

MQSC-Befehle verwenden bestimmte Sonderzeichen, um bestimmte Bedeutungen zu haben. Weitere Informationen zu diesen Sonderzeichen und zur Verwendung dieser Sonderzeichen finden Sie im Abschnitt [Zeichen mit besonderer Bedeutung](#).

Weitere Informationen zum Erstellen von Scripts mit Hilfe von MQSC-Befehlen finden Sie im Abschnitt [Befehlsscripts erstellen](#).

Die vollständige Liste der MQSC-Befehle finden Sie im Abschnitt [MQSC-Befehle](#).

Sie können diesen Befehl aus 2CR-Quellen ausgeben. Eine Erläuterung der Quellensymbole finden Sie im Abschnitt [Befehle unter z/OS verwenden](#).

Zugehörige Konzepte

[„Einführung in IBM MQ Programmable Command Formats“](#) auf Seite 22

Programmierbare Befehlsformate (PCFs) definieren Befehls- und Antwortnachrichten, die zwischen einem Programm und einem beliebigen WS-Manager (der PCFs unterstützt) in einem Netz ausgetauscht werden können. PCFs vereinfachen die Verwaltung von Warteschlangenmanagern und andere Netzadministrationsysteme. Sie können eingesetzt werden, um das Problem der komplexen Verwaltung verteilter Netzwerke zu lösen, insbesondere, wenn Netzwerke in Größe und Komplexität wachsen.

Zugehörige Informationen

[Befehlsscripts erstellen](#)

IBM MQ-Objektnamen in MQSC-Befehlen

Vorgehensweise beim Verwenden von Objektnamen in MQSC-Befehlen.

In den Beispielen verwenden wir einige lange Namen für Objekte. Dies soll Ihnen helfen, den Typ des Objekts zu identifizieren, mit dem Sie es zu tun haben.

Wenn Sie MQSC-Befehle absetzen, müssen Sie nur den lokalen Namen der Warteschlange angeben. In unseren Beispielen verwenden wir Warteschlangennamen, wie z. B.:

```
ORANGE . LOCAL . QUEUE
```

Der LOCAL . QUEUE-Teil des Namens soll veranschaulichen, dass es sich bei dieser Warteschlange um eine lokale Warteschlange handelt. Es ist **nicht** für die Namen von lokalen Warteschlangen im Allgemeinen erforderlich.

Der Name saturn . queue . manager wird auch als Name des Warteschlangenmanagers verwendet. Der queue . manager-Teil des Namens soll veranschaulichen, dass es sich bei diesem Objekt um einen Warteschlangenmanager handelt. Für Namen von Warteschlangenmanagern ist er im Allgemeinen nicht erforderlich.

Groß-/Kleinschreibung in MQSC-Befehlen

MQSC-Befehle, einschließlich ihrer Attribute, können in Groß- oder Kleinbuchstaben geschrieben werden. Objektnamen in MQSC-Befehlen werden in Großbuchstaben geschrieben (also QUEUE und Warteschlange werden nicht unterschieden), es sei denn, die Namen sind in einfache Anführungszeichen eingeschlossen. Wenn Anführungszeichen nicht verwendet werden, wird das Objekt mit einem Namen in Großbuchstaben verarbeitet. Weitere Informationen finden Sie unter [Zeichen mit Sonderbedeutungen](#).

Beim Aufruf des Befehls `runmqsc` ist in einigen IBM MQ-Umgebungen in allen IBM MQ-Steuerbefehlen die Groß-/Kleinschreibung zu beachten. Weitere Informationen finden Sie im Abschnitt „IBM MQ mithilfe von Steuerbefehlen verwalten“ auf Seite 9.

Standardeingabe und-ausgabe

Die *Standardeingabeeinheit*, die auch als `stdin` bezeichnet wird, ist die Einheit, von der die Eingabe für das System ausgeführt wird. In der Regel ist dies die Tastatur, aber Sie können angeben, dass die Eingabe beispielsweise aus einem seriellen Anschluss oder einer Plattendatei stammen soll. Die *Standardausgabeeinheit*, die auch als `stdout` bezeichnet wird, ist die Einheit, an die die Ausgabe des Systems gesendet wird. Normalerweise ist dies eine Anzeige, aber Sie können die Ausgabe an einen seriellen Anschluss oder an eine Datei umleiten.

Bei Betriebssystembefehlen und IBM MQ -Steuerbefehlen leitet der Operator `<` die Eingabe um. Wenn diesem Operator ein Dateiname gefolgt wird, wird die Eingabe aus der Datei übernommen. Entsprechend leitet der Operator `>` die Ausgabe um. Wenn diesem Operator ein Dateiname folgt, erfolgt die Ausgabe in die genannte Datei.

MQSC-Befehle interaktiv verwenden

Sie können MQSC-Befehle interaktiv verwenden, indem Sie ein Befehlsfenster oder eine -Shell verwenden.

Wenn Sie MQSC-Befehle interaktiv verwenden möchten, öffnen Sie ein Befehlsfenster oder eine Shell und geben Sie Folgendes ein:

```
runmqsc
```

In diesem Befehl wurde kein Warteschlangenmanagername angegeben, so dass die MQSC-Befehle vom Standardwarteschlangenmanager verarbeitet werden. Wenn Sie einen anderen WS-Manager verwenden möchten, geben Sie den Namen des WS-Managers im Befehl **runmqsc** an. Verwenden Sie beispielsweise den folgenden Befehl, um MQSC-Befehle auf WS-Manager `jupiter.queue.manager` auszuführen:

```
runmqsc jupiter.queue.manager
```

Danach werden alle MQSC-Befehle, die Sie eingeben, von diesem WS-Manager verarbeitet, vorausgesetzt, er befindet sich auf demselben Knoten und ist bereits aktiv.

Jetzt können Sie bei Bedarf alle MQSC-Befehle eingeben. Versuchen Sie es zum Beispiel wie folgt:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Verwenden Sie für Befehle, die zu viele Parameter haben, die in eine Zeile passen, Fortsetzungszeichen, um anzugeben, dass ein Befehl in der folgenden Zeile fortgesetzt wird:

- Ein Minuszeichen (-) gibt an, dass der Befehl ab dem Anfang der folgenden Zeile fortgesetzt werden soll.
- Ein Pluszeichen (+) gibt an, dass der Befehl aus dem ersten nicht leeren Zeichen in der folgenden Zeile fortgesetzt werden soll.

Die Befehlseingabe endet mit dem letzten Zeichen einer nicht leeren Zeile, die kein Fortsetzungszeichen ist. Sie können die Befehlseingabe auch explizit beenden, indem Sie ein Semikolon (;) eingeben. (Dies ist besonders nützlich, wenn Sie versehentlich ein Fortsetzungszeichen am Ende der letzten Zeile der Befehlseingabe eingeben.)

Feedback von MQSC-Befehlen

Wenn Sie MQSC-Befehle absetzen, gibt der WS-Manager Bedienernachrichten zurück, die Ihre Aktionen bestätigen oder Ihnen mitteilen, welche Fehler Sie gemacht haben. Beispiel:

```
AMQ8006: IBM MQ queue created.
```

Diese Nachricht bestätigt, dass eine Warteschlange erstellt wurde.

```
AMQ8405: Syntax error detected at or near end of command segment below:-
```

```
AMQ8426: Valid MQSC commands are:
```

```
ALTER  
CLEAR  
DEFINE  
DELETE  
DISPLAY  
END  
PING  
REFRESH  
RESET  
RESOLVE  
RESUME  
START  
STOP  
SUSPEND  
4 : end
```

Diese Nachricht weist darauf hin, dass Sie einen Syntaxfehler gemacht haben.

Diese Nachrichten werden an die Standardausgabeeinheit gesendet. Wenn Sie den Befehl nicht korrekt eingegeben haben, lesen Sie die [MQSC-Befehle](#), um die korrekte Syntax zu erhalten.

Interaktive Eingabe von MQSC-Befehlen wird beendet

Geben Sie den Befehl END ein, um die Arbeit mit MQSC-Befehlen zu beenden.

Alternativ können Sie das EOF-Zeichen für Ihr Betriebssystem verwenden.

Zugehörige Konzepte

„MQSC-Befehle aus Textdateien ausführen“ auf Seite 16

Die Ausführung von MQSC-Befehlen ist interaktiv für Schnelltests geeignet. Wenn Sie jedoch über sehr lange Befehle verfügen oder eine bestimmte Folge von Befehlen wiederholt verwenden, sollten Sie die Umleitung von `stdin` aus einer Textdatei in Betracht ziehen.

Zugehörige Informationen

[runmqsc](#)

Befehl `runmqsc` unter UNIX und Linux

Die Befehlszeile von `runmqsc` in UNIX und Linux unterstützt den Befehlsrückruf, die Befehlsbeendigung und die Emacs-Befehlstasten.

Die folgenden Funktionen des Befehlszeileneditors sind verfügbar:

- Alle zuvor eingegebenen Befehle mit Hilfe der Aufwärtspfeiltaste und der Abwärtspfeiltaste erneut eingeben
- Automatischer Abschluss für das nächste Schlüsselwort eines Befehls unter Verwendung der Tabulator-taste und der Leertaste
- Emacs-Befehlstasten oder ähnliche Befehlsschlüsselfunktionen

Um diese Funktionen verwenden zu können, muss die `curses`-Bibliothek installiert sein. Wenn die `Curses`-Bibliothek nicht auf dem System installiert ist, verfügt `runmqsc` nicht über die Funktionen des Befehlszeileneditors und es wird eine Nachricht angezeigt, wenn die `runmqsc`-Befehlszeile gestartet wird. Der Name der zu installierenden `curses`-Bibliothek ist von der UNIX-Plattform abhängig:

-  Installieren Sie `curses` unter AIX.

- **HP-UX** Installieren Sie Xcurses unter HP-UX.
- Installieren Sie ncurses auf allen anderen UNIX -Plattformen und unter Linux.

Tastenbelegungen des Emacs anpassen

Sie können die Schlüssel anpassen, die an die Befehle gebunden sind. Sie können die Schlüssel beispielsweise an vi-Bindungen binden, anstatt die Standardbindungen des Emacs-Schlüssels zu verwenden.

Die Schlüssel werden angepasst, indem Sie die Datei `.editrc` bearbeiten, die im Ausgangsverzeichnis gespeichert ist. Weitere Informationen finden Sie unter [Editrc](#) auf den Seiten des FreeBSD-Mannes.

Befehlsschlüssel für den Befehl 'Abruf', 'Befehl' und 'Emacs' inaktivieren

Sie können den Befehl "recall", den Befehl "command completion" und den Befehl "Emacs" inaktivieren, indem Sie eine Umgebungsvariable festlegen. Setzen Sie die Umgebungsvariable **MQ_OVERRIDE_LIBEDIT_LOAD** auf TRUE.

Diese Umgebungsvariable kann als Fehlerumgehung verwendet werden, wenn **runmqsc** die folgende Informationsnachricht anzeigt:

```
AMQ8521I: Command completion and history unavailable
```

MQSC-Befehle aus Textdateien ausführen

Die Ausführung von MQSC-Befehlen ist interaktiv für Schnelltests geeignet. Wenn Sie jedoch über sehr lange Befehle verfügen oder eine bestimmte Folge von Befehlen wiederholt verwenden, sollten Sie die Umleitung von `stdin` aus einer Textdatei in Betracht ziehen.

Wenn Sie `stdin` aus einer Textdatei umleiten möchten, erstellen Sie zuerst mithilfe Ihres üblichen Texteditors eine Textdatei, die die MQSC-Befehle enthält, und führen Sie dann den Befehl **runmqsc** aus.

Anmerkung: Wenn Sie den Befehl **runmqsc** im Clientmodus ausführen, indem Sie `stdin` aus einer Textdatei umleiten, erwartet IBM MQ, dass die erste Zeile der Eingabedatei ein Kennwort ist.

Wenn Sie den Befehl **runmqsc** verwenden, verwenden Sie die Umleitungsoperatoren. Mit dem folgenden Befehl wird beispielsweise eine Folge von Befehlen ausgeführt, die in der Textdatei `myprog.in` enthalten sind:

```
runmqsc < myprog.in
```

In ähnlicher Weise können Sie die Ausgabe auch in eine Datei umleiten. Eine Datei, die die MQSC-Befehle für die Eingabe enthält, wird als MQSC-Befehlsdatei bezeichnet. Die Ausgabedatei, die Antworten aus dem Warteschlangenmanager enthält, wird als Ausgabedatei bezeichnet.

Wenn Sie `stdin` und `stdout` im Befehl **runmqsc** umleiten möchten, verwenden Sie folgende Form des Befehls:

```
runmqsc < myprog.in > myprog.out
```

Dieser Befehl ruft die MQSC-Befehle in der MQSC-Befehlsdatei auf `myprog.in`. Da kein Warteschlangenmanagername angegeben wurde, werden die MQSC-Befehle für den Standardwarteschlangenmanager ausgeführt. Die Ausgabe wird an die Textdatei `myprog.out` gesendet. [Abbildung 1 auf Seite 17](#) zeigt einen Auszug aus der MQSC-Befehlsdatei `myprog.in` und [Abbildung 2 auf Seite 18](#) zeigt den entsprechenden Auszug der Ausgabe in `myprog.out`.

Verwenden Sie zum Umleiten von `stdin` und `stdout` im Befehl **runmqsc** für einen Warteschlangenmanager (`saturn.queue.manager`), der nicht der Standardwert ist, das folgende Format:

```
runmqsc saturn.queue.manager < myprog.in > myprog.out
```

MQSC-Befehlsdateien

MQSC-Befehle werden in lesbarer Form geschrieben, d. A. in ASCII-Text. [Abbildung 1](#) auf [Seite 17](#) ist ein Auszug aus einer MQSC-Befehlsdatei, der einen MQSC-Befehl (**DEFINE QLOCAL**) mit seinen Attributen zeigt. [MQSC-Befehle](#) enthält eine Beschreibung der einzelnen MQSC-Befehle und ihre Syntax.

```
.  
. .  
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +  
DESCR(' ') +  
PUT(ENABLED) +  
DEFPRTY(0) +  
DEFPSIST(NO) +  
GET(ENABLED) +  
MAXDEPTH(5000) +  
MAXMSGL(1024) +  
DEFSOPT(SHARED) +  
NOHARDENBO +  
USAGE(NORMAL) +  
NOTRIGGER;  
. .  
.
```

Abbildung 1. Aus einer MQSC-Befehlsdatei extrahieren

Um Portierbarkeit zwischen IBM MQ-Umgebungen zu gewährleisten, müssen Sie die Zeilenlänge in MQSC-Befehlsdateien auf 72 Zeichen begrenzen. Das Pluszeichen gibt an, dass der Befehl in der nächsten Zeile fortgesetzt wird.

MQSC-Befehlsberichte

Der Befehl **runmqsc** gibt einen Bericht zurück, der an `stdout` gesendet wird. Der Bericht enthält:

- Ein Header, der MQSC-Befehle als Quelle für den Bericht identifiziert:

```
Starting MQSC for queue manager jupiter.queue.manager.
```

Dabei ist `jupiter.queue.manager` der Name des Warteschlangenmanagers.

- Eine optionale nummerierte Liste der ausgegebenen MQSC-Befehle. Standardmäßig wird der Text der Eingabe an die Ausgabe zurückgemeldet. In dieser Ausgabe steht vor jedem Befehl eine Folgenummer als Präfix (siehe [Abbildung 2](#) auf [Seite 18](#)). Sie können die Ausgabe jedoch durch Angabe des Flags **-e** im Befehl **runmqsc** unterdrücken.
- Es wurde eine Syntaxfehlermeldung für alle Befehle gefunden, die sich als fehlerhaft erwiesen haben.
- Eine Bedienermeldung mit dem Ergebnis jedes ausgeführten Befehls. Beispiel: Die Bedienermeldung für den erfolgreichen Abschluss eines **DEFINE QLOCAL** -Befehls lautet:

```
AMQ8006: IBM MQ queue created.
```

- Andere Nachrichten, die sich aus allgemeinen Fehlern beim Ausführen der Scriptdatei ergeben.

- Eine kurze statistische Zusammenfassung des Berichts, in der die Anzahl der gelesenen Befehle, die Anzahl der Befehle mit Syntaxfehlern und die Anzahl der Befehle angegeben sind, die nicht verarbeitet werden konnten.

Anmerkung: Der WS-Manager versucht, nur die Befehle zu verarbeiten, die keine Syntaxfehler aufweisen.

```
Starting MQSC for queue manager jupiter.queue.manager.
.
.
12:  DEFINE QLOCAL('ORANGE.LOCAL.QUEUE') REPLACE +
:    DESCR(' ') +
:    PUT(ENABLED) +
:    DEFPRTY(0) +
:    DEFPSIST(NO) +
:    GET(ENABLED) +
:    MAXDEPTH(5000) +
:    MAXMSGL(1024) +
:    DEFSOPT(SHARED) +
:    NOHARDENBO +
:    USAGE(NORMAL) +
:    NOTRIGGER;
AMQ8006: IBM MQ queue created.
.
.
.
```

Abbildung 2. Auszug aus einer MQSC-Befehlsberichtsdatei

Die bereitgestellten MQSC-Befehlsdateien ausführen

Die folgenden MQSC-Befehlsdateien werden mit IBM MQ bereitgestellt:

amqscos0.tst

Definitionen von Objekten, die von Beispielprogrammen verwendet werden.

amqscic0.tst

Definitionen von Warteschlangen für CICS-Transaktionen

Windows In IBM MQ for Windows befinden sich diese Dateien im Verzeichnis `MQ_INSTALLATION_PATH\tools\mqsc\samples`. `MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Linux **UNIX** Auf UNIX and Linux -Systemen befinden sich diese Dateien im Verzeichnis `MQ_INSTALLATION_PATH/samp`. `MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Der Befehl, der sie ausführt, lautet:

```
runmqsc < amqscos0.tst >test.out
```

Befehle 'runmqsc' zum Prüfen von Befehlen verwenden

Mit dem Befehl **runmqsc** können Sie MQSC-Befehle auf einem lokalen WS-Manager überprüfen, ohne sie tatsächlich auszuführen. Dazu setzen Sie das Flag **-v** im Befehl **runmqsc**, z. B.:

```
runmqsc -v < myprog.in > myprog.out
```

Wenn Sie **runmqsc** für eine MQSC-Befehlsdatei aufrufen, überprüft der Warteschlangenmanager jeden Befehl und gibt einen Bericht zurück, ohne die MQSC-Befehle tatsächlich auszuführen. Auf diese Weise

können Sie die Syntax der Befehle in Ihrer Befehlsdatei überprüfen. Dies ist besonders wichtig, wenn Sie die folgenden Schritte ausführen:

- Es wird eine große Anzahl Befehle aus einer Befehlsdatei ausgeführt.
- Die Verwendung einer MQSC-Befehlsdatei ist mehrfach zu überdauern.

Der zurückgegebene Bericht ähnelt dem in [Abbildung 2](#) auf Seite 18 dargestellten Bericht.

Sie können diese Methode nicht verwenden, um MQSC-Befehle über Remotezugriff zu überprüfen. Beispiel: Wenn Sie versuchen, diesen Befehl auszuführen:

```
runmqsc -w 30 -v jupiter.queue.manager < myprog.in > myprog.out
```

Das Flag **-w**, mit dem Sie angeben, dass es sich um einen fernen Warteschlangenmanager handelt, wird ignoriert und der Befehl wird lokal im Prüfmodus ausgeführt. 30 ist die Anzahl der Sekunden, die IBM MQ auf Antworten vom fernen Warteschlangenmanager wartet.

Zugehörige Konzepte

[„Standardeingabe und-ausgabe“](#) auf Seite 14

Die *Standardeingabeeinheit*, die auch als `stdin` bezeichnet wird, ist die Einheit, von der die Eingabe für das System ausgeführt wird. In der Regel ist dies die Tastatur, aber Sie können angeben, dass die Eingabe beispielsweise aus einem seriellen Anschluss oder einer Plattendatei stammen soll. Die *Standardausgabeeinheit*, die auch als `stdout` bezeichnet wird, ist die Einheit, an die die Ausgabe des Systems gesendet wird. Normalerweise ist dies eine Anzeige, aber Sie können die Ausgabe an einen seriellen Anschluss oder an eine Datei umleiten.

[„MQSC-Befehle interaktiv verwenden“](#) auf Seite 14

Sie können MQSC-Befehle interaktiv verwenden, indem Sie ein Befehlsfenster oder eine -Shell verwenden.

Zugehörige Informationen

[runmqsc](#)

MQSC-Befehle aus Stapeldateien ausführen

Wenn Sie über sehr lange Befehle verfügen oder eine bestimmte Folge von Befehlen wiederholt verwenden, sollten Sie die Umleitung von `stdin` aus einer Stapeldatei in Erwägung ziehen.

Um `stdin` aus einer Stapeldatei umzuleiten, müssen Sie zuerst eine Stapeldatei erstellen, die die MQSC-Befehle enthält, und verwenden Sie dabei Ihren üblichen Texteditor. Wenn Sie den Befehl `runmqsc` verwenden, verwenden Sie die Umleitungsoperatoren. Das folgende Beispiel:

1. Erstellt einen Testwarteschlangenmanager TESTQM
2. Erstellt eine übereinstimmende CLNTCONN-und Listener-Gruppe für die Verwendung von TCP/IP-Port 1600
3. Erstellt eine Testwarteschlange, TESTQ
4. Versetzt eine Nachricht in die Warteschlange mit Hilfe des Beispielprogramms 'amqspc'

```

export MYTEMPQM=TESTQM
export MYPOR=1600
export MQCHLLIB=/var/mqm/qmgrs/$MQTEMPQM/@ipcc

crtmqm $MYTEMPQM
stimqm $MYTEMPQM
runmqclsr -m $MYTEMPQM -t TCP -p $MYPOR &

runmqsc $MYTEMPQM << EOF
DEFINE CHANNEL(NTLM) CHLTYPE(SVRCONN) TRPTYPE(TCP)
DEFINE CHANNEL(NTLM) CHLTYPE(CLNTCONN) QMNAME('$MYTEMPQM') CONNAME('hostname($MYPOR)')
ALTER CHANNEL(NTLM) CHLTYPE(CLNTCONN)
DEFINE QLOCAL(TESTQ)
EOF

amqspucl TESTQ $MYTEMPQM << EOF
hello world
EOF

endmqm -i $MYTEMPQM

```

Abbildung 3. Beispielscript für die Ausführung von MQSC-Befehlen aus einer Stapeldatei

Probleme mit MQSC-Befehlen beheben

Wenn die Ausführung von MQSC-Befehlen nicht möglich ist, verwenden Sie die Informationen in diesem Thema, um festzustellen, ob eine dieser allgemeinen Probleme für Sie zutreffen. Es ist nicht immer klar, was das Problem ist, wenn Sie den Fehler lesen, den ein Befehl generiert.

Wenn Sie den Befehl `runmqsc` verwenden, müssen Sie Folgendes beachten:

- Mit dem Operator `<` können Sie die Eingabe aus einer Datei umleiten. Wenn Sie diesen Operator weglassen, interpretiert der WS-Manager den Dateinamen als WS-Managernamen und gibt die folgende Fehlernachricht aus:

```
AMQ8118E: IBM MQ queue manager does not exist.
```

- Wenn Sie die Ausgabe in eine Datei umleiten, verwenden Sie den Umleitungsoperator `>`. Standardmäßig wird die Datei zu dem Zeitpunkt im aktuellen Arbeitsverzeichnis gespeichert, an dem `runmqsc` aufgerufen wird. Geben Sie einen vollständig qualifizierten Dateinamen an, um die Ausgabe an eine bestimmte Datei und ein bestimmtes Verzeichnis zu senden.
- Überprüfen Sie, ob Sie den Warteschlangenmanager erstellt haben, der die Befehle ausführen soll, indem Sie den folgenden Befehl verwenden, um alle Warteschlangenmanager anzuzeigen:

```
dspmq
```

- Der WS-Manager muss aktiv sein. Wenn dies nicht der Fall ist, starten Sie es; (siehe [Warteschlangenmanager starten](#)). Sie erhalten eine Fehlernachricht, wenn Sie versuchen, einen Warteschlangenmanager zu starten, der bereits ausgeführt wird.
- Geben Sie im Befehl `runmqsc` einen WS-Manager-Namen an, wenn Sie keinen Standardwarteschlangenmanager definiert haben oder wenn Sie diesen Fehler erhalten:

```
AMQ8146E: IBM MQ queue manager not available.
```

- Sie können keinen MQSC-Befehl als Parameter für den Befehl `runmqsc` angeben. Dies ist z. B. nicht zulässig:

```
runmqsc DEFINE QLOCAL(FRED)
```

- Sie können keine MQSC-Befehle eingeben, bevor Sie den Befehl **runmqsc** ausgeben.

- Sie können über **runmqsc** keine Steuerbefehle ausführen. Beispielsweise können Sie den Befehl `strmqm` nicht zum Starten eines Warteschlangenmanagers ausgeben, während Sie die MQSC-Befehle interaktiv ausführen. Wenn Sie dies tun, erhalten Sie Fehlernachrichten ähnlich der folgenden:

```
runmqsc
.
.
Starting MQSC for queue manager jupiter.queue.manager.

1 : strmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of cmd segment below:-s

AMQ8426: Valid MQSC commands are:
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
2 : end
```

IBM MQ-Verwaltung mit PCF-Befehlen automatisieren

Sie können entscheiden, dass es für Ihre Installation von Vorteil wäre, einige Verwaltungs- und Überwachungstasks zu automatisieren. Sie können Verwaltungstasks sowohl für lokale als auch für ferne Warteschlangenmanager mit Hilfe von PCF-Befehlen (PCF = Programmable Command Format) automatisieren. In diesem Abschnitt wird vorausgesetzt, dass Sie Erfahrung mit der Verwaltung von IBM MQ-Objekten haben.

PCF-Befehle

Mithilfe von IBM MQ-PCF-Befehlen können Verwaltungstasks in einem Verwaltungsprogramm programmiert werden. Auf diese Weise können Sie von einem Programm aus Warteschlangenmanagerobjekte (Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsinformationsobjekte) bearbeiten und sogar die Warteschlangenmanager selbst bearbeiten.

PCF-Befehle decken den gleichen Funktionsumfang ab, der von MQSC-Befehlen bereitgestellt wird. Sie können ein Programm schreiben, um PCF-Befehle an jeden WS-Manager im Netz von einem einzelnen Knoten aus auszugeben. Auf diese Weise können Sie Verwaltungsaufgaben zentralisieren und automatisieren.

Jeder PCF-Befehl ist eine Datenstruktur, die in den Anwendungsdatenteil einer IBM MQ-Nachricht eingebettet ist. Jeder Befehl wird mit der MQI-Funktion `MQPUT` auf die gleiche Weise wie jede andere Nachricht an den Zielwarteschlangenmanager gesendet. Wenn der Befehlsserver auf dem Warteschlangenmanager ausgeführt wird, der die Nachricht empfängt, interpretiert der Befehlsserver die Nachricht als Befehlsnachricht und führt den Befehl aus. Um die Antworten abzurufen, gibt die Anwendung einen `MQGET` -Aufruf aus und die Antwortdaten werden in einer anderen Datenstruktur zurückgegeben. Die Anwendung kann dann die Antwort verarbeiten und entsprechend handeln.

Anmerkung: Im Gegensatz zu MQSC-Befehlen befinden sich PCF-Befehle und ihre Antworten nicht in einem Textformat, das Sie lesen können.

Kurz, dies sind einige der Dinge, die zum Erstellen einer PCF-Befehlsnachricht erforderlich sind:

Nachrichtendeskriptor

Dies ist ein Standardnachrichtendeskriptor von IBM MQ, in dem Folgendes gilt:

- Der Nachrichtentyp (*MsgType*) ist MQMT_REQUEST.
- Nachrichtenformat (*Format*) ist MQFMT_ADMIN.

Anwendungsdaten

Enthält die PCF-Nachricht (einschließlich des PCF-Headers), in der Folgendes gilt:

- Der PCF-Nachrichtentyp (*Type*) gibt MQCFT_COMMAND an.
- Die Befehls-ID gibt den Befehl an, z. B. *Change Queue* (MQCMD_CHANGE_Q).

Eine vollständige Beschreibung der PCF-Datenstrukturen und deren Implementierung finden Sie in [„Einführung in IBM MQ Programmable Command Formats“](#) auf Seite 22.

PCF-Objektattribute

Objektattribute in PCF sind nicht auf acht Zeichen beschränkt, da sie für MQSC-Befehle verwendet werden. Sie werden in diesem Handbuch in Kursivschrift angezeigt. Das PCF-Äquivalent von RQMNAME ist beispielsweise *RemoteQMGrName*.








Escape-PCFs

Escape-PCFs sind PCF-Befehle, die MQSC-Befehle innerhalb des Nachrichtentexts enthalten. Sie können PCFs verwenden, um Befehle an einen fernen Warteschlangenmanager zu senden. Weitere Informationen zu Escape-PCFs finden Sie unter [Escape](#).

Einführung in IBM MQ Programmable Command Formats

Programmierbare Befehlsformate (PCFs) definieren Befehls- und Antwortnachrichten, die zwischen einem Programm und einem beliebigen WS-Manager (der PCFs unterstützt) in einem Netz ausgetauscht werden können. PCFs vereinfachen die Verwaltung von Warteschlangenmanagern und andere Netzadministrationsysteme. Sie können eingesetzt werden, um das Problem der komplexen Verwaltung verteilter Netzwerke zu lösen, insbesondere, wenn Netzwerke in Größe und Komplexität wachsen.

Programmierbare Befehlsformate werden unterstützt von:

-  IBM MQ for AIX
-  IBM MQ for HP-UX
-  IBM MQ for IBM i
-  IBM MQ for Linux
-  IBM MQ for Solaris
-  IBM MQ for Windows
-  IBM MQ for z/OS

Probleme mit PCF-Befehlen lösen

Die Verwaltung verteilter Netze kann komplex werden. Die Probleme der Administration wachsen weiter, da die Netzwerke in Größe und Komplexität zunehmen.

Beispiele für die Verwaltung von Messaging und Warteschlangensteuerung sind:

- Ressourcenmanagement.
Zum Beispiel die Erstellung und Löschung von Warteschlangen.
- Leistungsüberwachung.
Beispiel: Maximale Warteschlangenlänge oder Nachrichtenrate.
- Steuerung.

Beispiel für die Optimierung von Warteschlangenparametern wie der maximalen Warteschlangenlänge, der maximalen Nachrichtenlänge und der Aktivierung und Inaktivierung von Warteschlangen.

- Nachrichtenweiterleitung.

Definition von alternativen Routen durch ein Netzwerk.

IBM MQ-PCF-Befehle können verwendet werden, um die Verwaltung von Warteschlangenmanagern und andere Netzadministrationsbefehle zu vereinfachen. Mithilfe von PCF-Befehlen können Sie eine einzelne Anwendung verwenden, um die Netzverwaltung über einen einzelnen Warteschlangenmanager im Netz auszuführen.

Was sind PCFs?

PCFs definieren Befehls- und Antwortnachrichten, die zwischen einem Programm und einem beliebigen Warteschlangenmanager (der PCFs unterstützt) in einem Netz ausgetauscht werden können. Sie können PCF-Befehle in einem Systemverwaltungsanwendungsprogramm für die Verwaltung von IBM MQ-Objekten verwenden: Authentifizierungsinformationsobjekte, Kanäle, Kanallistener, Namenslisten, Prozessdefinitionen, Warteschlangenmanager, Warteschlangen, Services und Speicherklassen. Die Anwendung kann von einem einzigen Punkt im Netz aus ausgeführt werden, um die Befehls- und Antwortinformationen mit jedem WS-Manager, lokal oder fern mit dem lokalen WS-Manager zu kommunizieren.


Jeder Warteschlangenmanager verfügt über eine Verwaltungswarteschlange mit einem Standardwarteschlangennamen, und Ihre Anwendung kann PCF-Befehlsnachrichten an diese Warteschlange senden. Jeder Warteschlangenmanager verfügt außerdem über einen Befehlsserver, um die Befehlsnachrichten aus der Verwaltungswarteschlange zu bedienen. PCF-Befehlsnachrichten können daher von jedem Warteschlangenmanager im Netz verarbeitet werden, und die Antwortdaten können mit Hilfe der angegebenen Antwortwarteschlange an Ihre Anwendung zurückgegeben werden. PCF-Befehle und Antwortnachrichten werden über die normale Nachrichtenwarteschlangenschnittstelle (MQI) gesendet und empfangen.

Eine Liste der verfügbaren PCF-Befehle, einschließlich ihrer Parameter, finden Sie im Abschnitt [Definitionen der programmierbaren Befehlsformate](#).

IBM MQ Programmable Command Formats verwenden

Sie können PCFs in einem Systemverwaltungsprogramm für die IBM MQ-Fernverwaltung verwenden.

Dieser Abschnitt enthält:

- „PCF-Befehlsnachrichten“ auf Seite 23
- „PCF-Antworten in IBM MQ“ auf Seite 26
-  „Erweiterte Antworten“ auf Seite 28
- Regeln für die Benennung von IBM MQ -Objekten
- „Berechtigungsprüfung für PCF-Befehle in IBM MQ“ auf Seite 30

PCF-Befehlsnachrichten

PCF-Befehlsnachrichten bestehen aus einem PCF-Header, den in diesem Header angegebenen Parametern und auch benutzerdefinierte Nachrichtendaten. Die Nachrichten werden unter Verwendung von Nachrichtenwarteschlangenschnittstellenaufrufen ausgegeben.

Jeder Befehl und seine Parameter werden als separate Befehlsnachricht gesendet, die einen PCF-Header gefolgt von einer Reihe von Parameterstrukturen enthält. Einzelheiten zum PCF-Header finden Sie im Abschnitt [MQCFH-PCF-Header](#) und in einem Beispiel für eine Parameterstruktur. Weitere Informationen finden Sie im Abschnitt [MQCFST-PCF-Zeichenfolgeparameter](#). Der PCF-Header gibt den Befehl und die Anzahl der Parameterstrukturen an, die in derselben Nachricht folgen. Jede Parameterstruktur stellt einen Parameter für den Befehl bereit.

Antworten auf die Befehle, die vom Befehlsserver generiert werden, haben eine ähnliche Struktur. Es gibt einen PCF-Header, gefolgt von einer Reihe von Parameterstrukturen. Antworten können aus mehr als einer Nachricht bestehen, aber Befehle bestehen immer nur aus einer Nachricht.

Multi Unter Multiplatforms wird die Warteschlange, an die die PCF-Befehle gesendet werden, immer als SYSTEM.ADMIN.COMMAND.QUEUE bezeichnet.

z/OS Unter z/OS werden Befehle an die Warteschlange SYSTEM.COMMAND.INPUT gesendet, obwohl SYSTEM.ADMIN.COMMAND.QUEUE ein Aliasname für sie sein kann. Der Befehlsserver, der diese Warteschlange bedient, sendet die Antworten an die Warteschlange, die durch die Felder *ReplyToQ* und *ReplyToQMgr* im Nachrichtendeskriptor der Befehlsnachricht definiert ist.

Informationen zum Absetzen von PCF-Befehlsnachrichten

Verwenden Sie die normalen MQI-Aufrufe (MQI = Message Queue Interface), MQPUT, MQGET usw. zum Einreihen und Abrufen von PCF-Befehls- und Antwortnachrichten in und aus ihren Warteschlangen.

Anmerkung:

Stellen Sie sicher, dass der Befehlsserver auf dem Zielwarteschlangenmanager ausgeführt wird, damit der PCF-Befehl in diesem WS-Manager verarbeitet wird.

Eine Liste der bereitgestellten Headerdateien finden Sie in IBM MQ COPY-, Header, Include- und Moduldateien.

Nachrichtendeskriptor für einen PCF-Befehl

Der IBM MQ-Nachrichtendeskriptor ist in MQMD-Nachrichtendeskriptor vollständig dokumentiert.

Eine PCF-Befehlsnachricht enthält die folgenden Felder im Nachrichtendeskriptor:

Bericht

Jeder gültige Wert, wie erforderlich.

MsgType

Dieses Feld muss MQMT_REQUEST sein, um eine Nachricht anzugeben, die eine Antwort erfordert.

Verfall

Jeder gültige Wert, wie erforderlich.

Feedback

Auf 'MQFB_NONE' gesetzt

Encoding

Wenn Sie an eines der folgenden Systeme senden, legen Sie dieses Feld auf die Codierung fest, die für die Nachrichtendaten verwendet wird. Bei Bedarf wird die Konvertierung durchgeführt:

- **IBM i** IBM i
- **Linux** Linux
- **UNIX** UNIX
- **Windows** Windows

CodedCharSetId

Wenn Sie an eines der folgenden Systeme senden, setzen Sie dieses Feld auf die ID des codierten Zeichensatzes, der für die Nachrichtendaten verwendet wird. Falls erforderlich, wird die Konvertierung durchgeführt:

- **IBM i** IBM i
- **Linux** Linux
- **UNIX** UNIX
- **Windows** Windows

Format

Setzen Sie sie auf MQFMT_ADMIN.

Priority

Jeder gültige Wert, wie erforderlich.

Permanenz

Jeder gültige Wert, wie erforderlich.

MsgId

Die sendende Anwendung kann jeden beliebigen Wert angeben, oder MQMI_NONE kann angegeben werden, um den Warteschlangenmanager anzufordern, eine eindeutige Nachrichten-ID zu generieren.

CorrelId

Die sendende Anwendung kann einen beliebigen Wert angeben, oder MQCI_NONE kann angegeben werden, um keine Korrelations-ID anzugeben.

ReplyToQ

Der Name der Warteschlange, die die Antwort empfangen soll.

ReplyToQMgr

Der Name des Warteschlangenmanagers für die Antwort (oder leer).

Nachrichtenkontextfelder

Diese Felder können, wie erforderlich, auf alle gültigen Werte gesetzt werden. In der Regel wird die Put-Nachrichtoption MQPMO_DEFAULT_CONTEXT verwendet, um die Nachrichtenkontextfelder auf die Standardwerte zu setzen.

Wenn Sie eine MQMD-Struktur der Version 2 verwenden, müssen Sie die folgenden zusätzlichen Felder festlegen:

GroupId

Auf MQGI_NONE gesetzt

MsgSeqNumber

Auf 1 setzen

Offset

Auf 0 setzen

MsgFlags

Auf MQMF_NONE gesetzt

OriginalLength

Auf MQOL_UNDEFINED setzen

Benutzerdaten senden

Die PCF-Strukturen können auch zum Senden benutzerdefinierter Nachrichtendaten verwendet werden. In diesem Fall muss das Feld für den Nachrichtendeskriptor *Format* auf MQFMT_PCF gesetzt werden.

PCF-Nachrichten in einer angegebenen Warteschlange senden und empfangen**PCF-Nachrichten an eine angegebene Warteschlange senden**

Um eine Nachricht an eine angegebene Warteschlange zu senden, konvertiert der mqPutBag-Aufruf den Inhalt des angegebenen Sacks in eine PCF-Nachricht und sendet die Nachricht an die angegebene Warteschlange. Die Inhalte des Behälters sind nach dem Aufruf unverändert.

Als Eingabe für diesen Aufruf müssen Sie Folgendes angeben:

- Eine MQI-Verbindungskennung.
- Eine Objektkennung für die Warteschlange, in die die Nachricht gestellt werden soll.
- Ein Nachrichtendeskriptor. Weitere Informationen zum Nachrichtendeskriptor finden Sie im Abschnitt [MQMD-Nachrichtendeskriptor](#) .

- Einreihen von Nachrichtenoptionen mit Hilfe der MQPMO-Struktur. Weitere Informationen zur MQPMO-Struktur finden Sie unter [MQPMO-Put-message-Optionen](#).
- Der Handle der Tasche, die in eine Nachricht umgewandelt werden soll.

Anmerkung: Wenn die Tasche eine Verwaltungsnachricht enthält und der Aufruf 'mqAddInquiry' zum Einfügen von Werten in die Tasche verwendet wurde, muss der Wert des Datenelements 'MQIASY_COM-MAND' ein Befehl INQUIRE sein, der von der MQAI erkannt wird.

Eine vollständige Beschreibung des mqPutBag-Aufrufs finden Sie unter [mqPutBag](#).

PCF-Nachrichten aus einer angegebenen Warteschlange empfangen

Um eine Nachricht aus einer angegebenen Warteschlange zu empfangen, ruft der mqGetBag-Aufruf eine PCF-Nachricht aus einer angegebenen Warteschlange ab und konvertiert die Nachrichtendaten in einen Datenbehälter.

Als Eingabe für diesen Aufruf müssen Sie Folgendes angeben:

- Eine MQI-Verbindungskennung.
- Eine Objektkennung der Warteschlange, aus der die Nachricht gelesen werden soll.
- Ein Nachrichtendeskriptor. Innerhalb der MQMD-Struktur muss der Parameter **Format** MQFMT_ADMIN, MQFMT_EVENT oder MQFMT_PCF sein.

Anmerkung: Wenn die Nachricht innerhalb einer UOWs (d. a. mit der Option MQGMO_SYNCPOINT) empfangen wird und die Nachricht ein nicht unterstütztes Format hat, kann die Arbeitseinheit zurückgesetzt werden. Die Nachricht wird dann in der Warteschlange wieder in die Warteschlange gestellt und kann mit dem MQGET-Aufruf anstelle des mqGetBag-Aufrufs abgerufen werden. Weitere Informationen zum Nachrichtendeskriptor finden Sie unter [MQGMO-Get-message options](#).

- Abrufen von Nachrichtenoptionen unter Verwendung der MQGMO-Struktur. Weitere Informationen zur MQGMO-Struktur finden Sie im Abschnitt [MQMD-Nachrichtendeskriptor](#).
- Der Handle der Tasche, die die konvertierte Nachricht enthalten soll.

Eine vollständige Beschreibung des mqGetBag-Aufrufs finden Sie in [mqGetBag](#).

PCF-Antworten in IBM MQ

Als Antwort auf jeden Befehl generiert der Befehlsserver eine oder mehrere Antwortnachrichten. Eine Antwortnachricht hat ein ähnliches Format wie eine Befehlsnachricht.

Der PCF-Header hat denselben Befehlskennungswert wie der Befehl, auf den es sich um eine Antwort handelt (Details hierzu finden Sie im Abschnitt [MQCFH-PCF-Header](#)). Die Nachrichten-ID und die Korrelations-ID werden entsprechend den Berichtsoptionen der Anforderung festgelegt.

Wenn der PCF-Header-Typ der Befehlsnachricht MQCFT_COMMAND ist, werden nur Standardantworten generiert. Solche Befehle werden auf allen Plattformen außer z/OS unterstützt. Ältere Anwendungen unterstützen PCF unter z/OS nicht. IBM MQ Explorer unter Windows ist eine solche Anwendung (IBM WebSphere MQ 6.0 oder höher IBM MQ Explorer unterstützt jedoch PCF unter z/OS).

Wenn der PCF-Header-Typ der Befehlsnachricht MQCFT_COMMAND_XR lautet, werden entweder erweiterte oder Standardantworten generiert. Solche Befehle werden auf z/OS- und einigen anderen Plattformen unterstützt. Befehle, die unter z/OS ausgegeben werden, generieren nur erweiterte Antworten. Auf anderen Plattformen kann entweder der Typ der Antwort generiert werden.

Wenn ein einzelner Befehl einen generischen Objektnamen angibt, wird eine separate Antwort in einer eigenen Nachricht für jedes übereinstimmende Objekt zurückgegeben. Bei der Antwortgenerierung wird ein einzelner Befehl mit einem generischen Namen als mehrere Einzelbefehle behandelt (außer dem Steuerfeld MQCFC_LAST oder MQCFC_NOT_LAST). Andernfalls generiert eine Befehlsnachricht eine Antwortnachricht.

Bestimmte PCF-Antworten geben möglicherweise eine Struktur zurück, selbst wenn sie nicht angefordert wird. Diese Struktur wird in der Definition der Antwort ([Definitionen der programmierbaren Befehlsforma-](#)

te) als *immer zurückgegeben* angezeigt. Der Grund, warum es für diese Antworten erforderlich ist, die Objekte in der Antwort zu benennen, um zu identifizieren, für welches Objekt die Daten gelten.

Nachrichtendeskriptor für eine Antwort

Eine Antwortnachricht enthält die folgenden Felder im Nachrichtendeskriptor:

MsgType

Dieses Feld ist MQMT_REPLY.

MsgId

Dieses Feld wird vom WS-Manager generiert.

CorrelId

Dieses Feld wird in Übereinstimmung mit den Berichtsoptionen der Befehlsnachricht generiert.

Format

Dieses Feld ist MQFMT_ADMIN.

Encoding

Setzen Sie diese Option auf MQENC_NATIVE.

CodedCharSetId

Setzen Sie diese Option auf MQCCSI_Q_MGR.

Permanenz

Das gleiche wie in der Befehlsnachricht.

Priority

Das gleiche wie in der Befehlsnachricht.

Die Antwort wird mit MQPMO_PASS_IDENTITY_CONTEXT generiert.

Standardantworten

Befehlsnachrichten mit dem Headertyp MQCFT_COMMAND, Standardantworten werden generiert. Solche Befehle werden auf allen Plattformen außer z/OS unterstützt.

Es gibt drei Typen von Standardantworten:

- OK-Antwort
- Fehlerantwort
- Datenantwort

OK-Antwort

Diese Antwort besteht aus einer Nachricht, die mit einem Befehlsformat-Header beginnt und ein *CompCode* -Feld von MQCC_OK oder MQCC_WARNING enthält.

Für MQCC_OK lautet der *Reason* MQRC_NONE.

Für MQCC_WARNING identifiziert der *Reason* die Art der Warnung. In diesem Fall kann dem Befehlsformat-Header ein oder mehrere Warnparameterstrukturen folgen, die diesem Ursachencode angemessen sind.

In beiden Fällen können weitere Parameterstrukturen für einen inquire-Befehl wie in den folgenden Abschnitten beschrieben ausgeführt werden.

Fehlerantwort

Wenn der Befehl einen Fehler aufweist, werden eine oder mehrere Fehlerantworten gesendet (mehr als eine Nachricht kann auch für einen Befehl gesendet werden, der normalerweise nur eine einzige Antwortnachricht hätte). Diese Fehlerantworten verfügen über MQCFC_LAST oder MQCFC_NOT_LAST, die entsprechend gesetzt sind.

Jede solche Nachricht beginnt mit einem Antwortformatheader mit einem *CompCode* -Wert von MQCC_FAILED und einem Feld *Reason*, in dem der jeweilige Fehler identifiziert wird. Im Allgemeinen

wird in jeder Nachricht ein anderer Fehler beschrieben. Darüber hinaus weist jede Nachricht entweder null oder eine (nie mehr als eine) Fehlerparameterstruktur nach dem Header auf. Diese Parameterstruktur, sofern vorhanden, ist eine MQCFIN-Struktur, wobei ein *Parameter*-Feld eine der folgenden Werte enthält:

- MQIACF_PARAMETER_ID

Das Feld *Value* in der Struktur ist die Parameter-ID des Fehlers, der fehlerhaft war (z. B. MQCA_Q_NAME).

- MQIACF_ERROR_ID

Dieser Wert wird zusammen mit einem *Reason*-Wert (im Befehlsformat-Header) von MQRC_UNEXPECTED_ERROR verwendet. Das Feld *Value* in der MQCFIN-Struktur ist der unerwartete Ursachencode, der vom Befehlsserver empfangen wurde.

- MQIACF_SELECTOR

Dieser Wert tritt auf, wenn eine mit dem Befehl gesendete Listenstruktur (MQCFIL) einen doppelten Selektor enthält oder eine nicht gültige Auswahl ist. Das Feld *Reason* im Befehlsformat-Header gibt den Fehler an, und das Feld *Value* in der MQCFIN-Struktur ist der Parameterwert in der MQCFIL-Struktur des fehlerhaft bestellten Befehls.

- MQIACF_ERROR_OFFSET

Dieser Wert tritt auf, wenn ein Datenvergleichsfehler im Befehl Ping Channel vorliegt. Das Feld *Value* in der Struktur ist der Offset des Ping-Channel-Vergleichsfehlers.

- MQIA_CODED_CHAR_SET_ID

Dieser Wert tritt auf, wenn die ID des codierten Zeichensatzes im Nachrichtendeskriptor der eingehenden PCF-Befehlsnachricht nicht mit der ID des Zielwarteschlangenmanagers übereinstimmt. Das Feld *Value* in der Struktur ist die ID des codierten Zeichensatzes des Warteschlangenmanagers.

Die letzte (oder einzige) Fehlerantwortnachricht ist eine Zusammenfassungsantwort mit einem *CompCode*-Feld von MQCC_FAILED und einem *Reason*-Feld von MQRCCF_COMMAND_FAILED. Diese Nachricht weist keine Parameterstruktur nach der Kopfzeile auf.

Datenantwort

Diese Antwort setzt sich aus einer OK-Antwort (wie zuvor beschrieben) in einen Befehl inquire zusammen. Auf die OK-Antwort folgen weitere Strukturen, die die angeforderten Daten enthalten, wie in [Definitionen der programmierbaren Befehlsformate](#) beschrieben.

Anwendungen dürfen nicht davon abhängig sein, dass diese zusätzlichen Parameterstrukturen in einer bestimmten Reihenfolge zurückgegeben werden.

Erweiterte Antworten

Befehle, die unter z/OS ausgegeben werden, generieren erweiterte Antworten.

Es gibt drei Typen von erweiterten Antworten:

- Nachrichtenantwort, mit Typ MQCFT_XR_MSG
- Elementantwort, mit Typ MQCFT_XR_ITEM
- Zusammenfassungsantwort mit Typ MQCFT_XR_SUMMARY

Jeder Befehl kann eine oder mehrere Gruppen von Antworten generieren. Jede Gruppe von Antworten besteht aus einer oder mehreren Nachrichten, die sequenziell von 1 im Feld *MsgSeqNumber* des PCF-Headers nummeriert sind. Das Feld *Control* der letzten (oder einzigen) Antwort in jeder Gruppe hat den Wert MQCFC_LAST. Für alle anderen Antworten in der Gruppe lautet dieser Wert 'MQCFC_NOT_LAST'.

Jede Antwort kann eine oder mehrere optionale MQCFBS-Strukturen enthalten, in denen das Feld *Parameter* auf den Wert MQBACF_RESPONSE_SET gesetzt ist, wobei der Wert eine Antwortsetkennung ist. Kennungen sind eindeutig und identifizieren die Gruppe von Antworten, die die Antwort enthalten. Für jede Gruppe von Antworten gibt es eine MQCFBS-Struktur, die sie identifiziert.

Die erweiterten Antworten verfügen über mindestens zwei Parameterstrukturen:

- Eine MQCFBS-Struktur mit dem Feld *Parameter*, die auf MQBACF_RESPONSE_ID gesetzt ist. Der Wert in diesem Feld ist die ID der Gruppe von Antworten, zu der die Antwort gehört. Die Kennung in der ersten Gruppe ist beliebig. In nachfolgenden Gruppen ist die ID eine zuvor in einer MQBACF_RESPONSE_SET-Struktur notifizierte ID.
- Eine MQCFST-Struktur, bei der das Feld *Parameter* auf MQCACF_RESPONSE_Q_MGR_NAME gesetzt ist, wobei der Wert der Name des Warteschlangenmanagers ist, aus dem die Gruppe der Antworten stammt.

Viele Antworten verfügen über zusätzliche Parameterstrukturen, und diese Strukturen werden in den folgenden Abschnitten beschrieben.

Sie können im Voraus nicht feststellen, wie viele Antworten in einer anderen Gruppe als Antworten vorhanden sind, bis eine Antwort gefunden wird, bis eine mit MQCFC_LAST gefunden wird. Sie können auch nicht im Voraus feststellen, wie viele Gruppen von Antworten in einer beliebigen Menge MQBACF_RESPONSE_SET-Strukturen enthalten, um anzuzeigen, dass zusätzliche Sätze generiert werden.

Erweiterte Antworten auf Inquire-Befehle

Die Befehle 'Inquire' generieren normalerweise eine Elementantwort (Typ MQCFT_XR_ITEM) für jedes Element, das gefunden wird, das den angegebenen Suchkriterien entspricht. Die Elementantwort weist ein *CompCode* -Feld in der Kopfzeile mit dem Wert MQCC_OK und ein Feld *Reason* mit dem Wert MQRC_NONE auf. Es enthält auch andere Parameterstrukturen, die das Element und dessen angeforderte Attribute beschreiben, wie im Abschnitt [Definitionen der programmierbaren Befehlsformate](#) beschrieben.

Wenn ein Element fehlerhaft ist, weist das Feld *CompCode* im Header einen Wert von MQCC_FAILED auf und das Feld *Reason* gibt den jeweiligen Fehler an. Es werden zusätzliche Parameterstrukturen angezeigt, um das Element zu identifizieren.

Bestimmte Inquire-Befehle geben zusätzlich zu den Elementantworten allgemeine (nicht namenspezifische) Nachrichtenantworten zurück. Bei diesen Antworten handelt es sich um Informations- oder Fehlermeldungen, die Antworten des Typs MQCFT_XR_MSG enthalten.

Wenn der Befehl 'Inquire' erfolgreich ausgeführt wird, kann es optional eine Zusammenfassungsantwort geben (Typ MQCFT_XR_SUMMARY), mit einem *CompCode* -Wert von MQCC_OK und einem *Reason* -Feldwert von MQRC_NONE.

Wenn der Befehl 'Inquire' fehlschlägt, werden möglicherweise Elementantworten zurückgegeben, und es kann optional eine Zusammenfassungsantwort (Typ MQCFT_XR_SUMMARY) mit einem *CompCode* -Wert von MQCC_FAILED und einem *Reason* -Feldwert von MQRCCF_COMMAND_FAILED bestehen.

Erweiterte Antworten auf andere Befehle als 'Inquire'

Erfolgreiche Befehle generieren Nachrichtenantworten, in denen das Feld *CompCode* im Header einen Wert von MQCC_OK hat, und das Feld *Reason* hat den Wert MQRC_NONE. Es gibt immer mindestens eine Nachricht. Es kann sich um eine Informationsnachricht (MQCFT_XR_MSG) oder eine Zusammenfassung (MQCFT_XR_SUMMARY) handeln. Es können optional zusätzliche Informationsnachrichten (Typ MQCFT_XR_MSG) vorhanden sein. Jede Informationsnachricht enthält möglicherweise eine Reihe zusätzlicher Parameterstrukturen mit Informationen zu dem Befehl; die einzelnen Befehlsbeschreibungen für die Strukturen, die auftreten können, finden Sie in den einzelnen Befehlsbeschreibungen.

Befehle, die fehlschlagen, generieren Fehlermeldungenantworten (Typ MQCFT_XR_MSG), in denen das Feld *CompCode* im Header den Wert MQCC_FAILED hat und das Feld *Reason* den jeweiligen Fehler identifiziert. Jede Nachricht enthält möglicherweise eine Reihe zusätzlicher Parameterstrukturen mit Informationen zu dem Fehler: Die einzelnen Fehlerbeschreibungen für die Strukturen, die auftreten können, finden Sie in den einzelnen Fehlerbeschreibungen. Es können Informationen zu Informationsnachrichten generiert werden. Es kann optional eine Zusammenfassungsantwort (MQCFT_XR_SUMMARY) mit einem *CompCode* -Wert von MQCC_FAILED und einem *Reason* -Feldwert von MQRCCF_COMMAND_FAILED vorhanden sein.

Erweiterte Antworten auf Befehle mit CommandScope

Wenn ein Befehl den Parameter **CommandScope** verwendet oder einen Befehl mit dem Parameter **CommandScope** generiert, der generiert werden soll, wird eine erste Antwort vom Warteschlangenmanager des Befehls empfangen, in dem der Befehl empfangen wurde. Anschließend wird für jeden Warteschlangenmanager, an den der Befehl gerichtet ist (als ob mehrere Einzelbefehle abgesetzt wurden) ein separater Satz oder eine separate Gruppe von Antworten generiert. Schließlich gibt es eine Antwort vom Empfangswarteschlangenmanager, die eine Gesamtantwortantwort (Typ MQCFT_XR_SUMMARY) enthält. Die Parameterstruktur MQCACF_RESPONSE_Q_Q_MGR_NAME gibt den Warteschlangenmanager an, der die einzelnen Gruppen generiert.


Die Anfangsantwortgruppe verfügt über die folgenden zusätzlichen Parameterstrukturen:

- MQIACF_COMMAND_INFO (MQCFIN). Mögliche Werte in dieser Struktur sind 'MQCMDI_CMDScope_ACCEPTED' oder 'MQCMDI_CMDScope_GENERATED'.
- MQIACF_CMDScope_Q_MGR_COUNT (MQCFIN). Diese Struktur gibt die Anzahl der WS-Manager an, an die der Befehl gesendet wird.

Berechtigungsprüfung für PCF-Befehle in IBM MQ

Bei der Verarbeitung eines PCF-Befehls wird der *UserIdentifier* aus dem Nachrichtendeskriptor in der Befehlsnachricht für die erforderlichen IBM MQ-Objektberechtigungsprüfungen verwendet. Die Berechtigungsprüfung wird auf jeder Plattform, wie in diesem Thema beschrieben, unterschiedlich implementiert.

Die Prüfungen werden auf dem System ausgeführt, auf dem der Befehl verarbeitet wird. Daher muss diese Benutzer-ID auf dem Zielsystem vorhanden sein und die erforderlichen Berechtigungen zum Verarbeiten des Befehls haben. Wenn die Nachricht von einem fernen System stammt, ist eine Möglichkeit, die auf dem Zielsystem vorhandene ID zu erreichen, eine übereinstimmende Benutzer-ID auf dem lokalen und dem fernen System zu haben.

Anmerkung:  Informationen zur Berechtigungsprüfung für z/OS finden Sie in [Task 1: z/OS-Systemparameter identifizieren](#).

IBM MQ for IBM i



Um einen PCF-Befehl verarbeiten zu können, muss die Benutzer-ID die Berechtigung *dsp* für das IBM MQ-Objekt auf dem Zielsystem besitzen.

Darüber hinaus werden IBM MQ-Objektberechtigungsprüfungen für bestimmte PCF-Befehle ausgeführt, wie in [Tabelle 1](#) auf Seite 31 dargestellt.

In den meisten Fällen handelt es sich bei diesen Prüfungen um dieselben Prüfungen, die von den entsprechenden Befehlen der IBM MQ-Steuersprache, die auf einem lokalen System abgesetzt wurden, durchgeführt werden. Weitere Informationen zur Zuordnung von IBM MQ -Berechtigungen zu IBM i -Systemberechtigungen und zu den Berechtigungsanforderungen für die CL-Befehle von IBM MQ finden Sie im Abschnitt [Sicherheit unter IBM i einrichten](#). Details zur Sicherheit in Bezug auf Exits werden in der Dokumentation [Sicherheit auf Verbindungsebene mit Hilfe einer Sicherheitsexit](#) angegeben.

Gehen Sie wie folgt vor, um einen der folgenden Befehle zu verarbeiten : Die Benutzer-ID muss Mitglied des Gruppenprofils QMQMADM sein:

- Pingkanal
- Kanal ändern
- Kanal kopieren
- Kanal erstellen
- Kanal löschen
- Kanal zurücksetzen
- Kanal auflösen

- Kanal starten
- Kanal stoppen
- Kanalinitiator starten
- Kanal-Listener starten

IBM MQ for UNIX, Linux, and Windows



Um einen PCF-Befehl verarbeiten zu können, muss die Benutzer-ID über die *dsp* -Berechtigung für das WS-Manager-Objekt auf dem Zielsystem verfügen. Darüber hinaus werden IBM MQ-Objektberechtigungsprüfungen für bestimmte PCF-Befehle ausgeführt, wie in [Tabelle 1](#) auf Seite 31 dargestellt.

Wenn Sie einen der folgenden Befehle verarbeiten möchten, muss die Benutzer-ID zur Gruppe *mqm* gehören.

Anmerkung: Nur für Windows gilt, dass die Benutzer-ID zur Gruppe *Administratoren* oder zur Gruppe *mqm* gehören kann.

- Kanal ändern
- Kanal kopieren
- Kanal erstellen
- Kanal löschen
- Pingkanal
- Kanal zurücksetzen
- Kanal starten
- Kanal stoppen
- Kanalinitiator starten
- Kanal-Listener starten
- Kanal auflösen
- Cluster zurücksetzen
- Cluster aktualisieren
- WS-Manager aussetzen
- WS-Manager wiederaufnehmen

IBM MQ-Objektberechtigungen für Multiplatforms



Tabelle 1. Objektberechtigungen		
Befehl	IBM MQ-Objektberechtigung	Klassenberechtigung (für Objekttyp)
Authentifizierungsinformationen ändern	dsp und chg	Nicht zutreffend
Kanal ändern	dsp und chg	Nicht zutreffend
Channel-Listener ändern	dsp und chg	Nicht zutreffend
Clientverbindungskanal ändern	dsp und chg	Nicht zutreffend
Namensliste ändern	dsp und chg	Nicht zutreffend
Prozess ändern	dsp und chg	Nicht zutreffend

<i>Tabelle 1. Objektberechtigungen (Forts.)</i>		
Befehl	IBM MQ-Objektberechtigung	Klassenberechtigung (für Objekttyp)
Warteschlange ändern	dsp und chg	Nicht zutreffend
Warteschlangenmanager ändern	chg <i>siehe Anmerkung 3 und Anmerkung 5</i>	Nicht zutreffend
Change Service	dsp und chg	Nicht zutreffend
Warteschlange löschen	clr	Nicht zutreffend
Authentifizierungsinformationen kopieren	DSP	crt
Authentifizierungsinformationen kopieren (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: chg</i>	crt
Kanal kopieren	DSP	crt
Copy Channel (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: chg</i>	crt
Channel-Listener kopieren	DSP	crt
Copy Channel Listener (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: chg</i>	crt
Clientverbindungskanal kopieren	DSP	crt
Clientverbindungskanal kopieren (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: chg</i>	crt
Namensliste kopieren	DSP	crt
Namensliste kopieren (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: dsp und chg</i>	crt
Prozess kopieren	DSP	crt
Prozess kopieren (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: chg</i>	crt
Warteschlange kopieren	DSP	crt
Warteschlange kopieren (Ersetzen) <i>siehe Anmerkung 1</i>	<i>from: dsp to: dsp und chg</i>	crt
Authentifizierungsinformationen erstellen	<i>(Systemstandardauthentifizierungsinformationen) dsp</i>	crt
Authentifizierungsinformationen erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	<i>(Systemstandardauthentifizierungsinformationen) dsp to: chg</i>	crt
Kanal erstellen	<i>(Systemstandardkanal) dsp</i>	crt
Kanal erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	<i>(Systemstandardkanal) dsp to: chg</i>	crt
Kanallistener erstellen	<i>(Systemstandardlistener) dsp</i>	crt
Create Channel Listener (Ersetzen) <i>siehe Anmerkung 1</i>	<i>(system default listener) dsp to: chg</i>	crt
Clientverbindungskanal erstellen	<i>(Systemstandardkanal) dsp</i>	crt

<i>Tabelle 1. Objektberechtigungen (Forts.)</i>		
Befehl	IBM MQ-Objektberechtigung	Klassenberechtigung (für Objekttyp)
Clientverbindungskanal erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	(Systemstandardkanal) dsp to: chg	crt
Namensliste erstellen	(Systemstandardnamensliste) dsp	crt
Create Namelist (Ersetzen) <i>siehe Anmerkung 1</i>	(system default namelist) dsp to: dsp und chg	crt
Prozess erstellen	(Systemstandardprozess) dsp	crt
Erzeugen: Prozess (Ersetzen) <i>siehe Anmerkung 1</i>	(Systemstandardprozess) dsp to: chg	crt
Warteschlange erstellen	(Systemstandardwarteschlange) dsp	crt
Warteschlange erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	(Systemstandardwarteschlange) dsp to: dsp und chg	crt
Service erstellen	(Systemstandardwarteschlange) dsp	crt
Service erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	(Systemstandardwarteschlange) dsp to: chg	crt
Authentifizierungsinformationen löschen	dsp und dlt	Nicht zutreffend
Berechtigungssatz löschen	(Warteschlangenmanagerobjekt) chg <i>siehe Anmerkung 4</i>	<i>siehe Anmerkung 4</i>
Kanal löschen	dsp und dlt	Nicht zutreffend
Kanallistener löschen	dsp und dlt	Nicht zutreffend
Clientverbindungskanal löschen	dsp und dlt	Nicht zutreffend
Delete Namelist	dsp und dlt	Nicht zutreffend
Prozess löschen	dsp und dlt	Nicht zutreffend
Warteschlange löschen	dsp und dlt	Nicht zutreffend
Service löschen	dsp und dlt	Nicht zutreffend
Authentifizierungsinformationen abgefragt	DSP	Nicht zutreffend
Berechtigungsdatensätze anfragen	<i>siehe Anmerkung 4</i>	<i>siehe Anmerkung 4</i>
Kanalinquire	DSP	Nicht zutreffend
Channel-Listener inquire	DSP	Nicht zutreffend
Kanalstatus abfragen (für ChannelType MQCHT_CLSSDR)	inq	Nicht zutreffend
Clientverbindungskanal inquire	DSP	Nicht zutreffend
Namensliste inquire	DSP	Nicht zutreffend
Prozess inquire	DSP	Nicht zutreffend
Warteschlange einfragen	DSP	Nicht zutreffend

<i>Tabelle 1. Objektberechtigungen (Forts.)</i>		
Befehl	IBM MQ-Objektberechtigung	Klassenberechtigung (für Objekttyp)
Warteschlangenmanager abfragen	<i>siehe Anmerkung 3</i>	Nicht zutreffend
Warteschlangenstatus abgefragt	DSP	Nicht zutreffend
Inquire Service	DSP	Nicht zutreffend
Pingkanal	Strg	Nicht zutreffend
Ping-WS-Manager	<i>siehe Anmerkung 3</i>	Nicht zutreffend
WS-Manager aktualisieren	(Warteschlangenmanagerobjekt) chg	Nicht zutreffend
Sicherheit aktualisieren (für SecurityType MQSECTYPE_SSL)	(Warteschlangenmanagerobjekt) chg	Nicht zutreffend
Kanal zurücksetzen	ctrlx	Nicht zutreffend
Warteschlangenmanager zurücksetzen	(Warteschlangenmanagerobjekt) chg	Nicht zutreffend
Warteschlangenstatistik zurücksetzen	dsp und chg	Nicht zutreffend
Kanal auflösen	ctrlx	Nicht zutreffend
Berechtigungssatz festlegen	<i>(Warteschlangenmanagerobjekt) chg siehe Anmerkung 4</i>	<i>siehe Anmerkung 4</i>
Kanal starten	Strg	Nicht zutreffend
Kanal stoppen	Strg	Nicht zutreffend
Stop Connection	(Warteschlangenmanagerobjekt) chg	Nicht zutreffend
Listener starten	Strg	Nicht zutreffend
Listener stoppen	Strg	Nicht zutreffend
Service starten	Strg	Nicht zutreffend
Service stoppen	Strg	Nicht zutreffend
Escapezeichen	<i>siehe Anmerkung 2</i>	<i>siehe Anmerkung 2</i>

Anmerkungen:

1. Dieser Befehl wird angewendet, wenn das zu ersetzende Objekt vorhanden ist, andernfalls ist die Berechtigungs-Prüfung für "Erstellen" oder "Kopieren ohne Ersetzen".
2. Die erforderliche Berechtigung wird durch den MQSC-Befehl bestimmt, der durch den Escape-Text definiert wird, und entspricht einem der vorherigen Befehle.
3. Um einen PCF-Befehl verarbeiten zu können, muss die Benutzer-ID über dsp-Berechtigung für das WS-Manager-Objekt auf dem Zielsystem verfügen.
4. Dieser PCF-Befehl ist nur dann autorisiert, wenn der Befehlsserver mit dem Parameter -a gestartet wurde. Standardmäßig wird der Befehlsserver gestartet, wenn der Warteschlangenmanager gestartet wird, und ohne den Parameter -a. Weitere Informationen hierzu finden Sie im Abschnitt [Programmierbare Befehlsformate-Referenz](#).
5. Das Erteilen einer Benutzer-ID chg für einen Warteschlangenmanager bietet die Möglichkeit, Berechtigungsdatensätze für alle Gruppen und Benutzer festzulegen. Erteilen Sie diese Berechtigung nicht für normale Benutzer oder Anwendungen.

IBM MQ stellt auch einige Kanalsicherheitsexitpunkte bereit, sodass Sie Ihre eigenen Benutzerexitprogramme für die Sicherheitsprüfung angeben können. Weitere Informationen finden Sie im Abschnitt [Kanal anzeigen](#).

Multi Verwendung von MQAI zur Vereinfachung der Verwendung von PCFs

IBM MQ Administration Interface (MQAI) ist eine Verwaltungsschnittstelle zu IBM MQ, die unter AIX, HP-UX, IBM i, Linux, Solaris, und Windows verfügbar ist. Sie führt Verwaltungstasks in einem IBM MQ-Warteschlangenmanager mithilfe von Datenbehältern aus, um Eigenschaften (oder Parameter) von Objekten auf eine Art und Weise zu handhaben, die einfacher ist als die Verwendung von PCFs (Programmable Command Formats).

Der MQAI führt Verwaltungstasks in einem Warteschlangenmanager durch die Verwendung von *Datensäcken* aus. Mit Hilfe von Datensäcken können Sie Eigenschaften (oder Parameter) von Objekten in einer Weise bearbeiten, die einfacher ist als die Verwendung von PCFs.

Die Vorteile der Verwendung des MQAI sind folgende:

Einfachere Verwendung von PCF-Nachrichten

Die MQAI ist eine einfachere Methode für die Verwaltung von IBM MQ. Wenn Sie die MQAI verwenden, müssen Sie Ihre eigenen PCF-Nachrichten nicht schreiben. Dadurch werden die Probleme im Zusammenhang mit komplexen Datenstrukturen vermieden.

Um Parameter in Programmen zu übergeben, die mit MQI-Aufrufen geschrieben werden, muss die PCF-Nachricht den Befehl und die Details der Zeichenfolge oder ganzzahligen Daten enthalten. Um diese Konfiguration manuell zu erstellen, müssen Sie in Ihrem Programm mehrere Anweisungen für jede Struktur hinzufügen, und Sie müssen Speicherbereich zuordnen. Diese Aufgabe kann langwierig und mühselig sein.

Programme, die mit dem Parameter MQAI geschrieben wurden, übergeben Parameter in den entsprechenden Datenbehälter, und Sie benötigen nur eine Anweisung für jede Struktur. Die Verwendung der MQAI-Datentaschen entfernt die Notwendigkeit, Arrays zu bearbeiten und Speicher zuzuordnen, und bietet einen gewissen Grad an Isolation von den Details der PCF-Datei.

Fehlerbedingungen leichter handhaben

Es ist schwierig, Rückkehrcodes von PCF-Befehlen zurückzugeben. Die MQAI erleichtert es dem Programm, Fehlerbedingungen zu verarbeiten.

Daten zwischen Anwendungen austauschen

Die Anwendungsdaten werden im PCF-Format gesendet und von der MQAI gepackt und entpackt. Wenn Ihre Nachrichtendaten aus Ganzzahlen und Zeichenfolgen bestehen, können Sie mit der MQAI die Vorteile der integrierten IBM MQ-Datenkonvertierung für PCF-Daten nutzen. Auf diese Weise wird vermieden, dass Datenkonvertierungsexits geschrieben werden müssen.

Nachdem Sie Ihren Datenbehälter erstellt und gefüllt haben, können Sie mithilfe des Aufrufs `mqExecute` eine Verwaltungsbefehlsnachricht an den Befehlsserver eines Warteschlangenmanagers senden. Dieser Aufruf wartet auf alle Antwortnachrichten. Der Aufruf `mqExecute` verarbeitet den Austausch mit dem Befehlsserver und gibt Antworten in einem *Antwortbehälter* zurück.

Beispiele für die Verwendung der MQAI

Die angezeigte Liste enthält einige Beispielprogramme, die die Verwendung von MQAI veranschaulichen. Die Beispiele führen die folgenden Tasks aus:

1. Erstellen Sie eine lokale Warteschlange, „[C-Beispielprogramm zum Erstellen einer lokalen Warteschlange \(amqsaicq.c\)](#)“ auf Seite 41
2. Ereignisse in der Anzeige mit einem einfachen Ereignismonitor anzeigen, „[C-Beispielprogramm für die Anzeige von Ereignissen mit einem Ereignismonitor \(amqsaieim.c\)](#)“ auf Seite 45
3. Drucken Sie eine Liste mit allen lokalen Warteschlangen und deren aktuellen Tiefen aus, „[C-Beispielprogramm für die Abfrage von Warteschlangen und Druckinformationen \(amqsailq.c\)](#)“ auf Seite 57

4. Drucken Sie eine Liste aller Kanäle und deren Typen aus „[C-Beispielprogramm für Abfrage von Kanalobjekten \(amqsaicl.c\)](#)“ auf Seite 52

Erstellen Ihrer MQAI-Anwendung

Wenn Sie Ihre Anwendung mit Hilfe der MQAI erstellen möchten, verlinken Sie dieselben Bibliotheken wie für IBM MQ. Informationen zum Erstellen von IBM MQ-Anwendungen finden Sie im Abschnitt [Prozedurale Anwendung erstellen](#).

Hinweise und Tipps zur Konfiguration von IBM MQ mithilfe von MQAI

Die MQAI verwendet PCF-Nachrichten, um Verwaltungsbefehle an den Befehlsserver zu senden, anstatt direkt mit dem Befehlsserver selbst zu arbeiten. Tipps zur Konfiguration von IBM MQ mithilfe der MQAI finden Sie im Abschnitt [„Hinweise und Tipps zur Verwendung von MQAI für die Konfiguration von IBM MQ“](#) auf Seite 36.

Zugehörige Informationen

[Referenzinformationen zur IBM MQ Administration Interface](#)

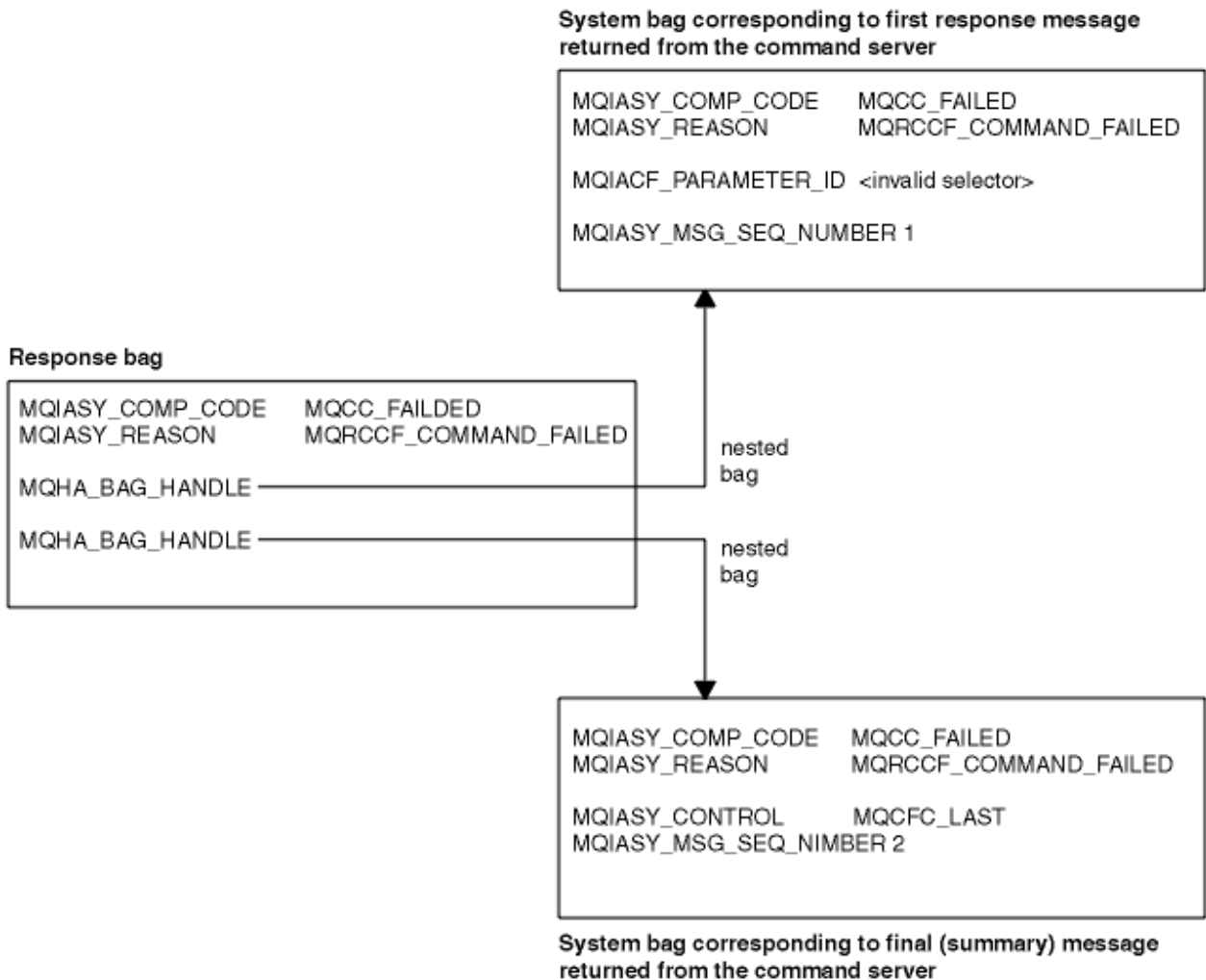
Hinweise und Tipps zur Verwendung von MQAI für die Konfiguration von IBM MQ

IBM MQ Administration Interface (MQAI) verwendet PCF-Nachrichten, um Verwaltungsbefehle an den Befehlsserver zu senden, anstatt direkt mit dem Befehlsserver selbst zu arbeiten. Im Folgenden finden Sie einige Tipps für die Konfiguration von IBM MQ mithilfe der MQAI.

- Zeichenfolgen in IBM MQ werden bis zu einer festen Länge mit Leerzeichen aufgefüllt. Bei Verwendung von C können auf null endende Zeichenfolgen normalerweise als Eingabeparameter an IBM MQ-Programmierschnittstellen übergeben werden.
- Wenn Sie den Wert eines Zeichenfolgeattributs löschen möchten, setzen Sie ihn auf ein einzelnes Leerzeichen und nicht auf eine leere Zeichenfolge.
- Berücksichtigen Sie im Voraus die Attribute, die Sie ändern möchten, und fragen Sie nur diese Attribute ab.
- Bestimmte Attribute können nicht geändert werden, z. B. ein Warteschlangenname oder ein Kanaltyp. Stellen Sie sicher, dass Sie versuchen, nur die Attribute zu ändern, die geändert werden können. Weitere Informationen finden Sie in der Liste der erforderlichen und optionalen Parameter für das jeweilige PCF-Änderungsobjekt. Siehe [Definitionen der programmierbaren Befehlsformate](#).
- Wenn ein MQAI-Aufruf fehlschlägt, wird ein Teil des Fehlers an den Antwortbehälter zurückgegeben. Weitere Details können dann in einem verschachtelten Sack gefunden werden, auf den der Selektor MQHA_BAG_HANDLE zugreifen kann. Wenn z. B. ein MQExecute-Aufruf mit dem Ursachencode MQRCCF_COMMAND_FAILED fehlschlägt, wird diese Information in der Antworttasche zurückgegeben. Eine mögliche Ursache für diesen Ursachencode ist, dass ein angegebener Selektor für den Typ der Befehlsnachricht nicht gültig ist und dass diese Detailinformationen in einem verschachtelten Sack gefunden werden, auf den ein Taschengriff zugreifen kann.

Weitere Informationen zu MQExecute finden Sie in [„Verwaltungsbefehle mit dem mqExecute-Aufruf an den Befehl 'qm' senden“](#) auf Seite 70.

Das folgende Diagramm zeigt dieses Szenario:



Erweiterte MQAI-Themen

Informationen zur Indexierung, Datenkonvertierung und Verwendung des Nachrichtendeskriptors

- Indexierung

Indizes werden verwendet, wenn vorhandene Datenelemente aus einem Behälter ersetzt oder entfernt werden, um die Einfügefølge zu erhalten. Vollständige Details zur Indexierung finden Sie unter [„Indexierung in der MQAI“](#) auf Seite 37.

- Datenkonvertierung

Die in einem MQAI-Datenbehälter enthaltenen Zeichenfolgen können in einer Vielzahl von codierten Zeichensätzen enthalten sein und diese können mit dem Aufruf 'mqSetInteger' konvertiert werden. Vollständige Details zur Datenkonvertierung finden Sie unter [„Verarbeitung der Datenkonvertierung in der MQAI“](#) auf Seite 39.

- Verwendung des Nachrichtendeskriptors

MQAI generiert einen Nachrichtendeskriptor, der bei der Erstellung des Datensacks auf einen Anfangswert gesetzt wird. Vollständige Details zur Verwendung des Nachrichtendeskriptors finden Sie unter [„Verwendung des Nachrichtendeskriptors in der MQAI“](#) auf Seite 40.

Indexierung in der MQAI

Indizes werden zum Ersetzen oder Entfernen bereits vorhandener Datenelemente in einem Behälter verwendet. Es gibt drei Möglichkeiten zur Indexierung, mit denen Datenelemente auf einfache Weise abgerufen werden können.

Jedem Selektor und Wert eines Datenelements in einem Behälter werden drei Indexnummern zugeordnet:

- Der Index relativ zu anderen Elementen mit demselben Selektor.
- Der Index relativ zur Selektorkategorie (Benutzer oder System), zu der das Element gehört.
- Der Index relativ zu allen Datenelementen im Behälter (Benutzer und System).

Dies ermöglicht ein Indexieren nach Benutzerselektoren, Systemselektoren oder beiden Selektortypen, wie in [Abbildung 4 auf Seite 38](#) dargestellt.

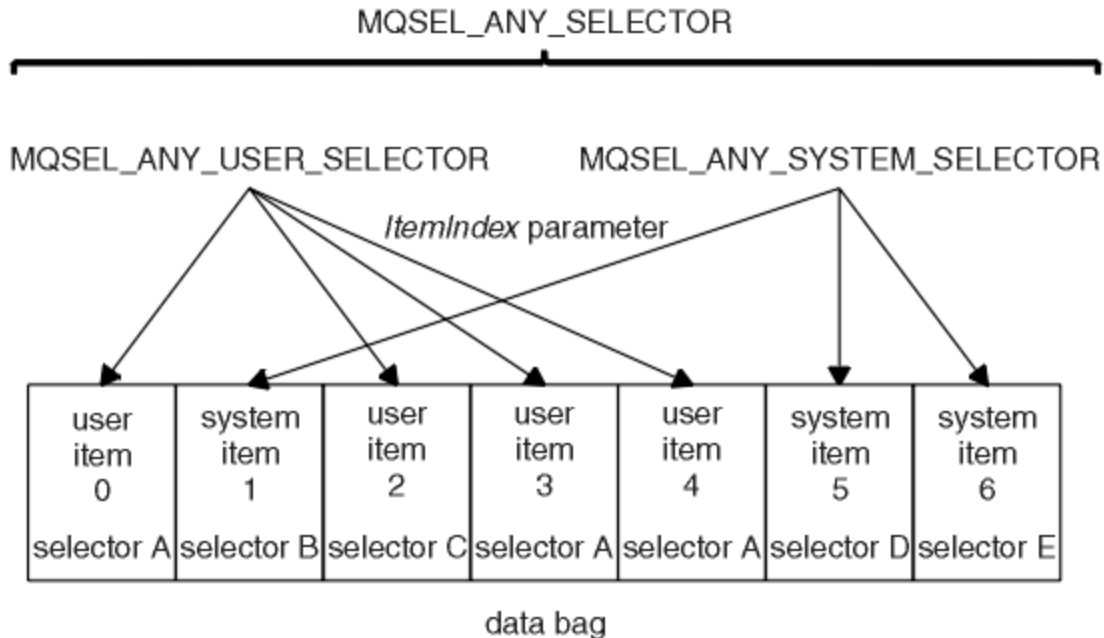


Abbildung 4. Indexierung

In [Abbildung 4 auf Seite 38](#) kann mit den folgenden Indexpaaren auf Benutzerelement 3 (Selektor A) verwiesen werden:

Selector	ItemIndex
Selektor A	1
MQSEL_ANY_USER_SELECTOR	2
MQSEL_ANY_SELECTOR	3

Der Index basiert auf null, genau wie eine Gruppe in C. Bei "n" Vorkommen reicht der Index lückenlos von null bis "n-1".

Indizes werden zum Ersetzen oder Entfernen bereits vorhandener Datenelemente in einem Behälter verwendet. Diese Verwendungsweise wirkt sich nicht auf die Einfügereihenfolge, aber möglicherweise auf die Indizes anderer Datenelemente aus. Beispiele hierzu finden Sie in den Abschnitten [Informationen in einem Behälter ändern](#) und [Datenelemente ändern](#).

Durch die drei Indexierungstypen lassen sich Datenelemente leicht abrufen. Wenn in einem Behälter beispielsweise drei Instanzen eines bestimmten Selektors vorkommen, kann mit dem Aufruf "mqCountItems" die Anzahl der Instanzen dieses Selektors gezählt werden und mit den Aufrufen vom Typ "mqInquire*" können sowohl der Selektor als auch der Index angegeben werden, um nur diese Werte abzufragen. Diese Vorgehensweise eignet sich für Attribute, die eine Werteliste aufweisen können, z. B. einige Exits von Kanälen.

Verarbeitung der Datenkonvertierung in der MQAI

Die in einem MQAI-Datenbehälter enthaltenen Zeichenfolgen können in einer Vielzahl von codierten Zeichensätzen enthalten sein. Diese Zeichenfolgen können mit dem Aufruf 'mqSetInteger' konvertiert werden.

Genau wie PCF-Nachrichten können auch die Zeichenfolgen in einem MQAI-Datenbehälter verschiedene codierte Zeichensätze aufweisen. Normalerweise gilt für alle Zeichenfolgen in einer PCF-Nachricht derselbe codierte Zeichensatz, der dem Zeichensatz des Warteschlangenmanagers entspricht.

Jedes Zeichenfolgeelement in einem Datenbehälter enthält zwei Werte: die Zeichenfolge selbst und die ID des codierten Zeichensatzes (CCSID). Die Zeichenfolge, die zum Behälter hinzugefügt wird, stammt aus dem Parameter **Buffer** des Aufrufs mqAddString oder mqSetString. Die CCSID stammt aus dem Systemelement, das den Selektor MQIASY_CODED_CHAR_SET_ID enthält. Diese ID wird als *Behälter-CCSID* bezeichnet und kann mit dem Aufruf "mqSetInteger" geändert werden.

Wenn Sie den Wert einer Zeichenfolge abrufen, die sich in einem Datenbehälter befindet, stellt die CCSID einen Ausgabeparameter des Aufrufs dar.

In Tabelle 2 auf Seite 39 sind die Regeln dargestellt, die beim Konvertieren von Datenbehältern in Nachrichten und umgekehrt angewendet werden:

Tabelle 2. CCSID-Verarbeitung			
MQAI-Aufruf	CCSID	Eingabe des Aufrufs	Ausgabe des Aufrufs
mqBagToBuffer	Behälter-CCSID (1)	Ignoriert	Nicht geändert
mqBagToBuffer	Zeichenfolge-CCSIDs in Behälter	Genutzt	Nicht geändert
mqBagToBuffer	Zeichenfolge-CCSIDs in Puffer	Nicht zutreffend	Kopiert aus Zeichenfolge-CCSIDs in Behälter
mqBufferToBag	Behälter-CCSID (1)	Ignoriert	Nicht geändert
mqBufferToBag	Zeichenfolge-CCSIDs in Puffer	Genutzt	Nicht geändert
mqBufferToBag	Zeichenfolge-CCSIDs in Behälter	Nicht zutreffend	Kopiert aus Zeichenfolge-CCSIDs in Puffer
mqPutBag	MQMD-CCSID	Genutzt	Nicht geändert (2)
mqPutBag	Behälter-CCSID (1)	Ignoriert	Nicht geändert
mqPutBag	Zeichenfolge-CCSIDs in Behälter	Genutzt	Nicht geändert
mqPutBag	Zeichenfolge-CCSIDs in gesendeter Nachricht	Nicht zutreffend	Kopiert aus Zeichenfolge-CCSIDs in Behälter
mqGetBag	MQMD-CCSID	Für Datenkonvertierung der Nachricht verwendet	Auf CCSID der zurückgegebenen Daten gesetzt (3)
mqGetBag	Behälter-CCSID (1)	Ignoriert	Nicht geändert
mqGetBag	Zeichenfolge-CCSIDs in Nachricht	Genutzt	Nicht geändert
mqGetBag	Zeichenfolge-CCSIDs in Behälter	Nicht zutreffend	Kopiert aus Zeichenfolge-CCSIDs in Nachricht
mqExecute	Anforderungsbehälter-CCSID	Für MQMD der Anforderungsnachricht verwendet (4)	Nicht geändert

Tabelle 2. CCSID-Verarbeitung (Forts.)			
MQAI-Aufruf	CCSID	Eingabe des Aufrufs	Ausgabe des Aufrufs
mqExecute	Antwortbehälter-CCSID	Für Datenkonvertierung der Antwortnachricht verwendet (4)	Auf CCSID der zurückgegebenen Daten gesetzt (3)
mqExecute	Zeichenfolge-CCSIDs in Anforderungsbehälter	Für Anforderungsnachricht verwendet	Nicht geändert
mqExecute	Zeichenfolge-CCSIDs in Antwortbehälter	Nicht zutreffend	Kopiert aus Zeichenfolge-CCSIDs in Antwortnachricht

Anmerkungen:

1. Die Behälter-CCSID ist das Systemelement mit dem Selektor MQIASY_CODED_CHAR_SET_ID.
2. MQCCSI_Q_MGR wird in die tatsächliche CCSID des Warteschlangenmanagers geändert.
3. Wenn eine Datenkonvertierung angefordert wird, entspricht die CCSID der zurückgegebenen Daten dem Ausgabewert. Wenn keine Datenkonvertierung angefordert wird, entspricht die CCSID der zurückgegebenen Daten dem Nachrichtenwert. Wenn eine Datenkonvertierung angefordert wird, aber fehlschlägt, wird keine Nachricht zurückgegeben.
4. Wenn die CCSID den Wert MQCCSI_DEFAULT aufweist, wird die CCSID des Warteschlangenmanagers verwendet.

Zugehörige Informationen

Datenkonvertierung

[ccsid_part2.tbl](#) (Datei)

Verwendung des Nachrichtendeskriptors in der MQAI

Der von der MQAI generierte Nachrichtendeskriptor ist bei der Erstellung des Datenbehälters auf einen Anfangswert gesetzt.

Der PCF-Befehlstyp stammt aus dem Systemelement mit dem Selektor MQIASY_TYPE. Beim Erstellen eines Datenbehälters wird der Anfangswert dieses Elements je nach Typ des erstellten Behälters festgelegt:

Tabelle 3. PCF-Befehlstyp	
Behältertyp	Anfangswert des Elements MQIASY_TYPE
MQCBO_ADMIN_BAG	MQCFT_COMMAND
MQCBO_COMMAND_BAG	MQCFT_COMMAND
MQCBO_*	MQCFT_USER

Wenn die MQAI einen Nachrichtendeskriptor generiert, hängen die für die Parameter **Format** und **MsgType** verwendeten Werte vom Wert des Systemelements mit dem Selektor MQIASY_TYPE ab, wie in [Tabelle 3 auf Seite 40](#) dargestellt.

Tabelle 4. Parameter "Format" und "MsgType" des Nachrichtendeskriptors		
PCF-Befehlstyp	Format	MsgType
MQCFT_COMMAND	MQFMT_ADMIN	MQMT_REQUEST
MQCFT_REPORT	MQFMT_ADMIN	MQMT_REPORT
MQCFT_RESPONSE	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQMT_DATAGRAM

PCF-Befehlstyp	Format	MsgType
MQCFT_EVENT	MQFMT_EVENT	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Tabelle 4 auf Seite 40 zeigt Folgendes: Wenn Sie einen Verwaltungs- oder Befehlsbehälter erstellen, hat der Parameter *Format* des Nachrichtendeskriptors den Wert MQFMT_ADMIN und *MsgType* den Wert MQMT_REQUEST. Dies eignet sich, wenn eine PCF-Anforderungsnachricht an den Befehlsserver gesendet und eine Antwort erwartet wird.

Die anderen Parameter im Nachrichtendeskriptor nehmen die in [Tabelle 5 auf Seite 41](#) dargestellten Werte an.

Parameter	Wert
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	Siehe Tabelle 4 auf Seite 40
<i>Expiry</i>	30 Sekunden (Anmerkung „1“ auf Seite 41)
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	Je nach Behälter-CCSID (Anmerkung „2“ auf Seite 41)
<i>Format</i>	Siehe Tabelle 4 auf Seite 40
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	MQMI_NONE
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Siehe Anmerkung „3“ auf Seite 41
<i>ReplyToQMgr</i>	Leer

Anmerkungen:

1. Dieser Wert kann im Aufruf "mqExecute" durch die Verwendung des Parameters **OptionsBag** überschrieben werden. Weitere Informationen hierzu finden Sie unter [mqExecute](#).
2. Siehe „Verarbeitung der Datenkonvertierung in der MQAI“ auf Seite 39.
3. Name der benutzerdefinierten Antwortwarteschlange oder MQAI-generierten temporären dynamischen Warteschlange für Nachrichten vom Typ MQMT_REQUEST. Andernfalls leer.

C-Beispielprogramm zum Erstellen einer lokalen Warteschlange (amqsaicq.c)

Das C-Beispielprogramm `amqsaicq.c` erstellt eine lokale Warteschlange mithilfe der MQAI.

```
/*.....*/
```

```

/*                                                                    */
/* Program name: AMQSAICQ.C                                           */
/*                                                                    */
/* Description:  Sample C program to create a local queue using the   */
/*              IBM MQ Administration Interface (MQAI).               */
/*                                                                    */
/* Statement:   Licensed Materials - Property of IBM                 */
/*                                                                    */
/*              84H2000, 5765-B73                                     */
/*              84H2001, 5639-B42                                     */
/*              84H2002, 5765-B74                                     */
/*              84H2003, 5765-B75                                     */
/*              84H2004, 5639-B43                                     */
/*                                                                    */
/*              (C) Copyright IBM Corp. 1999, 2023.                 */
/*                                                                    */
/*****                                                                    */
/*                                                                    */
/* Function:                                                                    */
/* AMQSAICQ is a sample C program that creates a local queue and is an */
/* example of the use of the mqExecute call.                             */
/*                                                                    */
/* - The name of the queue to be created is a parameter to the program. */
/*                                                                    */
/* - A PCF command is built by placing items into an MQAI bag.         */
/*   These are:-                                                         */
/*     - The name of the queue                                           */
/*     - The type of queue required, which, in this case, is local.    */
/*                                                                    */
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q.   */
/*   The call generates the correct PCF structure.                      */
/*   The call receives the reply from the command server and formats into */
/*   the response bag.                                                  */
/*                                                                    */
/* - The completion code from the mqExecute call is checked and if there */
/*   is a failure from the command server then the code returned by the */
/*   command server is retrieved from the system bag that is           */
/*   embedded in the response bag to the mqExecute call.               */
/*                                                                    */
/* Note: The command server must be running.                           */
/*                                                                    */
/*****                                                                    */
/*                                                                    */
/* AMQSAICQ has 2 parameters - the name of the local queue to be created */
/* - the queue manager name (optional)                                  */
/*                                                                    */
/*****                                                                    */
/* Includes                                                                    */
/*****                                                                    */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF          */
#include <cmqbc.h>        /* MQAI        */

void CheckCallResult(MQCHAR *, MQLONG , MQLONG );
void CreateLocalQueue(MQHCONN, MQCHAR *);

int main(int argc, char *argv[])
{
    MQHCONN hConn;          /* handle to IBM MQ connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG connReason;     /* MQCONN reason code */
    MQLONG compCode;       /* completion code */
    MQLONG reason;         /* reason code */

    /*****                                                                    */
    /* First check the required parameters                                  */
    /*****                                                                    */
    printf("Sample Program to Create a Local Queue\n");
    if (argc < 2)
    {
        printf("Required parameter missing - local queue name\n");
        exit(99);
    }
}

```

```

/*****
/* Connect to the queue manager */
/*****
if (argc > 2)
    strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(QMName, &hConn, &compCode, &connReason);

/*****
/* Report reason and stop if connection failed */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("MQCONN", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Call the routine to create a local queue, passing the handle to the
/* queue manager and also passing the name of the queue to be created.
/*****
CreateLocalQueue(hConn, argv[1]);

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
}
return 0;

}

/*****
/*
/* Function: CreateLocalQueue
/* Description: Create a local queue by sending a PCF command to the command
/* server.
/*
/*
/*****
/*
/* Input Parameters: Handle to the queue manager
/* Name of the queue to be created
/*
/*
/* Output Parameters: None
/*
/*
/* Logic: The mqExecute call is executed with the command MQCMD_CREATE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply is read from the temporary queue and formatted into the
/* response bag.
/*
/* The completion code from the mqExecute call is checked and if there
/* is a failure from the command server then the code returned by the
/* command server is retrieved from the system bag that is
/* embedded in the response bag to the mqExecute call.
/*
/*
/*****
void CreateLocalQueue(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG reason; /* reason code */
    MQLONG compCode; /* completion code */
    MQHBAG commandBag = MQHB_UNUSABLE_HBAG; /* command bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG resultBag; /* result bag from mqExecute */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */

    printf("\nCreating Local Queue %s\n", qName);

    /*****
    /* Create a command Bag for the mqExecute call. Exit the function if the
    /* create fails.
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &commandBag, &compCode, &reason);
    CheckCallResult("Create the command bag", compCode, reason);
    if (compCode !=MQCC_OK)

```

```

return;

/*****
/* Create a response Bag for the mqExecute call, exit the function if the */
/* create fails. */
*****/
mqCreateBag(MQCB0_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create the response bag", compCode, reason);
if (compCode !=MQCC_OK)
    return;

/*****
/* Put the name of the queue to be created into the command bag. This will */
/* be used by the mqExecute call. */
*****/
mqAddString(commandBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, qName, &compCode,
            &reason);
CheckCallResult("Add q name to command bag", compCode, reason);

/*****
/* Put queue type of local into the command bag. This will be used by the */
/* mqExecute call. */
*****/
mqAddInteger(commandBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type to command bag", compCode, reason);

/*****
/* Send the command to create the required local queue. */
/* The mqExecute call will create the PCF structure required, send it to */
/* the command server and receive the reply from the command server into */
/* the response bag. */
*****/
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_CREATE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          commandBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response*/
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE*/
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n")
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call and find the error if it failed. */
*****/
if ( compCode == MQCC_OK )
    printf("Local queue %s successfully created\n", qName);
else
{
    printf("Creation of local queue %s failed: Completion Code = %d
           qName, compCode, reason);
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        /*****
        /* Get the system bag handle out of the mqExecute response bag. */
        /* This bag contains the reason from the command server why the */
        /* command failed. */
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &resultBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        *****/
        mqInquireInteger(resultBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                        &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(resultBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                        &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag", compCode,

```

```

        reason);
    printf("Error returned by the command server: Completion code = %d :
          Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}
/*****
/* Delete the command bag if successfully created. */
/*****
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
    CheckCallResult("Delete the command bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}
} /* end of CreateLocalQueue */

/*****
/*
/* Function: CheckCallResult */
/*
/*
/*****
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
/*
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d :
              Reason = %d\n", callText, cc, rc);
}
}

```

C-Beispielprogramm für die Anzeige von Ereignissen mit einem Ereignismonitor (amqsaiem.c)

Das C-Beispielprogramm amqsaiem.c zeigt einen Basisereignismonitor mithilfe der MQAI.

```

/*****
/*
/* Program name: AMQSAIEM.C */
/*
/* Description: Sample C program to demonstrate a basic event monitor */
/* using the IBM MQ Admin Interface (MQAI). */
/* Licensed Materials - Property of IBM */
/*
/* 63H9336 */
/* (c) Copyright IBM Corp. 1999, 2023. All Rights Reserved. */
/*
/* US Government Users Restricted Rights - Use, duplication or */
/* disclosure restricted by GSA ADP Schedule Contract with */
/* IBM Corp. */
/*****
/*
/* Function: */
/* AMQSAIEM is a sample C program that demonstrates how to write a simple */
/* event monitor using the mqGetBag call and other MQAI calls. */
/*
/* The name of the event queue to be monitored is passed as a parameter */
/* to the program. This would usually be one of the system event queues:- */
/* SYSTEM.ADMIN.QMGR.EVENT Queue Manager events */
/*

```

```

/*          SYSTEM.ADMIN.PERFM.EVENT          Performance events          */
/*          SYSTEM.ADMIN.CHANNEL.EVENT       Channel events             */
/*          SYSTEM.ADMIN.LOGGER.EVENT        Logger events               */
/*          */                                                            */
/* To monitor the queue manager event queue or the performance event queue, */
/* the attributes of the queue manager need to be changed to enable         */
/* these events. For more information about this, see Part 1 of the         */
/* Programmable System Management book. The queue manager attributes can   */
/* be changed using either MQSC commands or the MQAI interface.           */
/* Channel events are enabled by default.                                   */
/*          */                                                            */
/* Program logic                                                            */
/* Connect to the Queue Manager.                                           */
/* Open the requested event queue with a wait interval of 30 seconds.      */
/* Wait for a message, and when it arrives get the message from the queue  */
/* and format it into an MQAI bag using the mqGetBag call.                 */
/* There are many types of event messages and it is beyond the scope of   */
/* this sample to program for all event messages. Instead the program     */
/* prints out the contents of the formatted bag.                           */
/* Loop around to wait for another message until either there is an error  */
/* or the wait interval of 30 seconds is reached.                          */
/*          */                                                            */
/*****
/*
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored */
/* - the queue manager name (optional)                                    */
/*          */                                                            */
/*****

/*****
/* Includes                                                                */
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF         */
#include <cmqbc.h>        /* MQAI        */

/*****
/* Macros                                                                    */
/*****
#if MQAT_DEFAULT == MQAT_WINDOWS_NT
#define Int64 "I64"
#elif defined(MQ_64_BIT)
#define Int64 "l"
#else
#define Int64 "ll"
#endif

/*****
/* Function prototypes                                                       */
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
void GetQEvents(MQHCONN, MQCHAR *);
int PrintBag(MQHBAG);
int PrintBagContents(MQHBAG, int);

/*****
/* Function: main                                                            */
/*****
int main(int argc, char *argv[])
{
    MQHCONN hConn;          /* handle to connection          */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QM name             */
    MQLONG reason;         /* reason code                   */
    MQLONG connReason;     /* MQCONN reason code           */
    MQLONG compCode;       /* completion code               */

    /*****
    /* First check the required parameters                                  */
    /*****
    printf("Sample Event Monitor (times out after 30 secs)\n");
    if (argc < 2)
    {
        printf("Required parameter missing - event queue to be monitored\n");
        exit(99);
    }
}

/*****

```

```

/* Connect to the queue manager */
/*****
if (argc > 2)
    strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(QMName, &hConn, &compCode, &connReason);
/*****
/* Report the reason and stop if the connection failed */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("MQCONN", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Call the routine to open the event queue and format any event messages */
/* read from the queue. */
/*****
GetQEvents(hConn, argv[1]);

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
}

return 0;
}

/*****
/*
/* Function: CheckCallResult */
/*
/*****
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

/*****
/*
/* Function: GetQEvents */
/*
/*****
/*
/* Input Parameters: Handle to the queue manager */
/* Name of the event queue to be monitored */
/*
/* Output Parameters: None */
/*
/* Logic: Open the event queue. */
/* Get a message off the event queue and format the message into */
/* a bag. */
/* A real event monitor would need to be programmed to deal with */
/* each type of event that it receives from the queue. This is */
/* outside the scope of this sample, so instead, the contents of */
/* the bag are printed. */
/* The program waits for 30 seconds for an event message and then */
/* terminates if no more messages are available. */
/*
/*****
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason; /* MQOPEN reason code */

```

```

MQLONG reason; /* reason code */
MQLONG compCode; /* completion code */
MQHOBJ eventQueue; /* handle to event queue */

MQHBAG eventBag = MQHB_UNUSABLE_HBAG; /* event bag to receive event msg */
MQOD od = {MQOD_DEFAULT}; /* Object Descriptor */
MQMD md = {MQMD_DEFAULT}; /* Message Descriptor */
MQGMO gmo = {MQGMO_DEFAULT}; /* get message options */
MQLONG bQueueOK = 1; /* keep reading msgs while true */

/*****
/* Create an Event Bag in which to receive the event. */
/* Exit the function if the create fails. */
*****/
mqCreateBag(MQCB0_USER_BAG, &eventBag, &compCode, &reason);
CheckCallResult("Create event bag", compCode, reason);
if (compCode != MQCC_OK)
    return;

/*****
/* Open the event queue chosen by the user */
*****/
strncpy(od.ObjectName, qName, (size_t)MQ_Q_NAME_LENGTH);
MQOPEN(hConn, &od, MQOO_INPUT_AS_Q_DEF+MQOO_FAIL_IF_QUIESCING, &eventQueue,
        &compCode, &openReason);
CheckCallResult("Open event queue", compCode, openReason);

/*****
/* Set the GMO options to control the action of the get message from the */
/* queue. */
*****/
gmo.WaitInterval = 30000; /* 30 second wait for message */
gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF_QUIESCING + MQGMO_CONVERT;
gmo.Version = MQGMO_VERSION_2; /* Avoid need to reset Message ID */
gmo.MatchOptions = MQMO_NONE; /* and Correlation ID after every */
/* mqGetBag */

/*****
/* If open fails, we cannot access the queue and must stop the monitor. */
*****/
if (compCode != MQCC_OK)
    bQueueOK = 0;

/*****
/* Main loop to get an event message when it arrives */
*****/
while (bQueueOK)
{
    printf("\nWaiting for an event\n");

    /*****
    /* Get the message from the event queue and convert it into the event */
    /* bag. */
    *****/
    mqGetBag(hConn, eventQueue, &md, &gmo, eventBag, &compCode, &reason);

    /*****
    /* If get fails, we cannot access the queue and must stop the monitor. */
    *****/
    if (compCode != MQCC_OK)
    {
        bQueueOK = 0;

        /*****
        /* If get fails because no message available then we have timed out, */
        /* so report this, otherwise report an error. */
        *****/
        if (reason == MQRC_NO_MSG_AVAILABLE)
        {
            printf("No more messages\n");
        }
        else
        {
            CheckCallResult("Get bag", compCode, reason);
        }
    }
}

/*****
/* Event message read - Print the contents of the event bag */
*****/
else
{
    if ( PrintBag(eventBag) )

```



```

        printf("\nError found while printing bag contents\n");
    } /* end of msg found */
} /* end of main loop */
/*****
/* Close the event queue if successfully opened */
/*****
if (openReason == MQRC_NONE)
{
    MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
    CheckCallResult("Close event queue", compCode, reason);
}

/*****
/* Delete the event bag if successfully created. */
/*****
if (eventBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&eventBag, &compCode, &reason);
    CheckCallResult("Delete the event bag", compCode, reason);
}

} /* end of GetQEvents */

/*****
/*
/* Function: PrintBag
/*
/*****
/*
/* Input Parameters: Bag Handle
/*
/*
/* Output Parameters: None
/*
/*
/* Returns: Number of errors found
/*
/*
/* Logic: Calls PrintBagContents to display the contents of the bag.
/*
/*
/*****

int PrintBag(MQHBAG dataBag)
{
    int errors;

    printf("\n");
    errors = PrintBagContents(dataBag, 0);
    printf("\n");

    return errors;
}

/*****
/*
/* Function: PrintBagContents
/*
/*****
/*
/* Input Parameters: Bag Handle
/*
/* Indentation level of bag
/*
/*
/* Output Parameters: None
/*
/*
/* Returns: Number of errors found
/*
/*
/* Logic: Count the number of items in the bag
/*
/* Obtain selector and item type for each item in the bag.
/*
/* Obtain the value of the item depending on item type and display the
/*
/* index of the item, the selector and the value.
/*
/* If the item is an embedded bag handle then call this function again
/*
/* to print the contents of the embedded bag increasing the
/*
/* indentation level.
/*
/*****
int PrintBagContents(MQHBAG dataBag, int indent)
{
    /*****
    /* Definitions
    /*****
    /*****
    #define LENGTH 500 /* Max length of string to be read*/
    #define INDENT 4 /* Number of spaces to indent */
    /* embedded bag display */

```

```

/*****
/* Variables
/*****
MQLONG itemCount; /* Number of items in the bag */
MQLONG itemType; /* Type of the item */
int i; /* Index of item in the bag */
MQCHAR stringVal[LENGTH+1]; /* Value if item is a string */
MQBYTE byteStringVal[LENGTH]; /* Value if item is a byte string */
MQLONG stringLength; /* Length of string value */
MQLONG ccsid; /* CCSID of string value */
MQINT32 iValue; /* Value if item is an integer */
MQINT64 i64Value; /* Value if item is a 64-bit
/* integer
MQLONG selector; /* Selector of item
MQHBAG bagHandle; /* Value if item is a bag handle
MQLONG reason; /* reason code
MQLONG compCode; /* completion code
MQLONG trimLength; /* Length of string to be trimmed
int errors = 0; /* Count of errors found
char blanks[] = " /* Blank string used to
/* indent display

/*****
/* Count the number of items in the bag
/*****
mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("
    printf("
    printf("
}

/*****
/* If no errors found, display each item in the bag
/*****
if (!errors)
{
    for (i = 0; i < itemCount; i++)
    {

/*****
/* First inquire the type of the item for each item in the bag
/*****
mqInquireItemInfo(dataBag, /* Bag handle
/*
MQSEL_ANY_SELECTOR, /* Item can have any selector*/
i, /* Index position in the bag */
&selector, /* Actual value of selector
/* returned by call
/*
&itemType, /* Actual type of item
/* returned by call
/*
&compCode, /* Completion code
/*
&reason); /* Reason Code

if (compCode != MQCC_OK)
    errors++;

switch(itemType)
{
case MQITEM_INTEGER:
/*****
/* Item is an integer. Find its value and display its index,
/*
/* selector and value.
/*****
mqInquireInteger(dataBag, /* Bag handle
/*
MQSEL_ANY_SELECTOR, /* Allow any selector
/*
i, /* Index position in the bag */
&iValue, /* Returned integer value
/*
&compCode, /* Completion code
/*
&reason); /* Reason Code

if (compCode != MQCC_OK)
    errors++;
else
    printf("%.s %-2d %-4d (%d)\n",
        indent, blanks, i, selector, iValue);
break

```

```

case MQITEM_INTEGER64:
/*****
/* Item is a 64-bit integer. Find its value and display its */
/* index, selector and value. */
/*****
mqInquireInteger64(dataBag, /* Bag handle */
MQSEL_ANY_SELECTOR, /* Allow any selector */
i, /* Index position in the bag */
&i64Value, /* Returned integer value */
&compCode, /* Completion code */
&reason); /* Reason Code */

if (compCode != MQCC_OK)
errors++;
else
printf("%.s %-2d %-4d (%"Int64"d)\n",
indent, blanks, i, selector, i64Value);
break;

case MQITEM_STRING:
/*****
/* Item is a string. Obtain the string in a buffer, prepare */
/* the string for displaying and display the index, selector, */
/* string and Character Set ID. */
/*****
mqInquireString(dataBag, /* Bag handle */
MQSEL_ANY_SELECTOR, /* Allow any selector */
i, /* Index position in the bag */
LENGTH, /* Maximum length of buffer */
stringVal, /* Buffer to receive string */
&stringLength, /* Actual length of string */
&ccsid, /* Coded character set ID */
&compCode, /* Completion code */
&reason); /* Reason Code */

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check */
/* explicitly for call failure. */
/*****
if (compCode == MQCC_FAILED)
errors++;
else
{
/*****
/* Remove trailing blanks from the string and terminate with*/
/* a null. First check that the string should not have been */
/* longer than the maximum buffer size allowed. */
/*****
if (stringLength > LENGTH)
trimLength = LENGTH;
else
trimLength = stringLength;
mqTrim(trimLength, stringVal, stringVal, &compCode, &reason);
printf("%.s %-2d %-4d '%s' %d\n",
indent, blanks, i, selector, stringVal, ccsid);
}
break;

case MQITEM_BYTE_STRING:
/*****
/* Item is a byte string. Obtain the byte string in a buffer, */
/* prepare the byte string for displaying and display the */
/* index, selector and string. */
/*****
mqInquireByteString(dataBag, /* Bag handle */
MQSEL_ANY_SELECTOR, /* Allow any selector */
i, /* Index position in the bag */
LENGTH, /* Maximum length of buffer */
byteStringVal, /* Buffer to receive string */
&stringLength, /* Actual length of string */
&compCode, /* Completion code */
&reason); /* Reason Code */

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check */
/* explicitly for call failure. */
/*****
if (compCode == MQCC_FAILED)
errors++;

```

```

else
{
    printf("%.s %-2d    %-4d    X'",
           indent, blanks, i, selector);

    for (i = 0 ; i < stringLength ; i++)
        printf("

    printf("\n");
}
break;

case MQITEM_BAG:
/*****
/* Item is an embedded bag handle, so call the PrintBagContents*/
/* function again to display the contents.
*****/
mqInquireBag(dataBag,          /* Bag handle          */
             MQSEL_ANY_SELECTOR, /* Allow any selector */
             i,                /* Index position in the bag */
             &bagHandle,       /* Returned embedded bag handle*/
             &compCode,        /* Completion code          */
             &reason);         /* Reason Code              */

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("%.s %-2d    %-4d    (%d)\n", indent, blanks, i,
           selector, bagHandle);
    if (selector == MQHA_BAG_HANDLE)
        printf("
    else
        printf("
        PrintBagContents(bagHandle, indent+INDENT);
    }
    break;

default:
    printf("
}
}
}
return errors;
}

```

C-Beispielprogramm für Abfrage von Kanalobjekten (amqsaicl.c)

Das C-Beispielprogramm amqsaicl.c fragt Kanalobjekte mithilfe der MQAI ab.

```

/*****
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/*              using the IBM MQ Administration Interface (MQAI)
/*
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2023. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
*****/
/*
/* Function:
/* AMQSAICL is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires all channels and their types.
/*
/* - A PCF command is built from items placed into an MQAI administration
/*   bag.
/*   These are:-
/*   - The generic channel name "*"
/*

```

```

/*          - The attributes to be inquired. In this sample we just want */
/*          name and type attributes */
/*          */
/*          - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed. */
/*          The call generates the correct PCF structure. */
/*          The default options to the call are used so that the command is sent */
/*          to the SYSTEM.ADMIN.COMMAND.QUEUE. */
/*          The reply from the command server is placed on a temporary dynamic */
/*          queue. */
/*          The reply from the MQCMD_INQUIRE_CHANNEL is read from the */
/*          temporary queue and formatted into the response bag. */
/*          */
/*          - The completion code from the mqExecute call is checked and if there */
/*          is a failure from the command server, then the code returned by the */
/*          command server is retrieved from the system bag that has been */
/*          embedded in the response bag to the mqExecute call. */
/*          */
/* Note: The command server must be running. */
/*          */
/*****
/* AMQSAICL has 2 parameter - the queue manager name (optional) */
/*          - output file (optional) default varies */
*****/

/*****
/* Includes */
*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF          */
#include <cmqbc.h>        /* MQAI         */
#include <cmqxc.h>        /* MQCD         */

/*****
/* Function prototypes */
*****/
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* DataTypes */
*****/
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFILE OUTFILEHDL;
#else
typedef FILE OUTFILEHDL;
#endif

/*****
/* Constants */
*****/
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "*SDR      ", /* MQCHT_SENDER */
    "*SVR      ", /* MQCHT_SERVER */
    "*RCVR     ", /* MQCHT_RECEIVER */
    "*RQSTR    ", /* MQCHT_REQUESTER */
    "*ALL      ", /* MQCHT_ALL */
    "*CLTCN    ", /* MQCHT_CLNTCONN */
    "*SVRCONN  ", /* MQCHT_SVRCONN */
    "*CLUSRCVR", /* MQCHT_CLUSRCVR */
    "*CLUSSDR  "  /* MQCHT_CLUSSDR */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER */
    "svr      ", /* MQCHT_SERVER */

```

```

"rcvr      ", /* MQCHT_RECEIVER */
"rqstr     ", /* MQCHT_REQUESTER */
"all       ", /* MQCHT_ALL */
"cltconn   ", /* MQCHT_CLNTCONN */
"svrcn     ", /* MQCHT_SVRCONN */
"clusrcvr  ", /* MQCHT_CLUSRCVR */
"clusdr    ", /* MQCHT_CLUSSDR */
};
#endif

/*****
/* Macros */
*****/
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
(hdl) = _Ropen((fname), "wr", rtncode=Y);
#define CLOSEOUTFILE(hdl) \
_Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
_Rwrite((hdl), (buf), (buflen));
#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
(hdl) = fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#else
#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(fname) \
fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#endif

#define ChlType2String(t) ChlTypeMap[(t)-1].name

/*****
/* Function: main */
*****/
int main(int argc, char *argv[])
{
/*****
/* MQAI variables */
*****/
MQHCONN hConn; /* handle to MQ connection */
MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
MQLONG reason; /* reason code */
MQLONG connReason; /* MQCONN reason code */
MQLONG compCode; /* completion code */
MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
MQHBAG cAttrsBag; /* bag containing chl attributes */
MQHBAG errorBag; /* bag containing cmd server error */
MQLONG mqExecuteCC; /* mqExecute completion code */
MQLONG mqExecuteRC; /* mqExecute reason code */
MQLONG chlNameLength; /* Actual length of chl name */
MQLONG chlType; /* Channel type */
MQLONG i; /* loop counter */
MQLONG numberOfBags; /* number of bags in response bag */
MQCHAR chlName[MQ_OBJECT_NAME_LENGTH+1]; /* name of chl extracted from bag */
MQCHAR OutputBuffer[100]; /* output data buffer */
OUTFILEHDL *outfp = NULL; /* output file handle */

/*****
/* Connect to the queue manager */
*****/
if (argc > 1)
strcpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(qmName, &hConn, &compCode, &connReason);

/*****
/* Report the reason and stop if the connection failed. */
*****/
if (compCode == MQCC_FAILED)

```

```

}
  CheckCallResult("Queue Manager connection", compCode, connReason);
  exit( (int)connReason);
}

/*****
/* Open the output file */
/*****
if (argc > 2)
{
  OPENOUTFILE(outfp, argv[2]);
}
else
{
  OPENOUTFILE(outfp, OUTFILE);
}

if(outfp == NULL)
{
  printf("Could not open output file.\n");
  goto MOD_EXIT;
}
/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag;, &compCode;, &reason;);
CheckCallResult("Create admin bag", compCode, reason);

/*****
/* Create a response bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag;, &compCode;, &reason;);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic channel name into the admin bag */
/*****
mqAddString(adminBag, MQCACH_CHANNEL_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode;, &reason;);
CheckCallResult("Add channel name", compCode, reason);

/*****
/* Put the channel type into the admin bag */
/*****
mqAddInteger(adminBag, MQIACH_CHANNEL_TYPE, MQCHT_ALL, &compCode;, &reason;);
CheckCallResult("Add channel type", compCode, reason);

/*****
/* Add an inquiry for various attributes */
/*****
mqAddInquiry(adminBag, MQIACH_CHANNEL_TYPE, &compCode;, &reason;);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the channel names and channel types. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* MQ connection handle */
          MQCMD_INQUIRE_CHANNEL, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode;, /* Completion code from the mqexecute */
          &reason;); /* Reason code from mqexecute call */

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
  printf("Please start the command server: <strmqcsv QMgrName="">\n");
  goto MOD_EXIT;
}

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */

```

```

/*****
if ( compCode == MQCC_OK )          /* Successful mqExecute */
{
/*****
/* Count the number of system bags embedded in the response bag from the */
/* mqExecute call. The attributes for each channel are in separate bags. */
/*****
mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags;,
              &compCode;, &reason;);
CheckCallResult("Count number of bag handles", compCode, reason);

for ( i=0; i<numberOfBags; i++)
{
/*****
/* Get the next system bag handle out of the mqExecute response bag. */
/* This bag contains the channel attributes */
/*****
mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &AttrsBag,
              &compCode, &reason);
CheckCallResult("Get the result bag handle", compCode, reason);

/*****
/* Get the channel name out of the channel attributes bag */
/*****
mqInquireString(cAttrsBag, MQCACH_CHANNEL_NAME, 0, MQ_OBJECT_NAME_LENGTH,
                chlName, &chlNameLength, NULL, &compCode, &reason);
CheckCallResult("Get channel name", compCode, reason);

/*****
/* Get the channel type out of the channel attributes bag */
/*****
mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                 &compCode, &reason);
CheckCallResult("Get type", compCode, reason);

/*****
/* Use mqTrim to prepare the channel name for printing. */
/* Print the result. */
/*****
mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
WRITEOUTFILE(outfp,OutputBuffer,29)
}
}

else          /* Failed mqExecute */
{
printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
        compCode, reason);
/*****
/* If the command fails get the system bag handle out of the mqexecute */
/* response bag.This bag contains the reason from the command server */
/* why the command failed. */
/*****
if (reason == MQRCCF_COMMAND_FAILED)
{
mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
              &compCode, &reason);
CheckCallResult("Get the result bag handle", compCode, reason);

/*****
/* Get the completion code and reason code, returned by the command */
/* server, from the embedded error bag. */
/*****
mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                 &compCode, &reason );
CheckCallResult("Get the completion code from the result bag",
                compCode, reason);
mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                 &compCode, &reason);
CheckCallResult("Get the reason code from the result bag",
                compCode, reason);
printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
        mqExecuteCC, mqExecuteRC);
}
}
}

MOD_EXIT:
/*****
/* Delete the admin bag if successfully created. */
/*****

```



```

if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created.
*****/
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected
*****/
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
}

/*****
/* Close the output file if open
*****/
if(outfp != NULL)
    CLOSEOUTFILE(outfp);

return 0;
}

/*****
/*
/* Function: CheckCallResult
/*
*****/
/* Input Parameters:  Description of call
/*                    Completion code
/*                    Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
*****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %ld : Reason = %ld\n", callText,
            cc, rc);
}

```

C-Beispielprogramm für die Abfrage von Warteschlangen und Druckinformationen (amqsailq.c)

Das C-Beispielprogramm amqsailq.c fragt die aktuelle Tiefe der lokalen Warteschlangen mithilfe der MQAI ab.

```

/*****
/*
/* Program name: AMQSAILQ.C
/*
/* Description:  Sample C program to inquire the current depth of the local
/*              queues using the IBM MQ Administration Interface (MQAI)
/*
/* Statement:    Licensed Materials - Property of IBM
/*
/*              84H2000, 5765-B73
/*              84H2001, 5639-B42
/*              84H2002, 5765-B74
/*              84H2003, 5765-B75
/*              84H2004, 5639-B43

```

```

/*
/*          (C) Copyright IBM Corp. 1999, 2023.
/*
/*
/*****
/*
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/* bag.
/* These are:-
/* - The generic queue name "*"
/* - The type of queue required. In this sample we want to
/* inquire local queues.
/* - The attribute to be inquired. In this sample we want the
/* current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_Q command is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* - If the call is successful, the depth of each local queue is placed
/* in system bags embedded in the response bag of the mqExecute call.
/* The name and depth of each queue is obtained from each of the bags
/* and the result displayed on the screen.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
/*****

/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI
#include <cmqcfh.h>       /* PCF
#include <cmqbc.h>        /* MQAI

/*****
/* Function prototypes
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    /*****
    MQHCONN hConn;          /* handle to IBM MQ connection
    MQCHAR  qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name
    MQLONG  reason;         /* reason code
    MQLONG  connReason;    /* MQCONN reason code
    MQLONG  compCode;      /* completion code
    MQHBAG  adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute
    MQHBAG  responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute
    MQHBAG  qAttrsBag;    /* bag containing q attributes
    MQHBAG  errorBag;     /* bag containing cmd server error
    MQLONG  mqExecuteCC;  /* mqExecute completion code
    MQLONG  mqExecuteRC;  /* mqExecute reason code
    /*****

```

```

MQLONG qNameLength;          /* Actual length of q name          */
MQLONG qDepth;              /* depth of queue                  */
MQLONG i;                   /* loop counter                    */
MQLONG numberOfBags;        /* number of bags in response bag */
MQCHAR qName[MQ_Q_NAME_LENGTH+1]; /* name of queue extracted from bag*/

printf("Display current depths of local queues\n\n");

/*****
/* Connect to the queue manager          */
*****/
if (argc > 1)
    strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(qmName, &hConn, &compCode, &connReason);

/*****
/* Report the reason and stop if the connection failed.          */
*****/
if (compCode == MQCC_FAILED)
{
    CheckCallResult("Queue Manager connection", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Create an admin bag for the mqExecute call          */
*****/
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
CheckCallResult("Create admin bag", compCode, reason);
/*****
/* Create a response bag for the mqExecute call          */
*****/
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic queue name into the admin bag          */
*****/
mqAddString(adminBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode, &reason);
CheckCallResult("Add q name", compCode, reason);

/*****
/* Put the local queue type into the admin bag          */
*****/
mqAddInteger(adminBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type", compCode, reason);

/*****
/* Add an inquiry for current queue depths          */
*****/
mqAddInquiry(adminBag, MQIA_CURRENT_Q_DEPTH, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the local queue names and queue depths.          */
/* The mqExecute call creates the PCF structure required, sends it to          */
/* the command server, and receives the reply from the command server into          */
/* the response bag. The attributes are contained in system bags that are          */
/* embedded in the response bag, one set of attributes per bag.          */
*****/
mqExecute(hConn,          /* IBM MQ connection handle          */
          MQCMD_INQUIRE_Q, /* Command to be executed          */
          MQHB_NONE,      /* No options bag          */
          adminBag,      /* Handle to bag containing commands          */
          responseBag,   /* Handle to bag to receive the response          */
          MQHO_NONE,     /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE          */
          MQHO_NONE,     /* Create a dynamic q for the response          */
          &compCode,     /* Completion code from the mqExecute          */
          &reason);      /* Reason code from mqExecute call          */

/*****
/* Check the command server is started. If not exit.          */
*****/
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
    exit(98);
}

```

```

}

/*****
/* Check the result from mqExecute call. If successful find the current */
/* depths of all the local queues. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
/*****
/* Count the number of system bags embedded in the response bag from the */
/* mqExecute call. The attributes for each queue are in a separate bag. */
/*****
mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
&reason);
CheckCallResult("Count number of bag handles", compCode, reason);

for ( i=0; i<numberOfBags; i++)
{
/*****
/* Get the next system bag handle out of the mqExecute response bag. */
/* This bag contains the queue attributes */
/*****
mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &qAttrsBag, &compCode,
&reason);
CheckCallResult("Get the result bag handle", compCode, reason);

/*****
/* Get the queue name out of the queue attributes bag */
/*****
mqInquireString(qAttrsBag, MQCA_Q_NAME, 0, MQ_Q_NAME_LENGTH, qName,
&qNameLength, NULL, &compCode, &reason);
CheckCallResult("Get queue name", compCode, reason);

/*****
/* Get the depth out of the queue attributes bag */
/*****
mqInquireInteger(qAttrsBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
&compCode, &reason);
CheckCallResult("Get depth", compCode, reason);

/*****
/* Use mqTrim to prepare the queue name for printing. */
/* Print the result. */
/*****
mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason)
printf("%4d %-48s\n", qDepth, qName);
}
}
}

else /* Failed mqExecute */
{
printf("Call to get queue attributes failed: Completion Code = %d :
Reason = %d\n", compCode, reason);

/*****
/* If the command fails get the system bag handle out of the mqExecute */
/* response bag. This bag contains the reason from the command server */
/* why the command failed. */
/*****
if (reason == MQRCCF_COMMAND_FAILED)
{
mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag, &compCode,
&reason);
CheckCallResult("Get the result bag handle", compCode, reason);

/*****
/* Get the completion code and reason code, returned by the command */
/* server, from the embedded error bag. */
/*****
mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
&compCode, &reason );
CheckCallResult("Get the completion code from the result bag",
compCode, reason);
mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
&compCode, &reason);
CheckCallResult("Get the reason code from the result bag",
compCode, reason);
printf("Error returned by the command server: Completion Code = %d :
Reason = %d\n", mqExecuteCC, mqExecuteRC);
}
}
}

```

```

/*****
/* Delete the admin bag if successfully created. */
/*****
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from queue manager", compCode, reason);
}
return 0;
}

*****/
* Function: CheckCallResult */
* */
*****/
* Input Parameters: Description of call */
* Completion code */
* Reason code */
* */
* Output Parameters: None */
* */
* Logic: Display the description of the call, the completion code and the */
* reason code if the completion code is not successful */
* */
*****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

```

Datenbehälter und die MQAI

Ein Datenbehälter ist ein Mittel zur Handhabung von Eigenschaften oder Parametern von Objekten über die IBM MQ-Verwaltungsschnittstelle (MQAI).

Datenbehälter

- Der Datenbehälter enthält null oder mehr *Datenelemente*. Diese Datenelemente werden innerhalb des Beutels bestellt, wenn sie in die Tasche gesteckt werden. Dies wird als *Einfügeanordnung* bezeichnet. Jedes Datenelement enthält einen *Selektor*, der das Datenelement identifiziert, und einen *Wert* dieses Datenelements, das entweder eine ganze Zahl, eine 64-Bit-Ganzzahl, ein ganzzahliger Filter, eine Zeichenfolge, ein Zeichenfolgefilter, eine Bytefolge, ein Bytefolgefilter oder ein Handle eines anderen Beutels sein kann. Datenelemente werden in „Typen des in der MQAI verfügbaren Datenelements“ auf [Seite 64](#) detailliert beschrieben.

Es gibt zwei Typen von Selektoren: *Benutzerselektoren* und *Systemselektoren*. Diese werden in [MQAI Selectors](#) beschrieben. Die Selektoren sind in der Regel eindeutig, aber es ist möglich, mehrere Werte für denselben Selektor zu haben. In diesem Fall gibt ein *Index* das jeweilige Vorkommen des Selektors an, das erforderlich ist. Indizes werden unter „Indexierung in der MQAI“ auf [Seite 37](#) beschrieben.

Eine Hierarchie dieser Konzepte wird in Abbildung 1 dargestellt.

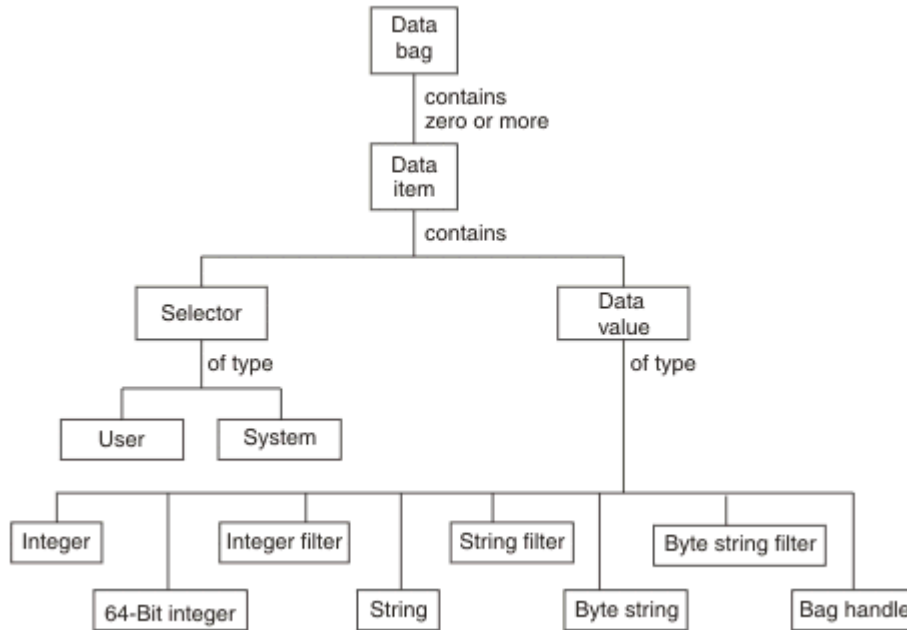


Abbildung 5. Hierarchie von MQAI-Konzepten

Die Hierarchie wurde in einem vorherigen Abschnitt erläutert.

Typen von Datenbehälter

Je nach Task, die Sie ausführen möchten, können Sie den Typ des Datenbeutels auswählen, den Sie erstellen möchten:

Benutzerbehälter

Eine einfache Tasche, die für Benutzerdaten verwendet wird.

Verwaltungstasche

Ein Behälter, der für Daten erstellt wurde, die zum Verwalten von IBM MQ-Objekten verwendet werden, indem Verwaltungsnachrichten an einen Befehlsserver gesendet werden. Der Verwaltungsbehälter schließt automatisch bestimmte Optionen ein, wie unter „Datenstaschen erstellen und löschen“ auf Seite 63 beschrieben wird.

Befehlsbehälter

Ein Behälter, der auch für Befehle zum Verwalten von IBM MQ-Objekten erstellt wurde. Im Gegensatz zum Verwaltungsbehälter impliziert der Befehlsbehälter jedoch nicht automatisch bestimmte Optionen, obwohl diese Optionen verfügbar sind. Weitere Informationen zu Optionen finden Sie unter „Datenstaschen erstellen und löschen“ auf Seite 63.

Gruppenbehälter

Ein Behälter, der zum Speichern einer Gruppe gruppierter Datenelemente verwendet wird. Gruppenbehälter können nicht für die Verwaltung von IBM MQ-Objekten verwendet werden.

Darüber hinaus wird die **Systemtasche** von der MQAI erstellt, wenn eine Antwortnachricht vom Befehlsserver zurückgegeben und in die Ausgabetasche eines Benutzers gestellt wird. Ein Systembehälter kann vom Benutzer nicht geändert werden.

Datenbeutel verwenden Die verschiedenen Möglichkeiten zur Verwendung von Datensäcken sind in diesem Thema aufgeführt:

Daten-Bags verwenden

Die verschiedenen Arten der Verwendung von Datensäcken werden in der folgenden Liste angezeigt:

- Sie können Datenbehälter erstellen und löschen „[Datenstaschen erstellen und löschen](#)“ auf Seite 63.
- Sie können mithilfe von Datenbehältern Daten zwischen Anwendungen versenden „[Datensäcke mit dem MQAI einschalten und empfangen](#)“ auf Seite 64.
- Sie können Datenbehältern Datenelemente hinzufügen „[Hinzufügen von Datenelementen zu Taschen mit der MQAI](#)“ auf Seite 65.
- Sie können innerhalb eines Datenbehälters einen Abfragebefehl hinzufügen „[Abfragebefehl zu einem Behälter hinzufügen](#)“ auf Seite 66.
- Sie können innerhalb von Datenbehältern Abfragen vornehmen „[Inquiring innerhalb von Datensäcken](#)“ auf Seite 66.
- Sie können Datenelemente innerhalb von Datenbehältern zählen „[Datenelemente zählen](#)“ auf Seite 69.
- Sie können Daten innerhalb eines Datenbehälters ändern „[Informationen in einer Tasche ändern](#)“ auf Seite 67.
- Sie können einen Datenbehälter leeren „[Inhalt einer Tasche mit dem mqClearBag-Aufruf löschen](#)“ auf Seite 68.
- Sie können einen Datenbehälter abschneiden „[Abschneiden eines Behälters mit dem mqTruncateBag-Aufruf](#)“ auf Seite 68.
- Sie können Behälter und Puffer konvertieren „[Konvertieren von Taschen und Puffern](#)“ auf Seite 68.

Datenstaschen erstellen und löschen

Datenstaschen erstellen

Wenn Sie die MQAI verwenden möchten, erstellen Sie zuerst einen Datenbehälter mit dem mqCreateBag-Aufruf. Als Eingabe für diesen Aufruf geben Sie eine oder mehrere Optionen an, um die Erstellung der Tasche zu steuern.

Mit dem Parameter **Options** des MQCreateBag-Aufrufs können Sie auswählen, ob eine Benutzergruppe, eine Befehlstasche, ein Gruppenbehälter oder ein Verwaltungsbehälter erstellt werden soll.

Zum Erstellen eines Benutzerbeutels, einer Befehlstasche oder eines Gruppenbeutels können Sie eine oder mehrere weitere Optionen auswählen, um:

- Verwenden Sie das Listenformular, wenn zwei oder mehr benachbarte Vorkommen desselben Selektors in einer Tasche vorhanden sind.
- Sie können die Datenelemente neu anordnen, wenn sie einer PCF-Nachricht hinzugefügt werden, um sicherzustellen, dass die Parameter in der richtigen Reihenfolge sind. Weitere Informationen zu Datenelementen finden Sie im Abschnitt „[Typen des in der MQAI verfügbaren Datenelements](#)“ auf Seite 64.
- Überprüfen Sie die Werte von Benutzerselektoren für Elemente, die Sie der Tasche hinzufügen.

Verwaltungssäcke implizieren automatisch diese Optionen.

Ein Datenbehälter wird durch den Griff identifiziert. Die Kennung des Sackes wird von mqCreateBag zurückgegeben und muss bei allen anderen Aufrufen, die den Datenbehälter verwenden, bereitgestellt werden.

Eine vollständige Beschreibung des mqCreateBag-Aufrufs finden Sie unter [mqCreateBag](#).

Datentaschen löschen

Alle Datenbehälter, die vom Benutzer erstellt werden, müssen auch mit dem mqDeleteBag-Aufruf gelöscht werden. Wenn z. B. eine Tasche im Benutzercode erstellt wird, muss sie auch im Benutzercode gelöscht werden.

Systemtaschen werden automatisch von der MQAI erstellt und gelöscht. Weitere Informationen hierzu finden Sie unter „Verwaltungsbefehle mit dem mqExecute-Aufruf an den Befehl 'qm' senden“ auf Seite 70. Der Benutzercode kann eine Systemhülle nicht löschen.

Eine vollständige Beschreibung des Aufrufs 'mqDeleteBag' finden Sie unter mqDeleteBag.

Datensäcke mit dem MQAI einschalten und empfangen

Daten können auch zwischen Anwendungen gesendet werden, indem Datensäcke mithilfe der mqPutBag- und mqGetBag-Aufrufe gestellt und erhalten werden. Auf diese Weise kann die IBM MQ-Verwaltungsschnittstelle (MQAI) den Puffer und nicht die Anwendung verarbeiten.

Der mqPutBag-Aufruf konvertiert den Inhalt des angegebenen Sacks in eine PCF-Nachricht und sendet die Nachricht an die angegebene Warteschlange, und der mqGetBag-Aufruf entfernt die Nachricht aus der angegebenen Warteschlange und konvertiert sie zurück in einen Datenbehälter. Daher ist der mqPutBag-Aufruf das Äquivalent des Aufrufs mqBagToBuffer, gefolgt von MQPUT, und der mqGetBag entspricht dem MQGET-Aufruf gefolgt von mqBufferToBag.

Weitere Informationen zum Senden und Empfangen von PCF-Nachrichten in einer bestimmten Warteschlange finden Sie in „PCF-Nachrichten in einer angegebenen Warteschlange senden und empfangen“ auf Seite 25.

Anmerkung: Wenn Sie sich für die Verwendung des mqGetBag-Aufrufs entscheiden, müssen die PCF-Details in der Nachricht korrekt sein. Wenn dies nicht der Fall ist, werden die entsprechenden Fehlerergebnisse angezeigt und die PCF-Nachricht wird nicht zurückgegeben.

Typen des in der MQAI verfügbaren Datenelements

Datenelemente werden von der IBM MQ-Verwaltungsschnittstelle (MQAI) verwendet, um Datenbehälter zu füllen, wenn sie erstellt werden. Bei diesen Datenelementen kann es sich um Benutzer- oder Systemelemente handeln.

Diese Benutzerelemente enthalten Benutzerdaten, wie z. B. Attribute von Objekten, die verwaltet werden. Systemelemente sollten für mehr Kontrolle über die generierten Nachrichten verwendet werden: z. B. die Generierung von Nachrichtenheadern. Weitere Informationen zu Systemelementen finden Sie im Abschnitt „Systemelemente und MQAI“ auf Seite 65.

Typen von Datenelementen

Wenn Sie einen Datenbehälter erstellt haben, können Sie ihn mit Ganzzahl- oder Zeichenfolgeelementen füllen. Sie können alle drei Typen von Elementen abgefragt werden.

Das Datenelement kann eine ganze Zahl oder ein Zeichenfolgeelement sein. Im Folgenden sind die Typen der Datenelemente in der MQAI verfügbar:

- Integer
- 64-Bit-Ganzzahl
- Filter für Ganzzahl
- Zeichenfolge
- Zeichenfolgefilter
- Bytefolge
- Filter für Bytefolge
- Behälterkennung

Datenelemente verwenden

Dies sind die folgenden Arten der Verwendung von Datenelementen:

- „Datenelemente zählen“ auf Seite 69.
- „Datenelemente löschen“ auf Seite 69.
- „Hinzufügen von Datenelementen zu Taschen mit der MQAI“ auf Seite 65.

- „Datenelemente filtern und abfragen“ auf Seite 66.

Systemelemente und MQAI

Systemelemente können von der IBM MQ-Verwaltungsschnittstelle (MQAI) für Folgendes verwendet werden:

- Die Generierung von PCF-Headern. Systemelemente können die PCF-Befehls-ID, die Steueroptionen, die Nachrichtenfolgennummer und den Befehlstyp steuern.
- Datenkonvertierung. Systemelemente behandeln die Zeichensatzes-ID für die Zeichenfolge-Elemente in der Tasche.

Wie alle Datenelemente bestehen Systemelemente aus einem Selektor und einem Wert. Weitere Informationen zu diesen Selektoren und zu ihrer Verwendung finden Sie unter [MQAI-Selektoren](#).

Systemelemente sind eindeutig. Ein oder mehrere Systemelemente können durch einen Systemselektor identifiziert werden. Es gibt nur ein Vorkommen jedes Systemselektors.

Die meisten Systemelemente können geändert werden (siehe „Informationen in einer Tasche ändern“ auf Seite 67), aber die Optionen zur Erstellung von Behältern können vom Benutzer nicht geändert werden. Systemelemente können nicht gelöscht werden. (Siehe „Datenelemente löschen“ auf Seite 69.)

Hinzufügen von Datenelementen zu Taschen mit der MQAI

Wenn ein Datenbehälter unter Verwendung der IBM MQ-Verwaltungsschnittstelle (MQAI) erstellt wird, können Sie ihn mit Datenelementen füllen. Bei diesen Datenelementen kann es sich um Benutzer- oder Systemelemente handeln.

Weitere Informationen zu Datenelementen finden Sie im Abschnitt „Typen des in der MQAI verfügbaren Datenelements“ auf Seite 64.

Mit der MQAI können Sie ganze Zahlen, 64-Bit-Ganzzahlelemente, Ganzzahlfilterelemente, Zeichenfolgeelemente, Zeichenfolgefilter, Bytefolgeelemente und Bytefolgefilterelemente zu Behältern hinzufügen. Dies wird in [Abbildung 6](#) auf Seite 65 gezeigt. Die Elemente werden durch einen Selektor identifiziert. Gewöhnlich identifiziert ein Selektor nur ein Element, aber dies ist nicht immer der Fall. Wenn ein Datenelement mit dem angegebenen Selektor bereits in der Tasche vorhanden ist, wird dem Ende der Tasche eine zusätzliche Instanz dieses Selektors hinzugefügt.



Abbildung 6. Datenelemente hinzufügen

Fügen Sie mit den `mqAdd *`-Aufrufen Datenelemente zu einer Tasche hinzu:

- Um ganzzahlige Elemente hinzuzufügen, verwenden Sie den `mqAddInteger`-Aufruf wie in [mqAddInteger](#) beschrieben.
- Um 64-Bit-Integer-Elemente hinzuzufügen, verwenden Sie den `mqAddInteger64`-Aufruf wie in [mqAddInteger64](#) beschrieben.
- Um ganzzahlige Filterelemente hinzuzufügen, verwenden Sie den `mqAddIntegerFilter`-Aufruf wie in [mqAddIntegerFilter](#) beschrieben.
- Um Zeichenfolgeelemente hinzuzufügen, verwenden Sie den `mqAddString`-Aufruf wie in [mqAddString](#) beschrieben.

- Um Zeichenfolgefilterelemente hinzuzufügen, verwenden Sie den `mqAddStringFilter`-Aufruf wie in [mqAddStringFilter](#) beschrieben.
- Um Bytefolgeelemente hinzuzufügen, verwenden Sie den `mqAddByteString`-Aufruf wie in [mqAddByteString](#) beschrieben.
- Um Bytefolgefilterelemente hinzuzufügen, verwenden Sie den `mqAddByteStringFilter`-Aufruf wie in [mqAddByteStringFilter](#) beschrieben.

Weitere Informationen zum Hinzufügen von Datenelementen zu einem Behälter finden Sie im Abschnitt „Systemelemente und MQAI“ auf Seite 65.

Abfragebefehl zu einem Behälter hinzufügen

Der Aufruf 'mqAddInquiry' wird verwendet, um einen Abfragebefehl zu einem Behälter hinzuzufügen. Der Aufruf ist speziell für Verwaltungszwecke, so dass er nur mit Verwaltungsbehältern verwendet werden kann. Sie können die Selektoren von Attributen angeben, für die Sie Abfragen von IBM MQ durchführen möchten.

Eine vollständige Beschreibung des Aufrufs 'mqAddInquiry' finden Sie unter [mqAddInquiry](#).

Datenelemente filtern und abfragen

Wenn Sie die MQAI verwenden, um die Attribute von IBM MQ-Objekten abzufragen, können Sie die Daten, die an Ihr Programm zurückgegeben werden, auf zwei Arten steuern.

- Sie können die Daten, die mit den Aufrufen 'mqAddInteger' und 'mqAddString' zurückgegeben werden, **filter**. Mit dieser Methode können Sie ein Paar aus *Selector* und *ItemValue* angeben, z. B.:

```
mqAddInteger(inputbag, MQIA_Q_TYPE, MQQT_LOCAL)
```

In diesem Beispiel wird angegeben, dass der Warteschlangentyp (*Selector*) lokal (*ItemValue*) sein muss und dass diese Spezifikation mit den Attributen des Objekts (in diesem Fall eine Warteschlange) übereinstimmen muss, über das/die Sie abgefragt werden.

Andere Attribute, die gefiltert werden können, entsprechen den PCF Inquire*-Befehlen, die unter „Einführung in IBM MQ Programmable Command Formats“ auf Seite 22 beschrieben sind. Wenn Sie beispielsweise die Attribute eines Kanals abgefragt haben, lesen Sie den Befehl Inquire Channel in dieser Produktdokumentation. Die Parameter "Required parameters" und "Optional parameters" des Befehls "Inquire Channel" geben die Selektoren an, die Sie zum Filtern verwenden können.

- Mithilfe des Aufrufs 'mqAddInquiry' können Sie bestimmte Attribute eines Objekts **abfragen**. Gibt den Selektor an, an dem Sie interessiert sind. Wenn Sie den Selektor nicht angeben, werden alle Attribute des Objekts zurückgegeben.

Im Folgenden finden Sie ein Beispiel für das Filtern und Abfragen der Attribute einer Warteschlange:

```
/* Request information about all queues */
mqAddString(adminbag, MQCA_Q_NAME, "*")

/* Filter attributes so that local queues only are returned */
mqAddInteger(adminbag, MQIA_Q_TYPE, MQQT_LOCAL)

/* Query the names and current depths of the local queues */
mqAddInquiry(adminbag, MQCA_Q_NAME)
mqAddInquiry(adminbag, MQIA_CURRENT_Q_DEPTH)

/* Send inquiry to the command server and wait for reply */
mqExecute(MQCMD_INQUIRE_Q, ...)
```

Inquiring innerhalb von Datensäcken

Sie können Folgendes einfragen:

- Der Wert eines ganzzahligen Elements mit Hilfe des Aufrufs 'mqInquireInteger'. Siehe [mqInquireInteger](#).

- Der Wert eines ganzzahligen 64-Bit-Elements mit dem Aufruf 'mqInquireInteger64'. Siehe [mqInquireInteger64](#).
- Der Wert eines Ganzzahlfilterelements mit dem Aufruf 'mqInquireIntegerFilter'. Siehe [mqInquireIntegerFilter](#).
- Der Wert eines Zeichenfolgeelements, das mit dem Aufruf 'mqInquireString' verwendet wird. Siehe [mqInquireString](#).
- Der Wert eines Zeichenfolgefilterelements mit dem Aufruf 'mqInquireStringFilter'. Siehe [mqInquireStringFilter](#).
- Der Wert eines Bytefolgeelements mit dem Aufruf 'mqInquireByteString'. Siehe [mqInquireByteString](#).
- Der Wert eines Bytefolgefilterelements mit Hilfe des Aufrufs 'mqInquireByteStringFilter'. Siehe [mqInquireByteStringFilter](#).
- Der Wert einer Taschenkennung, die mit dem mqInquireBag-Aufruf verwendet wird. Siehe [mqInquireBag](#).

Sie können auch den Typ (Ganzzahl, 64-Bit-Integer, Ganzzahlfilter, Zeichenfolge, Zeichenfolgefilter, Bytefolge, Bytefolgefilter oder Taschengriff) eines bestimmten Elements mit dem Aufruf mqInquireItemInfo abgefragt werden. Siehe [mqInquireItemInfo](#).

Informationen in einer Tasche ändern

Mit dem MQAI können Sie Informationen innerhalb einer Tasche mit Hilfe der mqSet *-Aufrufe ändern. Sie haben folgende Möglichkeiten:

1. Datenelemente in einer Tasche ändern. Der Index ermöglicht das Ersetzen einer einzelnen Instanz eines Parameters, indem das Vorkommen des zu änderbaren Elements angegeben wird (siehe [Abbildung 7](#) auf Seite 67).

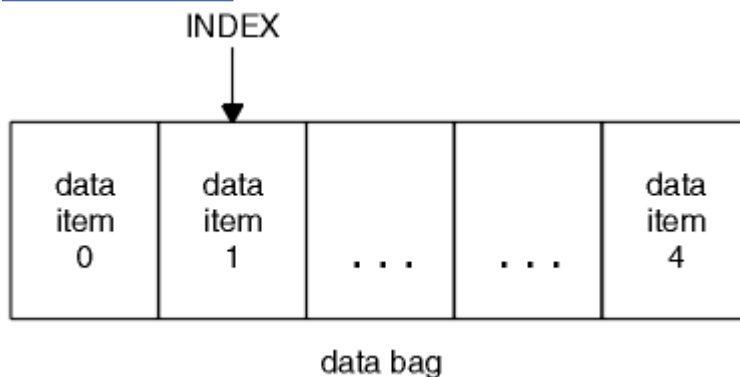


Abbildung 7. Ein einzelnes Datenelement ändern

2. Löschen Sie alle vorhandenen Vorkommen des angegebenen Selektors und fügen Sie dem Ende des Sacks ein neues Vorkommen hinzu. (Siehe [Abbildung 8](#) auf Seite 67.) Ein spezieller Indexwert ermöglicht **alle** Instanzen eines Parameters, der ersetzt werden soll.

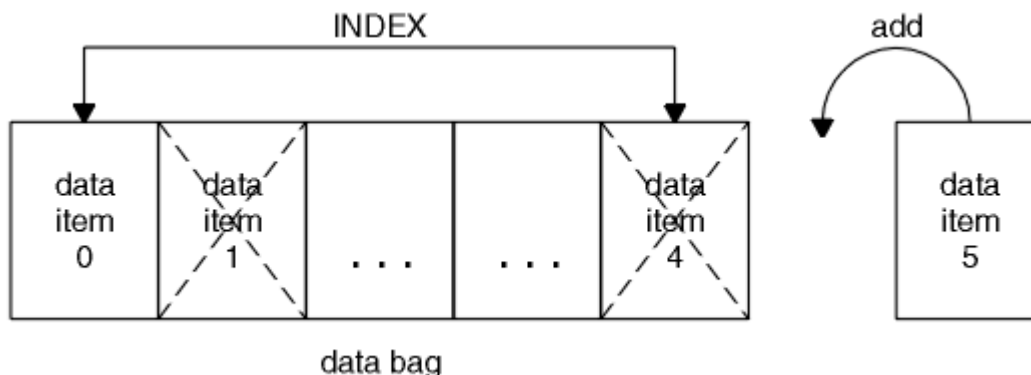


Abbildung 8. Alle Datenelemente ändern

Anmerkung: Der Index erfüllt die Einfügefølge innerhalb des Sacks, kann aber die Indizes anderer Datenelemente beeinflussen.

Mit dem `mqSetInteger`-Aufruf können Sie Integer-Elemente in einer Tasche ändern. Mit dem Aufruf `mqSetInteger64` können Sie 64-Bit-Integer-Elemente ändern. Mit dem `mqSetIntegerFilter`-Aufruf können Sie ganzzahlige Filterelemente ändern. Mit dem `mqSetString`-Aufruf können Sie Zeichenfolgeelemente ändern. Mit dem `mqSetStringFilter`-Aufruf können Sie Zeichenfolgefilterelemente ändern. Mit dem `mqSetByteString`-Aufruf können Sie Bytefolgeelemente ändern. Mit dem Aufruf '`mqSetByteStringFilter`' können Sie Bytefolgefilterelemente ändern. Alternativ können Sie diese Aufrufe verwenden, um alle vorhandenen Vorkommen des angegebenen Selektors zu löschen und am Ende der Tasche ein neues Vorkommen hinzuzufügen. Bei dem Datenelement kann es sich um ein Benutzerelement oder um ein Systemelement handeln.

Eine vollständige Beschreibung dieser Aufrufe finden Sie unter:

- [mqSetInteger](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetString](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringFilter](#)

Inhalt einer Tasche mit dem `mqClearBag`-Aufruf löschen

Mit dem `mqClearBag`-Aufruf werden alle Benutzerelemente aus einem Benutzerbehälter entfernt und die Systemelemente werden auf ihre Anfangswerte zurückversetzt. Die in der Tasche enthaltenen Systemtaschen werden ebenfalls gelöscht.

Eine vollständige Beschreibung des `mqClearBag`-Aufrufs finden Sie unter [mqClearBag](#).

Abschneiden eines Behälters mit dem `mqTruncateBag`-Aufruf

Der `mqTruncateBag`-Aufruf reduziert die Anzahl der Benutzerelemente in einem Benutzerbehälter, indem er die Elemente vom Ende der Tasche aus löscht, beginnend mit dem zuletzt hinzugefügten Element. Es kann beispielsweise verwendet werden, wenn dieselben Headerinformationen verwendet werden, um mehr als eine Nachricht zu generieren.

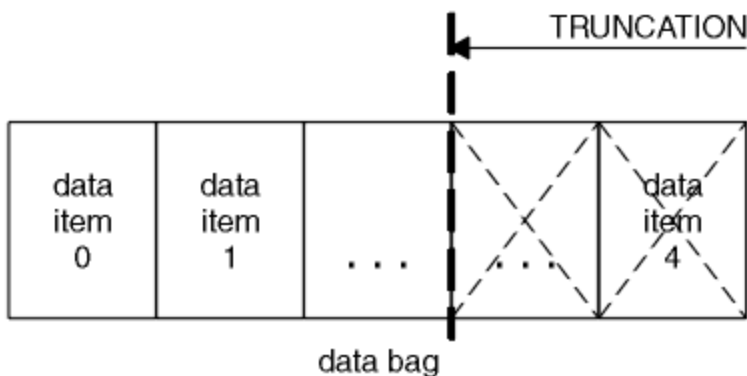


Abbildung 9. Einen Behälter abschneiden

Eine vollständige Beschreibung des `mqTruncateBag`-Aufrufs finden Sie unter [mqTruncateBag](#).

Konvertieren von Taschen und Puffern

Um Daten zwischen Anwendungen zu senden, werden zuerst die Nachrichtendaten in eine Tasche gestellt. Anschließend werden die Daten in der Tasche mit Hilfe des `mqBagToBuffer`-Aufrufs in eine PCF-Nachricht konvertiert. Die PCF-Nachricht wird unter Verwendung des `MQPUT`-Aufrufs an die erforderliche

Warteschlange gesendet. Dies wird in [Abbildung 10 auf Seite 69](#) gezeigt. Eine vollständige Beschreibung des Aufrufs 'mqBagToBuffer' finden Sie in [mqBagToBuffer](#).

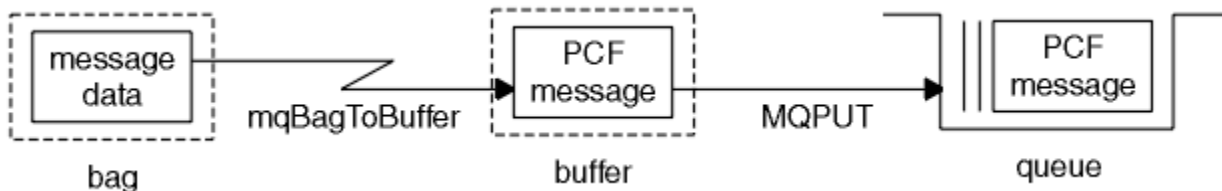


Abbildung 10. Konvertieren von Taschen in PCF-Nachrichten

Um Daten zu empfangen, wird die Nachricht mit dem MQGET-Aufruf in einen Puffer empfangen. Die Daten im Puffer werden dann mit Hilfe des mqBufferToBag-Aufrufs in eine Tasche konvertiert, so dass der Puffer eine gültige PCF-Nachricht enthält. Dies wird in [Abbildung 11 auf Seite 69](#) gezeigt. Eine vollständige Beschreibung des mqBufferToBag-Aufrufs finden Sie unter [mqBufferToBag](#).

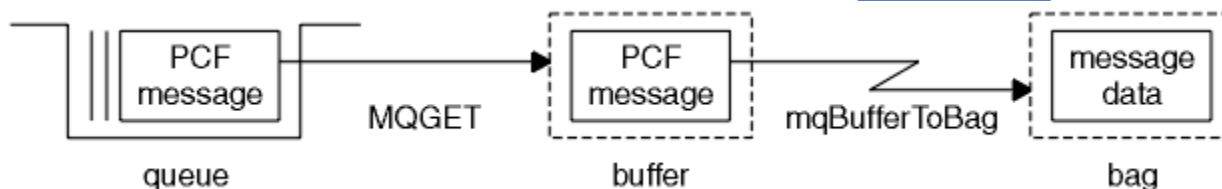


Abbildung 11. PCF-Nachrichten in Beutelformular umwandeln

Datenelemente zählen

Der Aufruf 'mqCountItems' zählt die Anzahl der Benutzerelemente, Systemelemente oder beides, die in einem Datenbehälter gespeichert sind, und gibt diese Zahl zurück. Beispiel: `mqCountItems(Bag, 7, ...)` gibt die Anzahl der Elemente im Behälter mit einem Selektor von 7 zurück. Sie kann Elemente durch einzelne Selektoren, durch Benutzerelektoren, durch Systemselektoren oder durch alle Selektoren zählen.

Anmerkung: Dieser Aufruf zählt die Anzahl der Datenelemente, nicht die Anzahl der eindeutigen Selektoren in der Tasche. Ein Selektor kann mehrfach vorkommen, so dass es in der Tasche weniger eindeutige Selektoren geben kann als Datenelemente.

Eine vollständige Beschreibung des Aufrufs 'mqCountItems' finden Sie unter [mqCountItems](#).

Datenelemente löschen

Sie können Artikel aus Taschen auf verschiedene Arten löschen. Sie haben folgende Möglichkeiten:

- Entfernt ein oder mehrere Benutzerelemente aus einer Tasche. Ausführliche Informationen hierzu finden Sie unter [„Datenelemente aus einer Tasche mit dem mqDeleteItem-Aufruf löschen“](#) auf Seite 69.
- Löschen Sie **alle** -Benutzerelemente aus einer Tasche, d. a. *clear*, eine Tasche. Weitere Informationen finden Sie unter [„Inhalt einer Tasche mit dem mqClearBag-Aufruf löschen“](#) auf Seite 68.
- Löschen Sie die Benutzereinträge am Ende eines Sacks, d. a. *Abschneiden* einer Tasche. Ausführliche Informationen hierzu finden Sie unter [„Abschneiden eines Behälters mit dem mqTruncateBag-Aufruf“](#) auf Seite 68.

Datenelemente aus einer Tasche mit dem mqDeleteItem-Aufruf löschen

Mit dem mqDeleteItem-Aufruf werden ein oder mehrere Benutzerelemente aus einer Tasche entfernt. Der Index wird zum Löschen verwendet:

1. Ein einzelnes Vorkommen des angegebenen Selektors. (Siehe [Abbildung 12 auf Seite 70](#).)

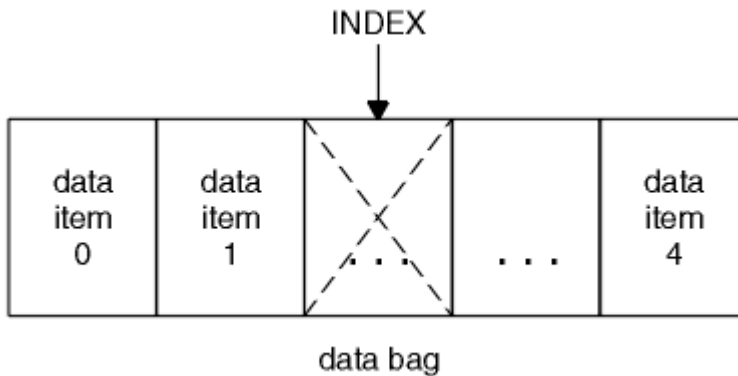


Abbildung 12. Löschen eines einzelnen Datenelements

oder

2. Alle Vorkommen des angegebenen Selektors. (Siehe [Abbildung 13](#) auf Seite 70.)

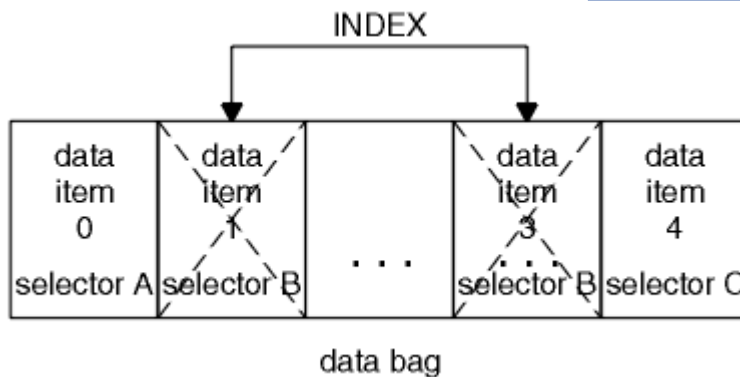


Abbildung 13. Alle Datenelemente löschen

Anmerkung: Der Index erfüllt die Einfügefolge innerhalb des Sacks, kann aber die Indizes anderer Datenelemente beeinflussen. Der Aufruf 'mqDeleteItem' beispielsweise behält die Indexwerte der Datenelemente, die dem gelöschten Element folgen, nicht bei, da die Indizes neu organisiert werden, um die Lücke zu füllen, die vom gelöschten Element übrig bleibt.

Eine vollständige Beschreibung des Aufrufs 'mqDeleteItem' finden Sie unter [mqDeleteItem](#).

Verwaltungsbefehle mit dem mqExecute-Aufruf an den Befehl 'qm' senden

Wenn ein Datenbehälter erstellt und gefüllt wurde, kann eine Verwaltungsbefehlsnachricht mit Hilfe des Aufrufs mqExecute an den Befehlsserver eines Warteschlangenmanagers gesendet werden. Dadurch wird der Austausch mit dem Befehlsserver durchgeführt und die Antworten in einem Behälter zurückgegeben.

Nachdem Sie Ihren Datenbehälter erstellt und gefüllt haben, können Sie eine Verwaltungsbefehlsnachricht an den Befehlsserver eines Warteschlangenmanagers senden. Dies ist der einfachste Weg, indem Sie den Aufruf mqExecute verwenden. Der mqExecute-Aufruf sendet eine Verwaltungsbefehlsnachricht als nicht persistente Nachricht und wartet auf alle Antworten. Antworten werden in einem Antwortbehälter zurückgegeben. Es sind möglicherweise Informationen zu Attributen enthalten, die sich beispielsweise auf mehrere IBM MQ-Objekte oder eine Reihe von PCF-Fehlerantwortnachricht beziehen. Daher kann der Antwortbehälter nur einen Rückkehrcode enthalten oder er könnte *verschachtelte Taschen* enthalten.

Antwortnachrichten werden in Systemtaschen gestellt, die vom System erstellt werden. Für Rückfragen zu den Namen von Objekten wird beispielsweise ein Systembehälter erstellt, in dem diese Objektnamen enthalten sind, und die Tasche wird in die Benutzertasche eingefügt. Handles zu diesen Taschen werden dann in die Antworttasche eingefügt und die verschachtelte Tasche kann von der Selektor MQHA_BAG_HANDLE aufgerufen werden. Der Systembehälter verbleibt im Speicher, wenn er nicht gelöscht wird, bis der Antwortbehälter gelöscht wird.

Das Konzept der *Verschachtelung* wird in [Abbildung 14](#) auf Seite 71 gezeigt.

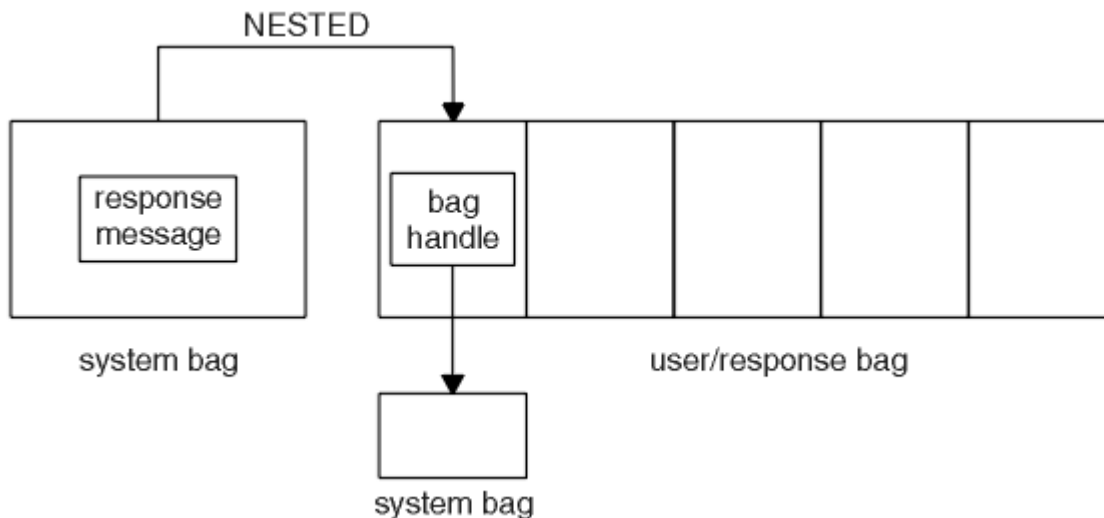


Abbildung 14. Verschachteln

Als Eingabe für den mqExecute-Aufruf müssen Sie Folgendes angeben:

- Eine MQI-Verbindungskennung.
- Der Befehl, der ausgeführt werden soll. Hierbei sollte es sich um einen der MQCMD_* -Werte handeln.
Anmerkung: Wenn dieser Wert von der MQAI nicht erkannt wird, wird der Wert trotzdem akzeptiert. Wenn jedoch der mqAddInquiry-Aufruf zum Einfügen von Werten in die Tasche verwendet wurde, muss dieser Parameter ein INQUIRE-Befehl sein, der von der MQAI erkannt wird. Das heißt, der Parameter muss das Format MQCMD_INQUIRE_* haben.
- Optional ein Handle der Tasche, die Optionen enthält, die die Verarbeitung des Aufrufs steuern. Hier können Sie auch die maximale Zeit in Millisekunden angeben, die der MQAI auf jede Antwortnachricht warten soll.
- Ein Handle des Verwaltungsbeutels, der Details zum Verwaltungsbefehl enthält, der ausgegeben werden soll.
- Ein Handle des Antwortbehälters, der die Antwortnachrichten empfängt.

Die folgenden Kennungen sind optional:

- Eine Objektkennung der Warteschlange, in die der Verwaltungsbefehl gestellt werden soll.
Wenn keine Objektkennung angegeben ist, wird der Verwaltungsbefehl in die Warteschlange SYSTEM.ADMIN.COMMAND.QUEUE des derzeit verbundenen Warteschlangenmanagers gestellt. Dies ist die Standardeinstellung.
- Eine Objektkennung der Warteschlange, in die Antwortnachrichten gestellt werden sollen.

Sie können auswählen, dass die Antwortnachrichten in eine dynamische Warteschlange, die automatisch von der MQAI erstellt wird, platziert werden sollen. Die erstellte Warteschlange ist nur für die Dauer des Aufrufs vorhanden und wird durch die MQAI beim Verlassen des mqExecute-Aufrufs gelöscht.

Beispiele für die Verwendung des mqExecute-Aufrufs finden Sie im Abschnitt [Beispielcode](#).

V 9.0.1 Verwaltung über die REST API

Sie können mit der administrative REST API IBM MQ-Objekte wie beispielsweise Warteschlangenmanager und Warteschlangen [V 9.0.5](#) sowie Managed File Transfer GET REST-Services verwalten. Informationen werden im JSON-Format an die administrative REST API gesendet und von ihr empfangen. Mithilfe dieser REST-konformen APIs können Sie die IBM MQ-Verwaltung in gängige DevOps und Automatisierungstools einbetten.

Vorbereitende Schritte

Referenzinformationen zu den verfügbaren REST-Ressourcen finden Sie im Abschnitt [Die administrative REST API-Referenz](#).

Anmerkung:

In IBM MQ 9.0.1 ist die administrative REST API nicht in die IBM MQ-Sicherheit integriert. Deshalb ist die administrative REST API standardmäßig inaktiviert. Sie müssen die administrative REST API manuell aktivieren, damit Sie sie verwenden können. Weitere Informationen zur Aktivierung der the administrative REST API finden Sie im Abschnitt [„Erste Schritte mit der administrative REST API“](#) auf Seite 72.

V 9.0.2 Ab IBM MQ 9.0.2 ist die administrative REST API in die IBM MQ-Sicherheit integriert. Die administrative REST API ist standardmäßig aktiviert. Sie müssen jedoch die Sicherheit konfigurieren, bevor Sie die administrative REST API verwenden können. Weitere Informationen hierzu finden Sie im Abschnitt [IBM MQ Console und REST API - Sicherheit](#).

Prozedur

- [„Erste Schritte mit der administrative REST API“](#) auf Seite 72
- [„administrative REST API verwenden“](#) auf Seite 78
- [„Fernverwaltung über die REST API“](#) auf Seite 84
- [„REST API-Zeitmarken“](#) auf Seite 88
- [„REST API-Fehlerbehandlung“](#) auf Seite 88
- [„REST API-Erkennung“](#) auf Seite 91
- [„Unterstützung landessprachunterstützter Sprachen in REST API“](#) auf Seite 92

V 9.0.1 Erste Schritte mit der administrative REST API

Damit Sie die administrative REST API starten können, müssen Sie zunächst die richtigen Komponenten installieren, die REST API aktivieren, die Sicherheit konfigurieren, und den mqweb-Server starten.

Vorbereitende Schritte

IBM i Unter IBM i sollten die Befehle in QSHLL ausgeführt werden.

Informationen zu diesem Vorgang

Die Prozedur für diese Task konzentriert sich auf den schnellen Einstieg in die administrative REST API. Die Schritte zum Konfigurieren der Sicherheit beschreiben, wie eine Basisbenutzerregistry konfiguriert wird, aber andere Optionen für die Konfiguration von Benutzern und Rollen vorhanden sind. Weitere Informationen zur Konfiguration der Sicherheit für die administrative REST API finden Sie im Abschnitt [IBM MQ Console und REST API-Sicherheit](#).

Anmerkung: Sie müssen ein [privilegierter Benutzer](#) sein, um auf die Datei `mqwebuser.xml` zugreifen zu können.

Vorgehensweise

1. Installieren Sie die Komponente IBM MQ Console und REST API:

- **V 9.0.4** **AIX** Installieren Sie unter AIX die Dateigruppe `mqm.web.rte`.
- **Linux** Installieren Sie unter Linux die Komponente `MQSeriesWeb`. Weitere Informationen zur Installation von Komponenten und Features unter Linux finden Sie im Abschnitt [Linux-Installationsaufgaben](#).

- **Windows** Installieren Sie unter Windows das Feature Web Administration. Weitere Informationen zur Installation von Komponenten und Features unter Windows finden Sie im Abschnitt [Windows-Installationsaufgaben](#).
 - **z/OS** Installieren Sie unter z/OS das Feature IBM MQ for z/OS Unix System Services Web Components. Weitere Informationen zur Installation von Komponenten und Features unter z/OS finden Sie im Abschnitt [z/OS-Installationsaufgaben](#).
2. Optional: Ab IBM MQ 9.0.2 ist die administrative REST API standardmäßig aktiviert. In IBM MQ 9.0.1 ist die administrative REST API jedoch standardmäßig inaktiviert. Wenn Sie IBM MQ 9.0.1 verwenden, aktivieren Sie die REST API, indem Sie die folgenden XML-Code in der Datei `mqwebuser.xml` innerhalb der `<server>`-Tags hinzufügen:

```
<variable name="mqRestAutostart" value="true"/>
```

Der `mqwebuser.xml` kann in einem der folgenden Verzeichnisse gefunden werden:

- **ULW** `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** `WLP_user_directory/servers/mqweb`

Dabei ist `WLP_user_directory` das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

3. **V 9.0.2**

Ab IBM MQ 9.0.2 ist die Sicherheitsfunktion für die administrative REST API aktiviert. Sie müssen Benutzer und Rollen konfigurieren, bevor Sie die administrative REST API verwenden können:

- a) Kopieren Sie die Datei `basic_registry.xml` aus dem Verzeichnis `MQ_INSTALLATION_PATH/web/mq/samp/configuration`.
- b) Stellen Sie die XML-Beispieldatei in das zutreffende Verzeichnis:

- **ULW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht `WLP-Benutzerverzeichnis` für das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

- c) Benennen Sie die XML-Beispieldatei in `mqwebuser.xml` um.

Anmerkung: Diese umbenannte Datei ersetzt eine vorhandene Datei, die auch für die IBM MQ Console verwendet wird. Kopieren Sie deshalb, falls Sie die Datei `mqwebuser.xml` für die IBM MQ Console geändert haben, Ihre Änderungen in die neue XML-Datei, bevor Sie sie umbenennen.

- d) Optional: Bearbeiten Sie die Datei `mqwebuser.xml`, um Benutzer und Gruppen hinzuzufügen. Weisen Sie diesen Benutzern und Gruppen die entsprechenden Rollen zu, damit sie berechtigt sind, die administrative REST API zu verwenden. Sie können auch die Kennwörter für die Benutzer ändern, die standardmäßig definiert sind, und die neuen Kennwörter codieren. Weitere Informationen hierzu finden Sie im Abschnitt [Benutzer und Rollen konfigurieren](#).

4. Fernen Verbindungen zum `mqweb`-Server aktivieren:

- Verwenden Sie ab Version 9.0.4 den Befehl `setmqweb` :
`setmqweb properties -k httpHost -v hostname`
- Fügen Sie ab Version 9.0.1 folgenden XML-Code in der Datei `mqwebuser.xml` innerhalb der `<server>`-Tags hinzu:

```
<variable name="httpHost" value="hostname"/>
```

Dabei gibt `hostname` die IP-Adresse, den Hostnamen des Domänennamensservers (DNS) mit dem Domänennamenssuffix oder den DNS-Hostnamen des Servers an, auf dem IBM MQ installiert ist. Verwenden Sie einen Stern (*), um alle verfügbaren Netzschnittstellen anzugeben.



Achtung: z/OS V 9.0.4

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist `WLP_user_directory` der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

5. Starten Sie den MQWeb-Server, der die REST API unterstützt:

- Windows Linux Geben Sie unter Windows und Linux als privilegierter Benutzer folgenden Befehl in der Befehlszeile ein:
`strmqweb`
- z/OS Starten Sie unter z/OS die Prozedur, die Sie in [Task 29: Create a procedure for the IBM WLP server](#) erstellt haben.

Nächste Schritte

1. V 9.0.2 Wählen Sie aus, wie Benutzer der administrative REST API auf dem mqweb-Server authentifiziert werden sollen. Sie müssen nicht für alle Benutzer die gleiche Methode verwenden. Die folgenden Optionen sind verfügbar:
 - Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall werden ein Benutzername und ein Kennwort codiert, jedoch nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, um den Benutzer für diese Anforderung zu authentifizieren und zu autorisieren. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie im Abschnitt [HTTP-Basisauthentifizierung mit der REST API verwenden](#).
 - Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie im Abschnitt [Tokenbasierte Authentifizierung mit der REST API verwenden](#).
 - Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der administrative REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie im Abschnitt [Clientzertifikatsauthentifizierung mit der REST API verwenden](#).
2. V 9.0.1 Konfigurieren Sie REST API-Einstellungen, einschließlich Aktivierung von HTTP-Verbindungen und Änderung der Portnummer. Weitere Informationen finden Sie im Abschnitt [IBM MQ Console und REST API konfigurieren](#).
3. V 9.0.5 Konfigurieren Sie optional die REST API für MFT. Weitere Informationen finden Sie unter [„Erforderliche Konfiguration für die MFT REST API“](#) auf Seite 75.
4. V 9.0.2 Konfigurieren Sie optional Cross Origin Resource Sharing für die REST API. Standardmäßig können Sie auf die REST API nicht über Webressourcen zugreifen, die nicht in derselben Domäne gehostet werden wie die REST API. Dies bedeutet, dass Kreuzursprungsanforderungen nicht aktiviert

sind. Sie können Cross Origin Resource Sharing (CORS) konfigurieren, um Cross-Origin-Anforderungen von angegebenen URLs zu ermöglichen. Weitere Informationen finden Sie im Abschnitt [CORS für den REST API konfigurieren](#).

5. Verwenden Sie den REST API. Weitere Informationen finden Sie unter „[administrative REST API verwenden](#)“ auf Seite 78, „[Fernverwaltung über die REST API](#)“ auf Seite 84, und in der [Referenz zu REST API](#).

Anmerkung: Sie können den MQWeb-Server jederzeit mit dem Befehl **endmqweb** stoppen. Wenn der mqweb-Server nicht aktiv ist, können Sie die REST API oder IBM MQ Console allerdings nicht verwenden.

V 9.0.5 Erforderliche Konfiguration für die MFT REST API

Auszuführende Schritte, um Managed File Transfer für die Verwendung der REST API zu konfigurieren

Geben Sie in der Datei `mqwebuser.xml` Folgendes ein:

- Eigenschaft **mqRestMftEnabled** muss zu *true*

Anmerkung: Sie müssen den mqweb-Server erneut starten, wenn Sie den Wert dieser Eigenschaft ändern.

- **mqRestMftCoordinationQmgr** zu einem geeigneten MFT-Koordinationswarteschlangenmanager, der lokal auf der Maschine ausgeführt wird, auf der der mqweb-Server aktiv ist.

Damit ein Warteschlangenmanager als MFT-Koordinationswarteschlangenmanager für die REST API fungieren kann, müssen Sie folgenden Befehl ausführen:

- **fteSetupCoordination** in demselben lokalen Warteschlangenmanager, der für die Eigenschaft **mqRestMftCoordinationQmgr** in der Datei `mqwebuser.xml` festgelegt wurde.

Dieser Befehl generiert eine MQSC-Datei, die Definitionen von IBM MQ-Objekten enthält. Beachten Sie, dass der Befehl auf jeder Maschine ausgeführt werden kann, auf der MFT installiert ist.

- Die im vorherigen Schritt generierte MQSC-Datei sollte für den Koordinationswarteschlangenmanager ausgeführt werden, um die erforderlichen IBM MQ-Objekte zu erstellen.

Zugehörige Informationen

[MFT REST API-Sicherheit konfigurieren](#)

[MFT REST API verwalten](#)

[GET - Liste der Übertragungen](#)

[GET - Übertragungsstatus](#)

V 9.0.1 URL der REST API bestimmen

Die Standard-URL für den Zugriff auf die administrative REST API in IBM MQ 9.0.4 und höher ist `https://localhost:9443/ibmmq/rest/v1/admin`. In IBM MQ 9.0.3 und früher lautet die Standard-URL `https://localhost:9443/ibmmq/rest/v1`. Die Standard-URL für den Zugriff auf die messaging REST API lautet: `https://localhost:9443/ibmmq/rest/v1/messaging`. Wenn der Standardhost oder -port geändert wird oder wenn HTTP aktiviert ist, können Sie die URL mit dem Befehl **dspmqweb** ermitteln.

Informationen zu diesem Vorgang

V 9.0.4 Ab IBM MQ 9.0.4 können Sie mit dem Befehl **dspmqweb status** die URL der REST API unter Windows und Linux sowie unter z/OS ermitteln.



In IBM MQ 9.0.3 und früher unter Windows und Linux können Sie den Befehl **dspmqweb** verwenden. Unter z/OS müssen Sie die Datei `messages.log` durchsuchen, um die URL zu bestimmen.

Prozedur


V 9.0.4

Verwenden Sie eine der folgenden Methoden, um die URL zu bestimmen:

- Verwenden Sie ab IBM MQ 9.0.4 den Befehl **dspmqweb status** als privilegierter Benutzer:
 1. Stellen Sie sicher, dass der mqweb-Server aktiv ist, indem Sie den Befehl **strmqweb** in der Befehlszeile eingeben.
 2. Zeigen Sie die URL an, indem Sie den Befehl `dspmqweb status` in der Befehlszeile eingeben.
- Verwenden Sie in IBM MQ 9.0.3 und früher eine der folgenden Methoden:

-   Verwenden Sie unter Windows oder Linux den Befehl **dspmqweb** als privilegierter Benutzer:

1. Stellen Sie sicher, dass der mqweb-Server aktiv ist, indem Sie den Befehl **strmqweb** in der Befehlszeile eingeben.
2. Zeigen Sie die URL an, indem Sie den Befehl `dspmqweb` in der Befehlszeile eingeben.

-  Suchen Sie unter z/OS in der Datei `messages.log` nach der URL:

1. Öffnen Sie die Datei `messages.log`.

Die Datei `messages.log` befindet sich im Pfad `WLP_user_directory/servers/mqweb/logs`, wobei `WLP_user_directory` für das Verzeichnis steht, das angegeben wurde, als das Script **crtmqweb.sh** zum Erstellen der Definition des mqweb-Servers ausgeführt wurde.

2. Suchen Sie nach dem neuesten CWWKT0016I-Nachrichtencode, der mit `ibmmq/rest/v1` endet. Die URL ist in dieser Nachricht enthalten.

Beispiel

Das folgende Beispiel zeigt den Nachrichtencode CWWKT0016I mit der URL als Standard-URL:

```
A CWWKT0016I: Web application available (default_host): https://local-host:9443/ibmmq/rest/v1
```

Überprüfen, ob der mqweb-Server unter z/OS richtig konfiguriert ist

Schritte, die Sie ausführen müssen, um zu überprüfen, ob der mqweb-Server unter z/OS richtig konfiguriert ist, und um allgemeine Konfigurationsprobleme zu beheben.

Vorgehensweise

1. Überprüfen Sie die gestartete Task des mqweb-Servers.

Es sollten Jobprotokollnachrichten ähnlich der folgenden angezeigt werden:

```
+CWWKE0001I: The server mqweb has been launched.
+CWWKZ0001I: Application com.ibm.mq.rest started in 1.493 seconds.
+CWWKZ0001I: Application com.ibm.mq.console started in 0.885 seconds.
+CWWKF0011I: The server mqweb is ready to run a smarter planet.
```

Es dürfen keine Nachrichten in STDERR vorhanden sein.

Es sollten Nachrichten in STDOUT vorhanden sein, die der folgenden entsprechen:

```
Launching mqweb (WebSphere Application Server 17.0.0.2, WAS FOR Z/OS 17.0.0.2,
MQM MVS/ESA V9 R0.4/wlp-1.0.17.cl170220170523-1818) on IBM J9 VM,
version pmz6480sr4fp7-20170627_02 (SR4 FP7) (en_US)
[AUDIT ] CWWKE0001I: The server mqweb has been laun
ched.
```

[AU□

```

DIT ] CWWKG0028A: Processing included configuration resource: /mqm/V9ROM4/web/mq/etc/
mqweb.xml
[AUDIT ] CWWKG0028A: Processing included configuration resource:
var/mqm/mqweb904/servers/mqweb/mqwebuser.xml
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/api/docs/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/api/explorer/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibm/api/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibm/api/explorer/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibm/api/docs/subscription/websocket/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport:9080/ibmmq/rest/v1/
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.493 seconds.
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibmmq/console/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibmmq/console/internal/
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.console started in 1.459 seconds.
[AUDIT ] CWWKF0012I: The server installed the following features:
[jsr-2.2, servlet-3.1, ssl-1.0, jndi-1.0, basicAuthenticationMQ-1.0,
apiDiscovery-1.0, localConnector-1.0, appSecurity-2.0, jaxrs-1.1,
concurrent-1.0, json-1.0, websocket-1.0, distributedMap-1.0,
applicationMonitorMQ-1.0].
[AUDIT ] CWWKF0011I: The server mqweb is ready to run a smarter planet.
[AUDIT ] MQWB2019I: MQ Console level: 9.0.4 - V904-GA904-L171016
[AUDIT ] MQWB0023I: MQ REST API level: 9.0.4 - V904-GA904-L171016

```

Anmerkungen:

- Wenn die Prozedur nicht gestartet werden kann, lösen Sie die Probleme.
- Wenn die Nachrichten ähnlich wie folgt lauten:

```

Web application available (default_host):
http://localhost:portnumber/ibmmq/console/

```

Sie haben die Eigenschaft **httpHost** nicht festgelegt, um ferne Verbindungen zum mqweb-Server zuzulassen.

```
<variable name="httpHost" value="*" />
```

- Wenn Sie die Eigenschaft **httpHost** geändert haben, erhalten Sie eine Nachricht ähnlich der folgenden:

```

Web application available (default_host):
yourhost:portnumber/ibmmq/console/

```

Abhängig von Ihrer TCP/IP-Konfiguration können Sie die Adresse, die als URL angezeigt wird, möglicherweise nicht verwenden.

- Änderungen an den Konfigurationsdateien des XML-Servers werden ein paar Sekunden später übernommen. Es ist nicht erforderlich, den mqweb-Server erneut zu starten.
- Stellen Sie eine Verbindung zu IBM MQ Console her.
Sie sollten ein Fenster mit einem Titel von IBM MQ Console anzeigen.

Anmerkungen:

- Wenn ein Fenster mit `This site can't be reached` oder `Context Root Not Found` angezeigt wird, ist IBM MQ Console nicht aktiv; Sie sollten warten.
- Wenn kein Fenster mit dem Titel `IBM MQ Console` angezeigt wird, gibt es möglicherweise weitere Diagnoseinformationen im Protokoll `WLP_USER_DIRECTORY/servers/mqweb/logs/messages.log`, wobei `WLP_USER_DIRECTORY` für das Verzeichnis steht, das Sie bei der Ausführung des Scripts **crtmqweb.sh** zum Erstellen der Definition des mqweb-Servers angegeben haben.

Beachten Sie, dass diese Datei in ASCII vorliegt. Zum Anzeigen der Datei können Sie **oedit** über eine USS-Befehlszeile oder die ISPF-Option 3.17 verwenden und den Zeilenbefehl **va** (ASCII anzeigen) verwenden.

c. Sie können Nachrichten in STDOUT ähnlich wie folgt erhalten:

```
[WARNING ] SRVE0190E: File not found: /nls/en_GB/labels.json
[WARNING ] SRVE0190E: File not found: /nls/en_GB/errors.json
[WARNING ] SRVE0190E: File not found: /nls/en_GB/strings.json
[WARNING ] SRVE0190E: File not found: /nls/en_GB/
pcf.json
```

Diese Nachrichten können ignoriert werden.

d. Wenn Warteschlangenmanager auf dem System, auf dem der mqweb-Server aktiv ist, definiert sind und auf derselben Ebene ausgeführt werden wie der mqweb-Server, werden diese Warteschlangenmanager im Widget **Lokale Warteschlangenmanager** in der MQ Console angezeigt.

Wenn keine WS-Manager angezeigt werden, verfügen Sie nicht über Warteschlangenmanager auf derselben Ebene wie der mqweb-Server, die seit dem letzten IPL gestartet wurden.

3. Wenn Sie eine Nachricht erhalten:

```
Lost communication with the server Could not establish communication with
the server.
```

a) Gehen Sie in der Prozedur zum Starten des mqweb-Servers wie folgt vor:

- i) Haben die STEPLIB-Bibliotheken die richtige Version und sind sie APF-autorisiert.
- ii) Überprüfen Sie die PATH- und LIBPATH-Punkte auf den richtigen Pfad. PATH und LIBPATH sind in der von Ihnen gestarteten Taskprozedur Ihres Web-Servers definiert, die Sie auf der Basis von CSQ4WEBS basierend auf CSQ4WEBS verwenden.

b) Verwenden Sie in USS den Befehl `ls -Eltr PathPrefix/web/bin/dspmq`, wobei PathPrefix der Installationspfad von IBM MQ Unix System Services Components ist.

Dies sollte eine Ausgabe ähnlich der folgenden anzeigen: `-rwxr-xr-t a-s- ... /mqm/V9R0M4/web/bin/dspmq`

Überprüfen Sie, ob die Attribute `t` und `a` gesetzt sind.

Verwenden Sie den folgenden Befehl:

- `chmod +t PathPrefix/web/bin/dspmq` zum Festlegen des Stickers (`t`)
- `extattr +a PathPrefix/web/bin/dspmq` zum Festlegen des APF-berechtigten Attributs (`a`)

V 9.0.1 administrative REST API verwenden

Wenn Sie die administrative REST API verwenden, rufen Sie HTTP-Methoden für URLs auf, die die verschiedenen IBM MQ-Objekte darstellen, z. B. Warteschlangenmanager oder Warteschlangen. Die HTTP-Methode, z. B. POST, stellt den Typ der Aktion dar, die für das Objekt ausgeführt werden soll, das durch die URL dargestellt wird. Weitere Informationen zu der Aktion können in JSON als Teil der Nutzdaten der HTTP-Methode bereitgestellt werden oder in Abfrageparametern codiert werden. Informationen zum Ergebnis der Ausführung der Aktion können als Hauptteil der HTTP-Antwort zurückgegeben werden.

Vorbereitende Schritte

Beachten Sie folgende Punkte, bevor Sie die administrative REST API verwenden:

- **V 9.0.2** Sie müssen sich beim MQWeb-Server authentifizieren, um die administrative REST API verwenden zu können. Die Authentifizierung kann über die HTTP-Basisauthentifizierung, die Clientzertifikatsauthentifizierung oder die tokenbasierte Authentifizierung erfolgen. Weitere Informationen zur Verwendung dieser Authentifizierungsmethoden finden Sie im Abschnitt [IBM MQ Console und REST API-Sicherheit](#).
- Bei der REST API muss die Groß-/Kleinschreibung beachtet werden. Beispiel: Ein HTTP GET in der folgenden URL zeigt keine Informationen an, wenn der WS-Manager `qmgr1` genannt wird.

```
/ibmmq/rest/v1/admin/qmgr/QMGR1
```

- Nicht alle Zeichen, die in IBM MQ-Objektnamen verwendet werden können, können direkt in einer URL codiert werden. Um diese Zeichen richtig zu codieren, muss die entsprechende URL-Codierung verwendet werden:
 - Ein Schrägstrich (/) wird durch %2F codiert.
 - Ein Prozentzeichen (%) wird durch %25 codiert.
- Aufgrund des Verhaltens einiger Browser können Sie keine Namensobjekte verwenden, indem Sie nur Punkt- oder Schrägstrich-Zeichen verwenden.

Informationen zu diesem Vorgang

Wenn Sie die REST API zum Ausführen einer Aktion für ein Objekt verwenden, müssen Sie zunächst eine URL erstellen, um dieses Objekt darzustellen. Jede URL beginnt mit einem Präfix, das den Hostnamen und den Port beschreibt, an die die Anforderung gesendet werden soll. Der Rest der URL beschreibt ein bestimmtes Objekt oder eine Gruppe von Objekten, die als Ressource bezeichnet werden.

Die Aktion, die für die Ressource ausgeführt werden soll, definiert, ob die URL Abfrageparameter benötigt oder nicht. Sie definiert außerdem die verwendete HTTP-Methode und die Angabe, ob zusätzliche Informationen an die URL gesendet oder von der URL in JSON-Form zurückgegeben werden. Die zusätzlichen Informationen können Teil der HTTP-Anforderung sein oder als Teil der HTTP-Antwort zurückgegeben werden.

Nachdem Sie die URL erstellt und optionale JSON-Nutzdaten zum Senden in der HTTP-Anforderung erstellt haben, können Sie die HTTP-Anforderung an IBM MQ senden. Sie können die Anforderung mithilfe der HTTP-Implementierung senden, die in die Programmiersprache Ihrer Wahl integriert ist. Sie können die Anforderungen auch über Befehlszeilentools wie z. B. cURL oder einen Webbrowser oder Web-Browser-Add-on senden.

Wichtig: Sie müssen als Minimum die Schritte „1.a“ auf Seite 79 und „1.b“ auf Seite 80 ausführen.

Vorgehensweise

1. Konstruieren Sie die URL:

a) Beginnen Sie mit der folgenden Präfix-URL:

V 9.0.4 IBM MQ 9.0.4 und höher:

```
https://host:port/ibmmq/rest/v1/admin
```

IBM MQ 9.0.3 und früher:

```
https://host:port/ibmmq/rest/v1
```

host

Gibt den Hostnamen oder die IP-Adresse an, unter der die administrative REST API verfügbar ist.

Der Standardwert ist localhost.

port

Gibt die HTTPS-Portnummer an, die von der administrativen REST API verwendet wird.

Der Standardwert ist 9443.

V 9.0.1 Wenn Sie HTTP-Verbindungen aktivieren, können Sie HTTP anstelle von HTTPS verwenden. Weitere Informationen zur Aktivierung von HTTP finden Sie im Abschnitt [HTTP- und HTTPS-Ports konfigurieren](#).

Weitere Informationen zum Bestimmen der Präfix-URL finden Sie im Abschnitt [„URL der REST API bestimmen“](#) auf Seite 75.

b) Fügen Sie die Ressource dem URL-Pfad hinzu.

Die folgenden Ressourcen sind verfügbar:

- [Installation](#)
- [qmgr](#)
- **V 9.0.2** [Warteschlange](#)
- **V 9.0.4** [Subskription](#)
- **V 9.0.4** [Kanal](#)

Wenn Sie beispielsweise mit Warteschlangenmanagern interagieren möchten, fügen Sie /qmgr der Präfix-URL hinzu, um die folgende URL zu erstellen:

V 9.0.4 IBM MQ 9.0.4 und höher:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr
```

IBM MQ 9.0.3 und früher:

```
https://localhost:9443/ibmmq/rest/v1/qmgr
```

c) Optional: Fügen Sie alle zusätzlichen optionalen Pfadsegmente zur URL hinzu.

In den Referenzinformationen für jeden Objekttyp können die optionalen Segmente in der URL durch die geschweiften Klammern, die sie umgeben, identifiziert werden.

Fügen Sie beispielsweise den WS-Managernamen QM1 der URL hinzu, um die folgende URL zu erstellen:

V 9.0.4 IBM MQ 9.0.4 und höher:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1
```

IBM MQ 9.0.3 und früher:

```
https://localhost:9443/ibmmq/rest/v1/qmgr/QM1
```

d) Optional: Fügen Sie der URL einen optionalen Abfrageparameter hinzu.

Fragezeichen hinzufügen,?, Variablenname, Gleichheitszeichen = und einen Wert oder eine Liste von Werten für die URL.

Wenn Sie beispielsweise alle Attribute des WS-Managers QM1 anfordern möchten, erstellen Sie die folgende URL:

V 9.0.4 IBM MQ 9.0.4 und höher:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1?attributes=*
```

IBM MQ 9.0.3 und früher:

```
https://localhost:9443/ibmmq/rest/v1/qmgr/QM1?attributes=*
```

e) Fügen Sie weitere optionale Abfrageparameter zur URL hinzu.

Fügen Sie ein Et-Zeichen, &, zur URL hinzu, und wiederholen Sie dann [Schritt d.](#)

2. Rufen Sie die relevante HTTP-Methode in der URL auf. Geben Sie alle optionalen JSON-Nutzdaten an und geben Sie die entsprechenden Sicherheitsberechtigungsnaehweise für die Authentifizierung an . Beispiel:

- Verwenden Sie die HTTP/REST-Implementierung der von Ihnen gewählten Programmiersprache.
- Verwenden Sie ein Tool wie beispielsweise ein REST-Client-Browser-Add-on oder cURL.

Vorgehensweise zum Abrufen der Managed File Transfer-Liste der Übertragungen mit der REST API

Weitere Informationen hierzu finden Sie im Abschnitt [Verwaltung mit dem REST API](#).

Gehen Sie wie folgt vor, um eine Liste der Übertragungen abzurufen:

1. Senden Sie eine [GET](#)-Anforderung an den Server, auf dem sich die Webanwendung MFT und MFT befinden. Sie müssen das Format der URL beim Senden einer Anforderung berücksichtigen.

Weitere Informationen finden Sie unter [admin/mft/transfer](#).

2. Wenn eine Anforderung empfangen wird, wird eine Prüfung für die Anforderung ausgeführt, um zu prüfen, ob die Anforderung gültig ist.

Weitere Informationen finden Sie im Abschnitt [MFT REST API-Sicherheit konfigurieren](#).

3. Wenn die Anforderung gültig ist, erhalten Sie eine erfolgreiche Antwort zusammen mit dem Antwortteil. Andernfalls wird ein entsprechender Fehlercode und Antwortcode generiert.

Eine Liste der Antworten finden Sie in [Antwortstatuscodes](#).

Gehen Sie wie folgt vor, um den Status einer Übertragung zu erhalten:

1. Senden Sie eine [GET](#)-Anforderung an den Server, auf dem sich die Webanwendung MFT und MFT befinden. Sie müssen das Format der URL beim Senden einer Anforderung berücksichtigen.

Weitere Informationen finden Sie unter [admin/mft/transfer](#).

2. Wenn eine Anforderung empfangen wird, wird eine Prüfung für die Anforderung ausgeführt, um zu prüfen, ob die Anforderung gültig ist.

Weitere Informationen finden Sie im Abschnitt [MFT REST API-Sicherheit konfigurieren](#).

3. Wenn die Anforderung gültig ist, erhalten Sie eine erfolgreiche Antwort zusammen mit dem Antwortteil. Andernfalls wird ein entsprechender Fehlercode und Antwortcode generiert.

Eine Liste der Antworten finden Sie in [Antwortstatuscodes](#).

Beispielantwortkörper

Die Übertragungsliste wird aus dem internen Speicher abgerufen, und für jeden der Übertragungen wird ein Antwortteil im JSON-Format generiert. Der Antworthauptteil für eine Übertragung sieht etwa wie folgt aus:

```
{
  "transfer": [
    {
      "destinationAgent": {
        "name": "AGENT.TRI.BANK"
      },
      "originator": {
        "host": "192.168.99.1",
        "userId": "johndoe"
      },
      "sourceAgent": {
        "name": "TESTAGENT"
      },
      "statistics": {
        "endTime": "2018-01-08T16:22:15.569Z",
        "numberOfFileFailures": 0,
        "numberOfFileSuccesses": 2,
        "numberOfFileWarnings": 0,
        "numberOfFiles": 2,
        "startTime": "2018-01-08T16:22:15.242Z"
      },
      "status": {
        "state": "successful"
      },
      "id": "414D51204D465444454D4F3320202020513E525A21109908"
    },
    {
      "destinationAgent": {
        "name": "AGENT.TRI.BANK"
      },
      "originator": {
        "host": "192.168.99.1",
        "userId": "ramsubbarao"
      }
    }
  ]
}
```

```

    "sourceAgent": {"name": "TESTAGENT"},
    "statistics": {
      "endTime": "2018-01-08T16:22:13.573Z",
      "numberOfFileFailures": 0,
      "numberOfFileSuccesses": 2,
      "numberOfFileWarnings": 0,
      "numberOfFiles": 2,
      "startTime": "2018-01-08T16:22:13.167Z"
    },
    "status": {
      "state": "successful"
    },
    "id": "414D51204D465444454D4F33202020513E525A21109702"
  }
}
]
}

```

Weitere Informationen zu den Attributen finden Sie unter [REST API JSON-Antworthauptteilattribute](#).

V 9.0.5 Status des MFT-Agenten abrufen

Vorgehensweise zum Abrufen des Managed File Transfer-Agentenstatus mit der REST API

Weitere Informationen finden Sie im Abschnitt [Verwaltung mit der REST API](#).

Gehen Sie wie folgt vor, um den Agentenstatus abzurufen

1. Senden Sie eine GET-Anforderung an den Agenten unter dem Standardkoordinationswarteschlangenmanager, der die Webanwendung MFT und MFT verwendet. Sie müssen das Format der URL beim Senden einer Anforderung berücksichtigen.

Weitere Informationen finden Sie unter [admin/mft/agent](#).

2. Wenn eine Anforderung empfangen wird, wird eine Prüfung für die Anforderung ausgeführt, um zu prüfen, ob die Anforderung gültig ist.

Weitere Informationen finden Sie im Abschnitt [MFT REST API-Sicherheit konfigurieren](#).

3. Wenn die Anforderung gültig ist, erhalten Sie eine erfolgreiche Antwort zusammen mit dem Antwortteil. Andernfalls wird ein entsprechender Fehlercode und Antwortcode generiert.

Eine Liste der Antworten finden Sie in [Antwortstatuscodes](#).

Beispiel-URLs

{baseURI}/ibmmq/rest/{version}/admin/mft/agent

Gibt eine Liste der Agenten **name**, **state** und **type** aller Agenten unter dem Koordinationswarteschlangenmanager aus.

Weitere Informationen zu diesen Attributen finden Sie unter [Agentenstatus REST API-Attribute](#).

Wenn Sie `?attributes=*` am Ende der URL-Zeichenfolge hinzufügen, enthält die Ausgabe eine Liste aller Attribute für jeden Agenten, der sich unter dem Koordinationswarteschlangenmanager befindet.

Wenn Sie `/ {agentName}` am Ende der URL-Zeichenfolge hinzufügen, enthält die Ausgabe die **name**, **state** und **type** des Agenten, der mit `agentName` übereinstimmt.

Analog enthält die Ausgabe eine Liste aller Attribute für den Agenten, die mit `agentName` übereinstimmen, wenn Sie `/ {agentName}?attributes=*` am Ende der URL-Zeichenfolge hinzufügen.

{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name={<prefix>*<suffix> }

`{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name=AGENT*TEST` gibt z. B. die Standardattributzusammenfassung für die Agenten aus, die mit dem Namen `AGENT` beginnen und mit dem Namen `TEST` enden.

Wenn Sie `*TEST` von der URL weglassen, enthält die Ausgabe die Standardattributzusammenfassung für alle Agenten, die mit dem Namen `AGENT` beginnen.

{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type={agentType}

`{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type=standard` gibt z. B. die Standardattributzusammenfassung für Agenten des Typs `standard` aus.

Wenn Sie `type={agentType}` durch `state={agentState}` ersetzen, zeigt die Ausgabe ebenfalls die Standardattributzusammenfassung für die Agenten des angegebenen Status an.

{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type={agentType}&attributes=*
`{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type=standard&attributes=*`
gibt z. B. alle Attribute für einen Agenten aus, dessen Typ *standard* ist.

Sie können auch nach einer Kombination von Attributen filtern. Beispiel:

- `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name=*bob&type=standard&state=ready&attributes=*`
- `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type=standard&state=ready&attributes=*`
- `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name=agent*&state=ready&attributes=*`

Beispielantwortkörper

Weitere Informationen zu den beschriebenen Attributen finden Sie im Abschnitt [Agentenstatus REST API-Attribute](#) und [Agentenstatus REST API-Antworthauptteilattribute](#).

Durch Absetzen des Befehls `/ibmmq/rest/v1/admin/mft/agent/` wird der Basisagentenstatus des Agentennamens, `-types` und `-status` aufgelistet. Beispiel:

```
{
  "agent": [ { "name": "AGENT1",
              "state": "ready",
              "type": "standard" },
            { "name": "AGENT2",
              "state": "ready",
              "type": "standard" },
            { "name": "BRIDGE_AGENT3",
              "type": "protocolBridge",
              "state": "ready" },
            { "name": "CD_AGENT",
              "type": "connectDirectBridge",
              "state": "ready" } ]
}
```

Wird jeder der folgenden Befehle ausgegeben:

```
/ibmmq/rest/v1/admin/mft/agent?type=standard
/ibmmq/rest/v1/admin/mft/agent?state=stopped
/ibmmq/rest/v1/admin/mft/agent?name=AGENT*
```

In sind die Basisinformationen für einen Agenten aufgelistet, der mit dem jeweiligen Namen, Status und Typ in jedem Befehl übereinstimmt. Beispiel:

```
{ "agent": [ { "name": "AGENT1",
              "state": "ready",
              "type": "standard" },
            { "name": "AGENT2",
              "state": "ready",
              "type": "standard" } ]
}
{ "agent": [ { "name": "AGENT1",
              "state": "stopped",
              "type": "standard" },
            { "name": "AGENT2",
              "state": "stopped",
              "type": "standard" } ]
}
{ "agent": [ { "name": "AGENT1",
              "state": "ready",
              "type": "standard" },
            { "name": "AGENT2",
              "state": "ready",
              "type": "standard" } ]
}
```

```

    "type": "standard" } ]
}

```

Wenn Sie den Befehl `/ibmmq/rest/v1/admin/mft/agent?attributes=general&type=standard` absetzen, werden alle Nachrichten des Typs *Standard* zusammen mit den Attributen `general` aufgelistet. Beispiel:

```

{
  "agent": [
    {
      "name": "AGENT1",
      "state": "ready",
      "type": "standard",
      "general": {
        "description": "Standard connected to the qmgr in client mode",
        "statusAge": "06:31:00",
        "version": "9.0.3.0",
        "level": "p903-L170513",
        "statusPublicationRate": 300,
        "statusPublishTime": "2017-10-31T06:57:07.000Z",
        "maximumQueuedTransfers": 1000,
        "maximumDestinationTransfers": 25,
        "maximumSourceTransfers": 25,
        "operatingSystem": "Windows7" }
    },
    {
      "name": "AGENT2",
      "state": "ready",
      "type": "standard",
      "general": {
        "description": "Standard connected to qmgr in Binding mode",
        "statusAge": "05:00:00",
        "version": "9.0.3.0",
        "level": "p903-L170513",
        "statusPublicationRate": 300,
        "statusPublishTime": "2017-09-13T09:10:09.000Z",
        "maximumQueuedTransfers": 1000,
        "maximumDestinationTransfers": 25,
        "maximumSourceTransfers": 25,
        "operatingSystem": "Windows7" }
    }
  ]
}

```

V 9.0.4 Fernverwaltung über die REST API

Sie können über die REST API ferne Warteschlangenmanager und die IBM MQ-Objekte, die diesen Warteschlangenmanagern zugeordnet sind, verwalten. Diese Fernverwaltung umfasst Warteschlangenmanager, die sich auf demselben System, aber nicht in derselben IBM MQ-Installation wie der mqweb-Server befinden. Daher können Sie die REST API verwenden, um Ihr gesamtes IBM MQ-Netz mit nur einer Installation zu verwalten, die den mqweb-Server ausführt. Zum Verwalten von fernen Warteschlangenmanagern müssen Sie das administrative REST API-Gateway so konfigurieren, dass mindestens ein Warteschlangenmanager in derselben Installation wie der mqweb-Server als Gateway-Warteschlangenmanager fungiert. Anschließend können Sie den fernen Warteschlangenmanager in der Ressourcen-URL der REST API angeben, um die angegebene Verwaltungsaktion auszuführen.

Vorbereitende Schritte

Sie können die Fernverwaltung verhindern, indem Sie das administrative REST API-Gateway inaktivieren. Weitere Informationen finden Sie im Abschnitt [administrative REST API-Gateway konfigurieren](#).

Um das administrative REST API-Gateway verwenden zu können, müssen folgende Bedingungen erfüllt sein:

- Der mqweb-Server muss konfiguriert und gestartet sein. Weitere Informationen zum Konfigurieren und Starten des mqweb-Servers finden Sie im Abschnitt [„Erste Schritte mit der administrative REST API“ auf Seite 72](#).
- Der Warteschlangenmanager, der als Gateway-Warteschlangenmanager konfiguriert werden soll, muss sich in derselben Installation befinden wie der mqweb-Server.
- Der ferne Warteschlangenmanager, den Sie verwalten möchten, muss IBM MQ 8.0 oder höher haben.

- Sie müssen sicherstellen, dass alle in Ihrer Anforderung angegebenen Attribute für das System, an das die Anforderung gesendet wird, gültig sind. Wenn sich beispielsweise der Gateway-Warteschlangenmanager unter Windows und der ferne Warteschlangenmanager unter z/OS befinden, können Sie nicht anfordern, dass das Attribut `dataCollection.statistics` für eine HTTP-GET-Anforderung an die Ressource `queue` zurückgegeben wird.
- Sie müssen sicherstellen, dass alle in Ihrer Anforderung angegebenen Attribute für die Version von IBM MQ gültig sind, an die die Anforderung gesendet wird. Wenn der ferne Warteschlangenmanager beispielsweise IBM MQ 8.0 ausführt, können Sie nicht anfordern, dass das Attribut `extended.enableMediaImageOperations` für eine HTTP GET-Anforderung in der Ressource `queue` zurückgegeben wird.
- Sie müssen eine der folgenden unterstützten REST-Ressourcen verwenden:

- `/queue`
- `/subscription`
- `V 9.0.5 /channel`
- `V 9.0.5 /mqsc`
- `/qmgr`

Die `/qmgr`-Ressource gibt nur eine Untergruppe der Attribute zurück, wenn Sie einen fernen WS-Manager abfragen: `name`, `status.started`, `status.channelInitiatorState`, `status.ldapConnectionState`, `status.connectionCount` und `status.publishSubscribeState`.

Informationen zu diesem Vorgang

Wenn Sie das administrative REST API-Gateway zum Verwalten von fernen Warteschlangenmanagern verwenden möchten, müssen Sie die Warteschlangenmanager auf die Fernverwaltung vorbereiten. Dies bedeutet, dass Übertragungswarteschlangen, Empfangsprogramme und Sender- und Empfängerkanäle zwischen dem Gateway-Warteschlangenmanager und dem fernen Warteschlangenmanager konfiguriert werden müssen. Anschließend können Sie eine REST-Anforderung an den fernen Warteschlangenmanager senden, indem Sie den WS-Manager in der Ressourcen-URL angeben. Der Gateway-Warteschlangenmanager wird angegeben, indem das Attribut `mqRestGatewayQmgr` mit dem `setmqweb`-Befehl auf den Namen des Gateway-Warteschlangenmanagers gesetzt wird oder indem der Name des Gateway-Warteschlangenmanagers in einem Header gesendet wird, der mit der Anforderung gesendet wird. Die Anforderung wird über den Gateway-WS-Manager an den fernen Warteschlangenmanager gesendet. Die Antwort wird mit einem Header zurückgegeben, der den Warteschlangenmanager angibt, der als Gateway-WS-Manager verwendet wurde.

Vorgehensweise

1. Konfigurieren Sie die Kommunikation zwischen dem Gateway-Warteschlangenmanager und den fernen WS-Managern, die Sie verwalten möchten. Diese Konfigurationsschritte sind dieselben Schritte, die erforderlich sind, um die ferne Verwaltung von `runmqsc` und `PCF` zu konfigurieren.

Weitere Informationen zu diesen Schritten finden Sie im Abschnitt [„Ferne Verwaltung von einem lokalen WS-Manager“](#) auf Seite 202.

2. Konfigurieren Sie die Sicherheit auf den fernen WS-Managern:

- a) Stellen Sie sicher, dass die relevanten Benutzer-IDs auf dem System vorhanden sind, auf dem der ferne WS-Manager ausgeführt wird. Die Benutzer-ID, die auf dem fernen System vorhanden sein muss, ist von der Rolle des Benutzers der REST API abhängig:
 - Wenn der Benutzer der REST API Mitglied der Gruppe `MQWebAdmin` oder `MQWebAdminRO` ist, muss die Benutzer-ID, die den `mqweb`-Server gestartet hat, auf dem fernen System vorhanden sein. Auf der IBM MQ Appliance ist `mqsystem` der Benutzer, der den `mqweb`-Server startet.
 - Wenn der Benutzer der REST API Mitglied der Gruppe `MQWebUser` ist, muss die Benutzer-ID der REST API auf dem fernen System vorhanden sein.

- b) Stellen Sie sicher, dass den relevanten Benutzer-IDs die erforderlichen Berechtigungsstufen für den Zugriff auf die entsprechenden REST API-Ressourcen auf dem fernen Warteschlangenmanager erteilt werden:
- Berechtigung zum Nachrichteneinlegen in den SYSTEM.ADMIN.COMMAND.QUEUE.
 - Berechtigung zum Nachrichteneinlegen in den SYSTEM.REST.REPLY.QUEUE.
 - Berechtigung zum Zugriff auf die Übertragungswarteschlangen, die für die Fernverwaltung definiert sind.
 - Berechtigung zum Anzeigen von Warteschlangenmanagerattributen.
 - Berechtigung zum Ausführen der REST-Anforderungen. Weitere Informationen finden Sie im Abschnitt über Sicherheitsanforderungen in den [Referenzinformationen zu REST API-Ressourcen](#).
3. Konfigurieren Sie, welcher lokale WS-Manager als Gateway verwendet wird. Sie können einen Standard-Gateway-WS-Manager konfigurieren, den Gateway-Warteschlangenmanager in einem HTTP-Header angeben oder eine Kombination aus beiden Methoden verwenden:
- Konfigurieren Sie einen Standard-Gateway-Warteschlangenmanager mit dem Befehl **setmqweb** :

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

Hierbei steht *qmgrName* für den Namen des Gateway-Warteschlangenmanagers.

Dieser Gateway-WS-Manager wird verwendet, wenn die beiden folgenden Anweisungen wahr sind:

- Es wurde kein Warteschlangenmanager im Header `ibm-mq-rest-gateway-qmgr` einer REST-Anforderung angegeben.
 - Der in der Ressourcen-URL der REST API angegebene Warteschlangenmanager ist kein lokaler Warteschlangenmanager.
- Konfigurieren Sie den Gateway-Warteschlangenmanager auf jeder REST-Anforderung, indem Sie den HTTP-Header `ibm-mq-rest-gateway-qmgr` auf den Namen des Gateway-WS-Managers setzen.
4. Geben Sie den Namen des fernen Warteschlangenmanagers an, der in der Ressourcen-URL verwaltet werden soll.
- Wenn Sie beispielsweise eine Liste der Warteschlangen vom fernen Warteschlangenmanager `remoteQM` abrufen möchten, verwenden Sie die folgende URL:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/remoteQM/queue
```

Ergebnisse

Ein `ibm-mq-rest-gateway-qmgr`-Header wird mit der REST-Antwort zurückgegeben. Dieser Header gibt an, welcher WS-Manager als Gateway-Warteschlangenmanager verwendet wurde.

Beispiel

Im folgenden Beispiel gibt es drei IBM MQ-Installationen auf zwei Maschinen. Unter `Machine 1` gibt es eine `Installation 1` und eine `Installation 2`. Unter `Machine 2` gibt es eine `Installation 3`. Ein `mqweb`-Server ist für `Installation 1` konfiguriert. In jeder Installation ist ein einzelner Warteschlangenmanager vorhanden, und diese Warteschlangenmanager sind für die Fernverwaltung konfiguriert. Das heißt, die folgenden Empfangsprogramme, Kanäle und Warteschlangen sind konfiguriert und gestartet:

- Auf WS-Manager `QM1` in `Installation 1` unter `Machine 1`:
 - Senderkanal `QM1.to.WSM2`
 - Empfängerkanal `QM2.to.QM1`
 - Senderkanal `QM1.to.QM3`
 - Empfängerkanal `QM3.to.QM1`

- Übertragungswarteschlange QM2
- Übertragungswarteschlange QM3
- Ein Listener, der an Port 1414 konfiguriert ist
- Auf WS-Manager QM2 in Installation 2 unter Machine 1:
 - Senderkanal QM2.to.QM1
 - Empfängerkanal QM1.to.WSM2
 - Übertragungswarteschlange QM1
 - Ein Listener ist an Port 1415 konfiguriert.
- Auf WS-Manager QM3 in Installation 3 unter Machine 2:
 - Senderkanal QM3.to.QM1
 - Empfängerkanal QM1.to.WSM3
 - Übertragungswarteschlange QM1
 - Der Standardlistener

Eine Warteschlange, Qon2 ist in QM2 definiert, und eine Warteschlange Qon3 ist in QM3 definiert.

Der Benutzer mquser ist auf beiden Maschinen definiert, hat die Rolle MQWebAdmin in der REST API inne und erhält die Berechtigung, auf die entsprechenden Warteschlangen in jedem Warteschlangenmanager zuzugreifen.

Der Befehl setmqweb wird zum Konfigurieren des Warteschlangenmanagers QM1 als Standard-Gateway-Warteschlangenmanager verwendet.

Das folgende Diagramm zeigt diese Konfiguration:

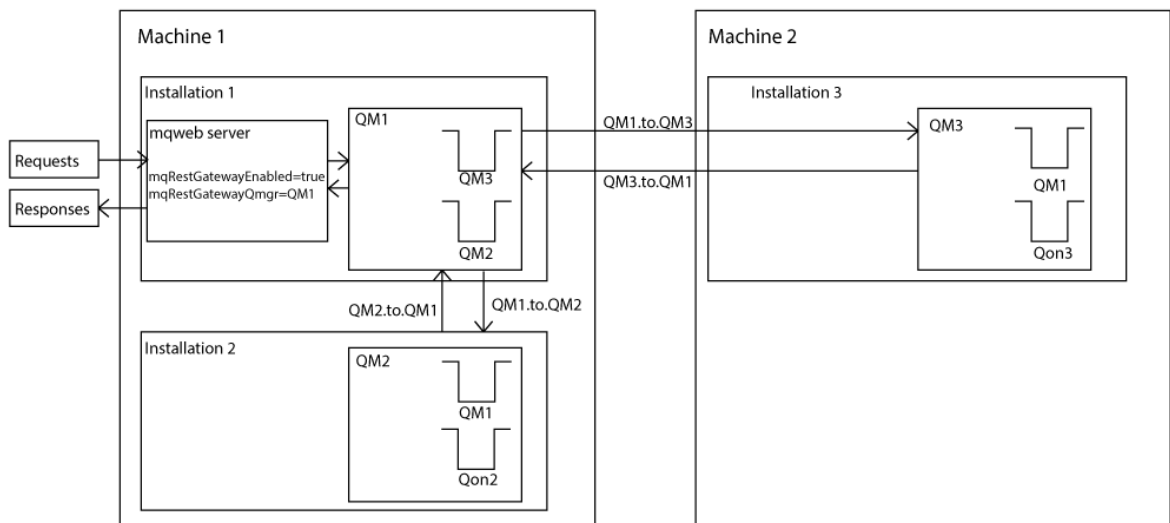


Abbildung 15. Diagramm einer Beispielkonfiguration für Fernverwaltung unter Verwendung der REST API

Die folgende REST-Anforderung wird an den mqweb-Server gesendet:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM2/queue?attributes=general.isTransmissionQueue
```

Die folgende Antwort wird empfangen:

```
{
  "queue" :
  [ {
    "general": {
```

```

        "isTransmissionQueue": true
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false
    },
    "name": "Qon2",
    "type": "local"
  }
]
}

```

Die folgende REST-Anforderung wird an den mqweb-Server gesendet:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM3/queue?attributes=general.isTransmissionQueue,general.description
```

Die folgende Antwort wird empfangen:

```

{
  "queue" :
  [
    {
      "general": {
        "isTransmissionQueue": true,
        "description": "Transmission queue for remote admin."
      },
      "name": "QM1",
      "type": "local"
    },
    {
      "general": {
        "isTransmissionQueue": false,
        "description": "A queue on QM3."
      },
      "name": "Qon3",
      "type": "local"
    }
  ]
}

```

V 9.0.2 REST API-Zeitmarken

Wenn Datums- und Zeitinformationen von der administrative REST API zurückgegeben werden, geschieht dies in UTC (koordinierte Weltzeit) und in einem festgelegten Format.

Das Datum und die Uhrzeit werden im folgenden Zeitmarkenformat zurückgegeben:

```
YYYY-MM-DDTHH:mm:ss:sssZ
```

Beispiel: 2012-04-23T18:25:43.000Z, wobei das Z angibt, dass die Zeitzone UTC (Coordinated Universal Time) ist.

In IBM MQ 9.0.2 ist die Genauigkeit dieser Zeitmarke nicht garantiert. Wenn der mqweb-Server beispielsweise nicht in derselben Zeitzone wie der Warteschlangenmanager gestartet wird, der in der Ressourcen-URL angegeben ist, ist die Zeitmarke möglicherweise nicht korrekt. Wenn die Sommerzeit-Anpassungen erforderlich sind, ist die Zeitmarke möglicherweise nicht korrekt.

REST API-Fehlerbehandlung

Die REST API meldet Fehler, indem sie einen entsprechenden HTTP-Antwortcode, z. B. 404 (Nicht gefunden), und eine JSON-Antwort zurückgibt. Jeder HTTP-Antwortcode, der sich nicht im Bereich von 200 bis 299 befindet, wird als Fehler angesehen.

Fehlerantwortformat

Die Antwort liegt im JSON-Format in UTF-8-Codierung vor. Es enthält verschachtelte JSON-Objekte:

- Ein äußeres JSON-Objekt, das ein einzelnes JSON-Array mit dem Namen `error` enthält.

- Jedes Element in dem Array ist ein JSON-Objekt, das Informationen zu einem Fehler darstellt. Jedes JSON-Objekt enthält die folgenden Eigenschaften:

Typ

Zeichenfolge.

Der Typ des Fehlers.

messageId

Zeichenfolge.

Eine eindeutige Kennung für die Nachricht des Formulars MQWBnnnnX. Diese Kennung weist die folgenden Elemente auf:

MQWB

Ein Präfix, das anzeigt, dass die Nachricht ursprünglich in der MQ-Rest-API ausgegeben wurde.

nnnn

Eine eindeutige Zahl, die die Nachricht angibt.

X

Ein einzelner Buchstabe, der den Schweregrad der Nachricht angibt:

- I , wenn eine Nachricht rein informativ ist.
- W , wenn eine Nachricht eine Warnung zu einem Problem darstellt.
- E , wenn eine Nachricht darauf hinweist, dass ein Fehler aufgetreten ist.
- S , wenn eine Nachricht darauf hinweist, dass ein schwer wiegender Fehler aufgetreten ist.

Nachricht

Zeichenfolge.

Eine Beschreibung des Fehlers.

Erläuterung

Zeichenfolge.


Eine Erläuterung des Fehlers.

Aktion

Zeichenfolge.


Eine Beschreibung der Schritte, die ausgeführt werden können, um den Fehler zu beheben.

qmgrName

 Dieses Feld ist nur für z/OS verfügbar, wo der Warteschlangenmanager Mitglied der Gruppe mit gemeinsamer Warteschlange ist. Sie müssen den optionalen **commandScope** -Abfrageparameter oder das Attribut **queueSharingGroupDisposition** angeben haben.

Zeichenfolge.

Der Name des Warteschlangenmanagers, der den Fehler festgestellt hat.

 Dieses Feld ist für die messaging REST API nicht zutreffend.

Dieses Feld ist nur verfügbar, wenn **type** `pcf`, `java` oder `rest` ist.

Zahl.

Der MQ-Beendigungscode, der dem Fehler zugeordnet ist.

reasonCode

Dieses Feld ist nur verfügbar, wenn **type** `pcf`, `java` oder `rest` ist.

Zahl.

Der MQ-Ursachencode, der dem Fehler zugeordnet ist.

Ausnahmen

Dieses Feld ist nur verfügbar, wenn **type** `java` ist.

Array.

Ein Array mit Java- oder JMS-Ausnahmebedingungen. Jedes Element der Ausnahme-Array enthält ein **stackTrace**-Zeichenfolgearray.

Das Zeichenfolgearray **stackTrace** enthält die Details zu den einzelnen Ausnahmen, die in Zeilen aufgeteilt sind.

Fehler in Gruppen mit gemeinsamer Warteschlange

z/OS

In einer Gruppe mit gemeinsamer Warteschlange kann der optionale Abfrageparameter **commandScope** für bestimmte Befehle angegeben werden. Mit diesem Parameter kann der Befehl an andere Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange weitergegeben werden. Jeder dieser Befehle kann unabhängig von einem anderen Befehl fehlschlagen, was dazu führt, dass einige Befehle erfolgreich für die Gruppe mit gemeinsamer Warteschlange ausgeführt werden und andere Befehle fehlschlagen.

In Fällen, in denen ein Befehl teilweise fehlschlägt, wird ein HTTP-Fehlercode von 500 zurückgegeben. Für jeden Warteschlangenmanager, der einen Fehler generiert hat, werden Informationen zu diesem Fehler als Element in der **error** JSON-Feldgruppe zurückgegeben. Für jeden Warteschlangenmanager, der den Befehl erfolgreich ausgeführt hat, wird der Name des Warteschlangenmanagers als Element in einem **success** JSON-Array zurückgegeben.

Beispiele

- Das folgende Beispiel zeigt die Fehlerantwort auf einen Versuch, Informationen zu einem Warteschlangenmanager abzurufen, der nicht vorhanden ist:

```
"error": [
  {
    "type": "rest",
    "messageId": "MQWB0009E",
    "message": "MQWB0009E: Could not query the queue manager 'QM1'",
    "explanation": "The MQ REST API was invoked specifying a queue manager name which cannot be located.",
    "action": "Resubmit the request with a valid queue manager name or no queue manager name, to retrieve a list of queue managers."
  }
]
```

- Das folgende Beispiel zeigt die Fehlerantwort auf den Versuch, eine Warteschlange in einer Gruppe mit gemeinsamer Warteschlange zu löschen, die für einige Warteschlangenmanager nicht vorhanden ist:

```
"error" : [
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name to retrieve a list of queues.",
    "qmgrName": "QM1"
  },
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name to retrieve a list of queues.",
    "qmgrName": "QM2"
  }
],
"success" : [{"qmgrName": "QM3"}, {"qmgrName": "QM4"}]
```

Fehler mit MFT-Anforderungen

Wenn MFT REST API-Services nicht aktiviert sind und Sie die MFT REST API aufrufen, erhalten Sie die folgende Ausnahme:

```
{
  "error": [
    {
      "action": "Enable the Managed File Transfer REST API and resubmit the request.",
      "completionCode": 0,
      "explanation": "Managed File Transfer REST calls are not permitted as the service is disabled.",
      "message": "MQWB0400E: Managed File Transfer REST API is not enabled.",
      "msgId": "MQWB0400E",
      "reasonCode": 0,
      "type": "rest"
    }
  ]
}
```

Wenn die Services der MFT REST API aktiviert sind, aber der Koordinationswarteschlangenmanager nicht in der Datei `mqwebuser.xml` angegeben ist, erhalten Sie die folgende Ausnahmebedingung:

```
{
  "error": [
    {
      "action": "Set the coordination queue manager name and restart the mqweb server.",
      "completionCode": 0,
      "explanation": "Coordination queue manager name must be set before using Managed File Transfer REST services.",
      "message": "MQWB0402E: Coordination queue manager name is not set.",
      "msgId": "MQWB0402E",
      "reasonCode": 0,
      "type": "rest"
    }
  ]
}
```

V 9.01 REST API-Erkennung



Documentation für REST API ist in IBM Documentation und im Swagger-Format verfügbar. Swagger ist eine gängige Methode zur Dokumentation von REST-APIs. Die Swagger-Dokumentation für die REST API kann angezeigt werden, indem die API-Erkennungsfunktion auf dem mqweb-Server aktiviert wird.

Vorbereitende Schritte

Sie müssen die Sicherheit für den mqweb-Server aktivieren, um die Swagger-Dokumentation mithilfe der API-Erkennung anzeigen zu können. Weitere Informationen zu den Schritten, die für die Aktivierung der Sicherheit erforderlich sind, finden Sie im Abschnitt [Sicherheit für IBM MQ Console konfigurieren](#).

Vorgehensweise

1. Suchen Sie die Datei `mqwebuser.xml` in einem der folgenden Verzeichnisse:

-  `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
-  `WLP_user_directory/servers/mqweb`

Dabei ist `WLP_user_directory` das Verzeichnis, das bei der Ausführung des Scripts `crtmqweb.sh` zum Erstellen der mqweb-Serverdefinition angegeben wurde.

2. Fügen Sie die entsprechende XML zur Datei `mqwebuser.xml` hinzu:

- Wenn die `<featureManager>`-Tags in der Datei `mqwebuser.xml` vorhanden sind, fügen Sie folgenden XML-Code innerhalb der `<featureManager>`-Tags hinzu:
`<feature>apiDiscovery-1.0</feature>`
- Wenn die `<featureManager>`-Tags nicht in der Datei `mqwebuser.xml` vorhanden sind, fügen Sie folgenden XML-Code innerhalb der `<server>`-Tags hinzu:

```
<featureManager>
  <feature>apiDiscovery-1.0</feature>
</featureManager>
```

3. Zeigen Sie die Swagger-Dokumentation an, indem Sie eine der folgenden Methoden verwenden:

- Rufen Sie eine Webseite auf, in der Sie navigieren können, und testen Sie die REST API, indem Sie folgende URL in einem Browser eingeben:

`https://host:port/ibm/api/explorer`

Zusätzlich zur Authentifizierung jeder Anforderung müssen Sie einen `ibm-mq-rest-csrf-token`-Header für jede POST-, PATCH- oder DELETE-Anforderung einschließen.

V 9.0.5 Der erforderliche Inhalt dieses Headers variiert abhängig von der Version von IBM MQ:

- Ab IBM MQ 9.0.5 muss der HTTP-Header `ibm-mq-rest-csrf-token` in der Anforderung vorhanden sein. Der zugehörige Wert ist beliebig und kann auch ein Leerzeichen sein.
- Vor IBM MQ 9.0.5 ist der Wert des Headers der Inhalt des Cookies `csrfToken`. Ein `csrfToken` wird generiert, wenn eine HTTP GET-Methode mit der REST API verwendet wird. Sie können den Inhalt von Cookies anzeigen, indem Sie den folgenden Text in die Adressleiste Ihres Browsers eingeben:

```
javascript:alert(document.cookie)
```

Dieser Anforderungsheader wird verwendet, um zu bestätigen, dass die Berechtigungsnachweise, die zur Authentifizierung der Anforderung verwendet werden, von dem Eigner der Berechtigungsnachweise verwendet werden. Dies bedeutet, dass das Token verwendet wird, um Cross-Site-Request-Forgery-Angriffe zu verhindern.

- Rufen Sie ein einzelnes Swagger 2-Dokument ab, das die gesamte REST API beschreibt, indem Sie eine HTTP GET-Anforderung an die folgende URL absetzen:

`https://host:port/ibm/api/docs`

Dieses Dokument kann für Anwendungen verwendet werden, in denen Sie programmgesteuert in den verfügbaren APIs navigieren möchten.

host

Gibt den Hostnamen oder die IP-Adresse an, unter der die REST API verfügbar ist.

Der Standardwert ist `localhost`.

port

Gibt die HTTPS-Portnummer an, die von der administrative REST API verwendet wird.

Der Standardwert ist `9443`.

Wenn der Hostname oder die Portnummer von der Standardeinstellung abweichen, können Sie die richtigen Werte aus der URL der REST API bestimmen. Weitere Informationen zum Bestimmen der URL finden Sie im Abschnitt [„URL der REST API bestimmen“](#) auf Seite 75.

V 9.0.4 Unterstützung landessprachunterstützter Sprachen in REST API

Die REST API unterstützt, mit bestimmten Einschränkungen, die Möglichkeit, Landessprachen als Teil einer HTTP-Anforderung anzugeben.

Hintergrund

HTTP-Header ermöglichen das Angeben bestimmter Verhalten in Anforderungen und zusätzliche Informationen, die in Antworten bereitgestellt werden sollen.

In den HTTP-Headern ist die Möglichkeit enthalten, dass Informationen in einer Landessprache zurückgegeben werden. Die REST API berücksichtigt diesen Header, sofern möglich.

Eine Landessprache angeben

Im HTTP-Header ACCEPT-LANGUAGE können ein oder mehrere Sprachentags angegeben werden. Optional können Sie einen Rang mit den Tags verknüpfen, sodass die Spezifikation einer Liste nach Vorgabe sortiert werden kann. [Diese Seite](#) enthält eine hilfreiche Erläuterung des Prinzips.

Die REST API berücksichtigt diesen Header, indem sie eine Sprache aus dem Header ACCEPT-LANGUAGE auswählt und Nachrichten in dieser Sprache zurückgibt. Wenn der Header ACCEPT-LANGUAGE keine Sprache enthält, die von der REST API unterstützt werden kann, werden Nachrichten in einer Standardsprache zurückgegeben. Diese Standardsprache entspricht der Standardländereinstellung des REST API-Web-Servers.

Im Abschnitt „[Welche Daten werden übersetzt?](#)“ auf Seite 93 wird erklärt, welche Daten übersetzt werden.

Angeben der anwendbaren Sprache in den Antworten

Der HTTP-Header CONTENT-LANGUAGE in den Antworten von der REST API gibt die Sprache an, in der die Nachrichten zurückgegeben werden.

Welche Daten werden übersetzt?

Fehler- und Informationsnachrichten werden übersetzt, andere Texte nicht.

- Daten, die von einem Warteschlangenmanager zurückgegeben werden, werden nicht übersetzt, z. B. werden bei der Ausführung eines MQSC-Befehls über die REST API die Antworten des Warteschlangenmanagers in der Sprache der Ländereinstellung des Warteschlangenmanagers zurückgegeben.
- Die generierte (Swagger-)Dokumentation für die REST API, die über das Feature apiDiscovery verfügbar ist, ist in englischer Sprache.

Welche Sprachen werden unterstützt?

Neben Englisch werden Fehler- und Informationsnachrichten der REST API in die folgenden Sprachen übersetzt.

Chinesisch (Vereinfacht)

Durch den Sprachentag zh_CN verweigert

Chinesisch (traditionell)

Durch den Sprachentag zh_TW verweigert

Czech

Durch den Sprachentag cs verweigert

French

Durch den Sprachentag fr verweigert

Hungarian

Durch den Sprachentag hu verweigert

Italian

Durch den Sprachentag it verweigert

Japanese

Durch den Sprachentag ja verweigert

Korean

Durch den Sprachentag ko verweigert

Polish

Durch den Sprachentag pl verweigert

Portugiesisch (Brasilien)

Durch den Sprachentag pt_BR verweigert

Russian

Durch den Sprachentag ru verweigert

Spanish

Durch den Sprachentag es verweigert

Beispiele

In den Beispielen weist der Web-Server eine englische Standardländereinstellung auf.

Eine einzelne unterstützte Sprache angeben

In den Anforderungsheadern ist ACCEPT-LANGUAGE auf fr gesetzt. Diese Einstellung gibt an, dass Französisch die bevorzugte Sprache für übersetzbaren Text ist.

In den Answerheadern wird CONTENT-LANGUAGE auf fr gesetzt. Diese Einstellung gibt an, dass in der Antwort Fehler- und Informationsnachrichten in Französisch angezeigt werden.

Angeben einer Liste von Sprachen

In den Anforderungsheadern ist ACCEPT-LANGUAGE auf am, fr gesetzt. Diese Einstellung gibt an, dass Amharic und Französisch akzeptable Sprachen für übersetzbaren Text sind und dass Amharic die bevorzugte Sprache für den übersetzbaren Text ist.

In den Answerheadern wird CONTENT-LANGUAGE auf fr gesetzt. Diese Einstellung gibt an, dass Fehler- und Informationsnachrichten in der Antwort in Französisch ausgegeben werden, da die REST API Amharic nicht unterstützt.

Eine einzelne nicht unterstützte Sprache angeben

In den Anforderungsheadern ist ACCEPT-LANGUAGE auf am gesetzt. Diese Einstellung gibt an, dass Amharic die bevorzugte Sprache für übersetzbaren Text ist.

In den Answerheadern wird CONTENT-LANGUAGE auf en gesetzt. Diese Einstellung gibt an, dass Fehler- und Informationsnachrichten in der Antwort in Englisch ausgegeben werden, da die REST API Amharic nicht unterstützt.

V 9.0.1 Verwaltung über die IBM MQ Console

Die IBM MQ Console ist eine webbasierte Benutzerschnittstelle, über die allgemeine Verwaltungsaufgaben ausgeführt werden können.

Vorbereitende Schritte

Anmerkung:

- Inaktivieren Sie bei Verwendung der IBM MQ Console nicht den Befehlsserver auf einem Ihrer Warteschlangenmanager. Wenn der Befehlsserver für einen Warteschlangenmanager inaktiviert wird, verliert die IBM MQ Console aufgrund langer Verzögerungen bei der Verarbeitung von Befehlen ihre Reaktionsfähigkeit. Alle Befehle, die an den Warteschlangenmanager ausgegeben werden, für den der Befehlsserver die Zeit inaktiviert hat.
- Wenn Sie eine Verbindung mit der IBM MQ Console herstellen, versucht der Browser, die IBM MQ Console in der vom Browser festgelegten Ländereinstellung anzuzeigen. Es wird geprüft, ob die IBM MQ Console die angegebene Sprache unterstützt. Wenn die Sprachdateien nicht gefunden werden, wird amerikanisches Englisch verwendet, und der mqweb-Server protokolliert dies als Datei nicht gefunden Fehler. Deshalb muss mit folgenden Nachrichten gerechnet werden, wenn der Browser auf eine Sprache eingestellt ist, die von der IBM MQ Console nicht unterstützt wird:

```
SRVE0190E: File not found: /nls/en_GB/labels.json
SRVE0190E: File not found: /nls/en_GB/pcf.json
SRVE0190E: File not found: /nls/en_GB/errors.json
SRVE0190E: File not found: /nls/en_GB/strings.json
```

Prozedur

- [Mit lokalen Warteschlangenmanagern arbeiten](#)

- [Mit IBM MQ-Objekten arbeiten](#)
- [Mit Berechtigungsdatensätzen arbeiten](#)
- [Systemressourcennutzung überwachen](#)
- [Dashboard-Layouts konfigurieren](#)

V 9.0.1 Erste Schritte mit der IBM MQ Console

Damit Sie die IBM MQ Console starten können, müssen Sie zunächst die richtigen Komponenten installieren und den mqweb-Server starten. Danach können Sie die IBM MQ Console in einem Browser starten.

Vorbereitende Schritte

IBM i Unter IBM i sollten die Befehle in QSHHELL ausgeführt werden.

Informationen zu diesem Vorgang

Die Vorgehensweise bei dieser Aufgabe konzentriert sich auf die grundlegenden Schritte, die für einen schnellen Einstieg in die Arbeit mit der IBM MQ Console erforderlich sind. Weitere Informationen zum Anpassen der Konfiguration finden Sie in den Links, die in der Liste aufgeführt sind, die Sie als Nächstes tun müssen.

Anmerkung: Sie haben während der Installation die Möglichkeit, die IBM MQ Console ohne Sicherheit zu konfigurieren.

Vorgehensweise

1. Installieren Sie die Komponente IBM MQ Console und REST API:

- **V 9.0.4** **AIX** Installieren Sie unter AIX die Dateigruppe `mqm.web.rte`.
- **Linux** Installieren Sie unter Linux die Komponente MQSeriesWeb. Weitere Informationen zur Installation von Komponenten und Features unter Linux finden Sie im Abschnitt [Linux-Installationsaufgaben](#).
- **Windows** Installieren Sie unter Windows das Feature Web Administration. Weitere Informationen zur Installation von Komponenten und Features unter Windows finden Sie im Abschnitt [Windows-Installationsaufgaben](#).
- **z/OS** Installieren Sie unter z/OS das Feature IBM MQ for z/OS Unix System Web Services Components. Weitere Informationen zur Installation von Komponenten und Features unter z/OS finden Sie im Abschnitt [z/OS-Installationsaufgaben](#).

2. Konfigurieren Sie grundlegende Sicherheit, um Benutzern und Gruppen den Zugriff auf die IBM MQ Console zu ermöglichen.

a) Kopieren Sie die Beispieldatei `basic_registry.xml` aus dem Verzeichnis `MQ_INSTALLATION_PATH/web/mq/samp/configuration` in Folgendes:

- **ULW** Unter UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** Unter z/OS: `WLP_user_directory/servers/mqweb`

Hierbei steht *WLP-Benutzerverzeichnis* für das Verzeichnis, das angegeben wurde, als das Script `crtmqweb.sh` ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

b) Benennen Sie die XML-Beispieldatei in `mqwebuser.xml` um.

Anmerkung: Diese umbenannte Datei ersetzt eine vorhandene Datei, die auch für die administrative REST API verwendet wird. Kopieren Sie deshalb, falls Sie die Datei `mqwebuser.xml` für die

administrative REST API geändert haben, Ihre Änderungen in die neue XML-Datei, bevor Sie sie umbenennen.

3. Abhängig von der Plattform aktivieren Sie ferne Verbindungen zum mqweb-Server:

- **Linux** Optional unter Linux
- **Windows** Optional unter Windows
- **z/OS** Unter z/OS
- Verwenden Sie ab IBM MQ 9.0.4 den Befehl **setmqweb**:
`setmqweb properties -k httpHost -v hostname`
- Fügen Sie ab IBM MQ 9.0.1 folgenden XML-Code in der Datei `mqwebuser.xml` innerhalb der `<server>`-Tags hinzu:
`<variable name="httpHost" value="hostname"/>`

Dabei gibt *hostname* die IP-Adresse, den Hostnamen des Domänennamensservers (DNS) mit dem Domänennamenssuffix oder den DNS-Hostnamen des Servers an, auf dem IBM MQ installiert ist. Verwenden Sie einen Stern (*), um alle verfügbaren Netzchnittstellen anzugeben.



Achtung: **z/OS** **V 9.0.4**

Bevor Sie den Befehl **setmqweb** oder **dspmqweb** unter ausgeben z/OS, müssen Sie die Umgebungsvariable `WLP_USER_DIR` so setzen, dass sie auf Ihre mqweb-Serverkonfiguration verweist.

Geben Sie dazu folgenden Befehl aus:

```
export WLP_USER_DIR=WLP_user_directory
```

Dabei ist *WLP_user_directory* der Name des Verzeichnisses, das an `crtmqweb.sh` übergeben wird. Beispiel:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Weitere Informationen finden Sie unter [Liberty-Serverdefinition erstellen](#).

4. Starten Sie den MQWeb-Server, der die IBM MQ Console unterstützt:

- **Windows** **Linux** Geben Sie unter Windows und Linux als privilegierter Benutzer folgenden Befehl in der Befehlszeile ein:
`strmqweb`
- **z/OS** Starten Sie unter z/OS die Prozedur, die Sie in [Task 29: Create a procedure for the IBM WLP server](#) erstellt haben.

5. Stellen Sie eine Verbindung zur IBM MQ Console her, indem Sie folgende URL in einem Browser eingeben:

```
https://hostname:9443/ibmmq/console
```

Dabei gibt *hostname* die IP-Adresse, den Hostnamen des Domänennamensservers mit dem Domänennamenssuffix oder den DNS-Hostnamen des Servers an, auf dem IBM MQ installiert ist. Wenn Sie in Schritt 3 keine fernen Verbindungen konfiguriert haben, lautet der Wert von *hostname* `localhost`.

6. Melden Sie sich bei der IBM MQ Console an. Verwenden Sie ab IBM MQ 9.0.2 den Benutzernamen `mqadmin` und das Kennwort `mqadmin`. Verwenden Sie in IBM MQ 9.0.1 den Benutzernamen `admin` und das Kennwort `admin`.

Nächste Schritte

- Weitere Informationen zur Konfiguration der Sicherheit für die IBM MQ Console, einschließlich der Konfiguration von Benutzern und Gruppen, LDAP und Clientzertifikaten, finden Sie im Abschnitt [Sicherheit für IBM MQ Console konfigurieren](#).
- Weitere Informationen zur Konfiguration von Einstellungen der IBM MQ Console, einschließlich Aktivierung von HTTP-Verbindungen, finden Sie im Abschnitt [IBM MQ-Konsole konfigurieren](#).
- Weitere Informationen zum Bestimmen der URL, falls nicht die Standard-URL verwendet wird, finden Sie im Abschnitt [„URL der IBM MQ Console bestimmen“](#) auf Seite 97.

V 9.0.1 URL der IBM MQ Console bestimmen

Die Standard-URL für den Zugriff auf die IBM MQ Console ist `https://localhost:9443/ibmmq/console`. Wenn der Standardwert für den HTTP-Host oder -Port geändert wird oder wenn der HTTP-Port aktiviert ist, können Sie die URL mit dem Befehl **dspmweb** ermitteln.

Informationen zu diesem Vorgang

V 9.0.4 Ab IBM MQ 9.0.4 können Sie mit dem Befehl **dspmweb status** die URL der IBM MQ Console unter Windows und Linux sowie unter z/OS ermitteln. In IBM MQ 9.0.3 und früher können Sie unter Windows und Linux den Befehl **dspmweb** verwenden. Unter z/OS müssen Sie die Datei `messages.log` durchsuchen, um die URL zu bestimmen.

Prozedur

V 9.0.4

Verwenden Sie eine der folgenden Methoden, um die URL zu bestimmen:

- Verwenden Sie ab IBM MQ 9.0.4 den Befehl **dspmweb status** als [privilegierter Benutzer](#):
 1. Stellen Sie sicher, dass der mqweb-Server aktiv ist, indem Sie den Befehl **strmqweb** in der Befehlszeile eingeben.
 2. Zeigen Sie die URL an, indem Sie den Befehl `dspmweb status` in der Befehlszeile eingeben.
- Verwenden Sie in IBM MQ 9.0.3 und früher eine der folgenden Methoden:

- **Windows** **Linux** Verwenden Sie unter Windows oder Linux den Befehl **dspmweb** als [privilegierter Benutzer](#):

1. Stellen Sie sicher, dass der mqweb-Server aktiv ist, indem Sie den Befehl **strmqweb** in der Befehlszeile eingeben.
2. Zeigen Sie die URL an, indem Sie den Befehl `dspmweb` in der Befehlszeile eingeben.

- **z/OS** Suchen Sie unter z/OS in der Datei `messages.log` nach der URL:

1. Öffnen Sie die Datei `messages.log`.

Die Datei `messages.log` befindet sich im Pfad `WLP_user_directory/servers/mqweb/logs`, wobei `WLP_user_directory` für das Verzeichnis steht, das angegeben wurde, als das Script **crtmqweb.sh** zum Erstellen der Definition des mqweb-Servers ausgeführt wurde.

2. Suchen Sie nach dem neuesten CWWKT0016I-Nachrichtencode, der mit `ibmmq/console` endet. Die URL ist in dieser Nachricht enthalten.

Anmerkung: Die IBM MQ Console verwendet die interne URL `/ibmmq/console/internal`. Diese URL wird in einigen CWWKT0016I-Nachrichten angezeigt, darf aber nur von der IBM MQ Console verwendet werden.

Beispiel

Das folgende Beispiel zeigt den Nachrichtencode CWWKT0016I mit der URL als Standard-URL:

A CWWKT0016I: Web application available (default_host): https://localhost:9443/ibmmq/console

z/OS **V 9.0.1** **Einschränkungen unter z/OS**

Es gelten die folgenden Einschränkungen, wenn die IBM MQ Console unter z/OS zum Verwalten von Warteschlangenmanagern verwendet wird.

- Warteschlangenmanager unter z/OS können nicht erstellt, gelöscht, gestartet oder gestoppt werden.
- Kanalinitiatoren unter z/OS können nicht gestartet oder gestoppt werden und der Kanalinitiatorstatus wird nicht angezeigt.
- Listeners können nicht angezeigt oder verwaltet werden.
- Befehle zum Starten, Ping, Auflösen und Zurücksetzen von Kanälen können nur mit CHLDISP (DEFAULT) ausgegeben werden.
- Neue Objekte können nur mit QSGDISP (QMGR) erstellt werden.
- Objekte, die mit QSGDISP (GROUP) definiert sind, können nicht angezeigt oder verwaltet werden.
- Die WS-Manager-Sicherheit kann nicht verwaltet werden.
- Die Systemressourcennutzung kann nicht überwacht werden.

Zugehörige Tasks

„Verwaltung über die IBM MQ Console“ auf Seite 94

Die IBM MQ Console ist eine webbasierte Benutzerschnittstelle, über die allgemeine Verwaltungsaufgaben ausgeführt werden können.

Zugehörige Informationen

Mit lokalen Warteschlangenmanagern arbeiten

V 9.0.1 **Mit lokalen Warteschlangenmanagern arbeiten**

Mit dem Widget für lokale Warteschlangenmanager in der IBM MQ Console können lokale Warteschlangenmanager erstellt, konfiguriert und gesteuert werden.

Informationen zu diesem Vorgang

Das Widget für lokale Warteschlangenmanager listet die lokalen Warteschlangenmanager auf, die zu der IBM MQ-Installation hinzugefügt werden, aus der die IBM MQ Console ausgeführt wird. Warteschlangenmanager, die anderen Installationen von IBM MQ auf demselben System zugeordnet sind, werden nicht aufgelistet. Sie können einzelne Warteschlangenmanager aus der Liste auswählen, um mit ihnen zu arbeiten.

Anmerkung: **V 9.0.4** Die IBM MQ Console unterstützt keine Warteschlangenmanager für replizierte Daten (Replicated Data Queue Managers, RDQMs).

Sie können ein Widget für lokale Warteschlangenmanager zu Ihrem Dashboard hinzufügen. Klicken Sie


dazu auf **Widget hinzufügen** **V 9.0.5** **Add widget**. Wählen Sie dann **Lokale Warteschlangenmanager** aus.

V 9.0.5 Sie können das Widget nach seiner Erstellung umbenennen. Wenn Sie den Mauszeiger in








der Titelleiste bewegen, wird das Symbol zum Umbenennen angezeigt. Klicken Sie auf

das Symbol, geben Sie im Fenster 'Widget umbenennen' einen neuen Namen ein und klicken Sie auf **Umbenennen**.

 Unter z/OS können keine Warteschlangenmanager erstellt, gestartet, gestoppt oder gelöscht werden.

Prozedur

- So erstellen Sie einen neuen lokalen Warteschlangenmanager:
 - a) Klicken Sie in der Symbolleiste des Widgets für lokale Warteschlangenmanager auf das Symbol  .
 - b) Geben Sie einen Namen für den neuen Warteschlangenmanager ein. Der Name kann bis zu 48 Zeichen enthalten. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ":", "/", "_ " und "%".
 - c) Optional: Geben Sie einen verfügbaren TCP/IP-Port ein, an dem der Warteschlangenmanager empfangsbereit sein soll. Die Port-Nummer darf 65535 nicht überschreiten.
 - d) Klicken Sie auf **Erstellen** . Der neue Warteschlangenmanager wird erstellt und gestartet.
- So starten Sie einen lokalen Warteschlangenmanager:
 - a) Wählen Sie den Warteschlangenmanager, der gestartet werden soll, aus der Liste im Widget für lokale Warteschlangenmanager aus.
 - b) Klicken Sie in der Symbolleiste des Widgets für den lokalen Warteschlangenmanager auf das Startsymbol  .
- So stoppen Sie einen lokalen Warteschlangenmanager:
 - a) Wählen Sie den Warteschlangenmanager, der gestoppt werden soll, aus der Liste im Widget für lokale Warteschlangenmanager aus.
 - b) Klicken Sie in der Symbolleiste des Widgets für lokale Warteschlangenmanager auf das Stoppsymbol  .
 - c) Bestätigen Sie durch Klicken auf **Stoppen**, dass der Warteschlangenmanager gestoppt werden soll.
- So löschen Sie einen lokalen Warteschlangenmanager:
 - a) Wählen Sie den Warteschlangenmanager, der gelöscht werden soll, aus der Liste im Widget für lokale Warteschlangenmanager aus.
 - b) Falls der Warteschlangenmanager aktiv ist, stoppen Sie ihn.
 - c) Klicken Sie in der Symbolleiste des Widgets für lokale Warteschlangenmanager auf das Löschsymbolsymbol  .
 - d) Bestätigen Sie durch Klicken auf **Löschen**, dass der Warteschlangenmanager gelöscht werden soll. Der Warteschlangenmanager und alle zugeordneten Objekte werden gelöscht.
- So können Sie die Eigenschaften eines lokalen Warteschlangenmanagers anzeigen und bearbeiten:
 - a) Stellen Sie sicher, dass der Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.
 - b) Klicken Sie in der Symbolleiste des Widgets für lokale Warteschlangenmanager auf das Eigenschaftensymbol  . Alternativ können Sie auf den Warteschlangenmanager doppelklicken.
 - c) Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile

bearbeitet werden. Informationen zu einer Eigenschaft finden Sie im Abschnitt Eigenschaften des Warteschlangenmanagers in der MQ Explorer-Dokumentation.

- So aktualisieren Sie die Sicherheit für den lokalen Warteschlangenmanager:
 - a) Stellen Sie sicher, dass der lokale Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.
 - b) Auswählen ... > **Sicherheit aktualisieren**
 - c) Wählen Sie die zu aktualisierende Warteschlangenmanagersicherheit aus:
 - Wählen Sie **Berechtigungsservice** aus, um die Berechtigungsliste zu aktualisieren, die intern von der Berechtigungsservicekomponente verwaltet wird.
 - Wählen Sie **Verbindungsauthentifizierung** aus, um die zwischengespeicherte Ansicht der Konfiguration für Verbindungsauthentifizierung zu aktualisieren.
 - Wählen Sie **SSL** aus, um die zwischengespeicherte Ansicht des SSL- oder TLS-Schlüsselrepositoriums zu aktualisieren. Diese Option aktualisiert auch die Standorte der LDAP-Server, die für zertifizierte Widerruflisten verwendet werden, sowie alle Verschlüsselungshardwareparameter.
- So arbeiten Sie mit Berechtigungssätzen für den lokalen Warteschlangenmanager:
 - a) Stellen Sie sicher, dass der lokale Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.
 - b) Wählen Sie eine der folgenden Optionen aus:
 - Auswählen ... > **Berechtigungsdatensätze verwalten** , um mit den Berechtigungsdatensätzen für den Warteschlangenmanager zu arbeiten und anzugeben, welche Aktionen Gruppen von Benutzern ausführen können.
 - Auswählen ... > **Erstellungsberechtigungsätze verwalten** , um mit den Erstellungsberechtigungsätzen für den Warteschlangenmanager zu arbeiten und anzugeben, welche Objektgruppen von Benutzern auf diesem Warteschlangenmanager erstellt werden können.
- So erstellen Sie automatisch eine Dashboardregisterkarte für einen lokalen Warteschlangenmanager:
 - a) Wählen Sie den Warteschlangenmanager im Widget für lokale Warteschlangenmanager aus.
 - b) Auswählen ... > **Neue Registerkarte 'Dashboard' hinzufügen**
Es wird eine neue Dashboardregisterkarte erstellt. Die Registerkarte erhält den Namen des Warteschlangenmanagers.
- So filtern Sie die Liste der lokalen Warteschlangenmanager:
 - a) Geben Sie den Filtertext im Suchfeld ein.
 - b) Stoppen Sie das Filtern, indem Sie den Text aus dem Suchfeld löschen.

V 9.0.1 Mit IBM MQ-Objekten arbeiten

Die Widgets für IBM MQ-Objekte in der IBM MQ Console ermöglichen die Arbeit mit verschiedenen Typen von IBM MQ-Objekten.

Informationen zu diesem Vorgang

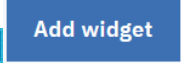
Jedes Widget für IBM MQ-Objekte enthält Objekte, die einem bestimmten Warteschlangenmanager zugeordnet sind. Sie können folgende IBM MQ-Objektwidgets zu Ihrem Dashboard hinzufügen:

- Warteschlangenwidget
- Themenwidget
- Widget für Empfangsprogramme
- Widget für Kanäle
- Widget für Clientverbindungskanäle
- Widget für Authentifizierungsinformationen

- Widget für Subskriptionen

Prozedur

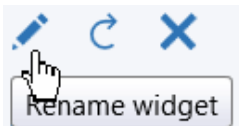
- So erstellen Sie ein Widget für IBM MQ-Objekte:

- Klicken Sie auf **Widget hinzufügen**  .
- Wählen Sie den zutreffenden Warteschlangenmanager aus der Liste aus.
- Klicken Sie auf den Namen des Objektwidgets, das Sie erstellen möchten.

- 

So benennen Sie das Objektwidget um:

- Wenn Sie den Mauszeiger in der Titelleiste bewegen, wird das Symbol zum Umbenennen angezeigt




. Klicken Sie auf das Symbol.

- Geben Sie im Fenster 'Widget umbenennen' einen neuen Namen ein und klicken Sie auf **Umbenennen**.

- 



So konfigurieren Sie das IBM MQ-Objektwidget:



- Klicken Sie in der Titelleiste des Widgets auf das Symbol zum Konfigurieren .
- Optional: Geben Sie den Warteschlangenmanager an, für den die IBM MQ-Objekte angezeigt werden.
- Optional: Geben Sie an, ob Systemobjekte angezeigt oder ausgeblendet werden.
- Klicken Sie auf **Speichern**.

- So filtern Sie die Objekte, die im Widget angezeigt werden:

- Geben Sie den Filtertext im Suchfeld ein.
- Stoppen Sie das Filtern, indem Sie den Text aus dem Suchfeld löschen.

- Wenn Sie den Inhalt des Widgets aktualisieren möchten, klicken Sie auf das Aktualisierungssymbol

  in der Titelleiste des Widgets.

- Klicken Sie zum Entfernen des Widgets auf das Löschsymbolsymbol   in der Titelleiste des Widgets.

Mit Warteschlangen arbeiten


Mit dem Warteschlangenwidget in der IBM MQ Console können die Warteschlangen, die es für einen bestimmten Warteschlangenmanager gibt, angezeigt werden. Sie haben dann die Möglichkeit, Warteschlangen hinzuzufügen oder zu löschen, Nachrichten zu einer Warteschlange hinzuzufügen oder daraus zu entfernen, Nachrichten anzuzeigen, die Eigenschaften einer Warteschlange anzuzeigen und festzulegen und die Berechtigungssätze einer Warteschlange zu verwalten.

Vorbereitende Schritte






Um ein Warteschlangenwidget verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „[Mit IBM MQ-Objekten arbeiten](#)“ auf Seite 100.

Informationen zu diesem Vorgang

Das Warteschlangenwidget listet die Warteschlangen auf, die es für einen bestimmten Warteschlangenmanager gibt. Sie können einzelne Warteschlangen aus der Liste auswählen, um mit ihnen zu arbeiten.

 Für Warteschlangen unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

Prozedur

- So fügen Sie eine Warteschlange hinzu:
 - a) Klicken Sie in der Symbolleiste des Warteschlangenwidgets auf das Symbol 'Erstellen'

 - b) Geben Sie einen Namen für die Warteschlange ein. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_ " und "%".
 - c) Wählen Sie den Typ der Warteschlange aus, die hinzugefügt werden soll.
 - d) Klicken Sie auf **Erstellen** . Die neue Warteschlange wird erstellt.
- So reihen Sie Nachrichten in eine Warteschlange ein:
 - a) Wählen Sie die Warteschlange, zu der Sie Nachrichten hinzufügen möchten, aus der Liste im Warteschlangenwidget aus. Es kann keine Modellwarteschlange ausgewählt werden.
 - b) Klicken Sie in der Symbolleiste des Widgets für Warteschlangen auf das Symbol zum Einreihen von Nachrichten

 - c) Geben Sie die Nachricht ein, die Sie in die Warteschlange einreihen möchten.
 - d) Klicken Sie auf **Einreihen**.
- So entfernen Sie Nachrichten aus einer Warteschlange:
 - a) Wählen Sie die lokale Warteschlange, aus der Sie Nachrichten entfernen möchten, aus der Liste im Warteschlangenwidget aus.
 - b) Auswählen  ... > **Warteschlange löschen**.
 - c) Bestätigen Sie durch Klicken auf **Warteschlange leeren**, dass die Warteschlange geleert werden soll.
- So zeigen Sie Nachrichten in einer Warteschlange an:
 - a) Wählen Sie die lokale oder Aliaswarteschlange, deren Nachrichten Sie anzeigen möchten, aus der Liste im Warteschlangenwidget aus.
 - b) Klicken Sie in der Symbolleiste des Warteschlangenwidgets auf das Suchsymbol  . Das Nachrichtenfenster wird geöffnet, in dem die Nachrichten, die in der Warteschlange enthalten sind, angezeigt werden.
- So löschen Sie eine Warteschlange:
 - a) Wählen Sie die Warteschlange, die Sie löschen möchten, aus der Liste im Warteschlangenwidget aus.
 - b) Klicken Sie in der Symbolleiste des Warteschlangenwidgets auf das Löschsymboll

 - c) Optional: Falls die Warteschlange Nachrichten enthält, bestätigen Sie durch Klicken auf **Warteschlange leeren**, dass die Warteschlange geleert werden kann.
 - d) Bestätigen Sie durch Klicken auf **Löschen**, dass die Warteschlange gelöscht werden soll. Die Warteschlange wird gelöscht.
- So können Sie die Eigenschaften einer Warteschlange anzeigen und bearbeiten:

- a) Wählen Sie die Warteschlange im Warteschlangenwidget aus.
- b) Klicken Sie in der Symbolleiste des Warteschlangenwidgets auf das Eigenschaftensymbol




. Alternativ können Sie auf die Warteschlange doppelklicken.

- c) 

Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Informationen zu einer Eigenschaft finden Sie im Abschnitt [Warteschlangeneigenschaften](#) in der MQ Explorer-Dokumentation.

- So können Sie Berechtigungssätze für eine Warteschlange anzeigen und bearbeiten:

- a) Wählen Sie die Warteschlange im Widget aus.
- b) Klicken Sie auf  ... > **Berechtigungssätze verwalten**.

Die Berechtigungssätze zeigen die Berechtigungen an, die Benutzer und Administratoren für die ausgewählte Warteschlange besitzen.

Mit Themen arbeiten

Mit dem Themenwidget in der IBM MQ Console können Themen hinzugefügt und gelöscht sowie die Eigenschaften eines Themas angezeigt und festgelegt werden.

Vorbereitende Schritte

Um ein Themenwidget verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt [„Mit IBM MQ-Objekten arbeiten“](#) auf Seite 100.

Informationen zu diesem Vorgang

Das Themenwidget listet die Themen auf, die es für einen bestimmten Warteschlangenmanager gibt. Sie können einzelne Themen aus der Liste auswählen, um mit ihnen zu arbeiten.



Für Themen unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

Prozedur

- So fügen Sie ein Thema hinzu:

- a) Klicken Sie in der Symbolleiste des Themenwidgets auf das Erstellungssymbol .

- b) Geben Sie einen Namen für das neue Thema ein. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_" und "%".

- c) Geben Sie die Themenzeichenfolge an, an die Sie Nachrichten für das Thema veröffentlichen. Weitere Informationen finden Sie unter [Themeneigenschaften](#).

- d) Klicken Sie auf **Erstellen**. Das neue Thema wird erstellt.

- So löschen Sie ein Thema:

- a) Wählen Sie das Thema, das Sie löschen möchten, aus der Liste im Themenwidget aus.

- b) Klicken Sie in der Symbolleiste des Themenwidgets auf das Löschsymbolsymbol .

- c) Bestätigen Sie durch Klicken auf **Löschen**, dass das Thema gelöscht werden soll. Das Thema wird gelöscht.

- So können Sie die Eigenschaften eines Themas anzeigen und bearbeiten:

- a) Wählen Sie das Thema im Themenwidget aus.

b) Klicken Sie in der Symbolleiste des Themenwidgets auf das Symbol 'Eigenschaften'



. Alternativ können Sie auf das Thema doppelklicken.

c) 

Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Informationen zu einer Eigenschaft finden Sie in den Eigenschaftsinformationen in [Themeneigenschaften](#) in der MQ Explorer-Dokumentation.

• So veröffentlichen Sie eine Nachricht für ein Thema:

a) Klicken Sie in der Symbolleiste des Themenwidgets auf das Symbol für das Einreihen von Nachrichten



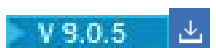
b) Geben Sie eine Nachricht im Feld **Nachricht** ein.

c) Geben Sie im Feld **Themenzeichenfolge** die Themenzeichenfolge ein, für die die Nachricht veröffentlicht werden soll.

d) Klicken Sie auf **Publish** (Veröffentlichen).

• So abonnieren Sie ein Thema:

a) Klicken Sie in der Symbolleiste des Themenwidgets auf das Symbol 'Subskribieren'



b) Geben Sie im Feld **Themenzeichenfolge** die Themenzeichenfolge ein, die Sie abonnieren möchten.

c) Klicken Sie auf **Subscribe** (Subskribieren).

• So können Sie Berechtigungssätze für ein Thema anzeigen und bearbeiten:

a) Wählen Sie das Thema im Themenwidget aus.

b) Klicken Sie auf  ... > **Berechtigungssätze verwalten**.

Die Berechtigungssätze zeigen die Berechtigungen an, die Benutzer und Administratoren für das ausgewählte Thema besitzen.

Mit Empfangsprogrammen arbeiten

Mit dem Widget für Empfangsprogramme in der IBM MQ Console können Empfangsprogramme hinzugefügt und gelöscht, Empfangsprogramme gestartet und gestoppt, Empfangsprogrammeigenschaften angezeigt und festgelegt sowie Berechtigungssätze für ein Empfangsprogramm verwaltet werden.

Vorbereitende Schritte

Um ein Widget für Empfangsprogramme verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt [„Mit IBM MQ-Objekten arbeiten“](#) auf Seite 100.

Informationen zu diesem Vorgang

Das Widget für Empfangsprogramme listet die Empfangsprogramme auf, die es für einen bestimmten Warteschlangenmanager gibt. Sie können einzelne Empfangsprogramme aus der Liste auswählen, um mit ihnen zu arbeiten.



Unter z/OS kann das Widget für Empfangsprogramme nicht verwendet werden.

Prozedur

• So fügen Sie ein TCP/IP-Empfangsprogramm hinzu:

a) Klicken Sie in der Symbolleiste des Listener-Widgets auf das Erstellungssymbol



b) Geben Sie einen Namen für das Empfangsprogramm ein. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_ " und "%".

c) Geben Sie einen verfügbaren TCP/IP-Port für das Empfangsprogramm ein. Die Port-Nummer darf 65535 nicht überschreiten.

d) Klicken Sie auf **Erstellen** . Das neue Empfangsprogramm wird erstellt.

• So löschen Sie ein Empfangsprogramm:


a) Wählen Sie das Empfangsprogramm, das Sie löschen möchten, aus der Liste im Widget für Empfangsprogramme aus.

b) Klicken Sie in der Symbolleiste des Listener-Widgets auf das Löschsymbolsymbol .

c) Bestätigen Sie durch Klicken auf **Löschen**, dass das Empfangsprogramm gelöscht werden soll. Das Empfangsprogramm wird gelöscht.


• So starten Sie ein Empfangsprogramm:

a) Wählen Sie das Empfangsprogramm, das Sie starten möchten, aus der Liste im Widget für Empfangsprogramme aus.

b) Klicken Sie in der Symbolleiste des Listener-Widgets auf das Startsymbol .

• So stoppen Sie ein Empfangsprogramm:

a) Wählen Sie das Empfangsprogramm, das Sie stoppen möchten, aus der Liste im Widget für Empfangsprogramme aus.

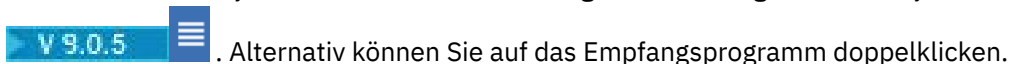
b) Klicken Sie in der Symbolleiste des Listener-Widgets auf das Stoppsymbol .


c) Bestätigen Sie durch Klicken auf **Stoppen**, dass das Empfangsprogramm gestoppt werden soll.

• So können Sie die Eigenschaften eines Empfangsprogramms anzeigen und bearbeiten:

a) Wählen Sie im Widget das Empfangsprogramm aus.

b) Klicken Sie in der Symbolleiste des Listener-Widgets auf das Eigenschaftensymbol



c)  Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Informationen zu einer Eigenschaft finden Sie in den Eigenschaftsinformationen in [Listener-Eigenschaften](#) in der MQ Explorer-Dokumentation.

• So können Sie Berechtigungssätze für ein Empfangsprogramm anzeigen und bearbeiten:

a) Wählen Sie im Widget das Empfangsprogramm aus.

b) Klicken Sie auf  ... > **Berechtigungssätze verwalten**. Die Berechtigungssätze zeigen die Berechtigungen an, die Benutzer und Administratoren für das ausgewählte Empfangsprogramm besitzen.

Mit Kanälen arbeiten

Mit dem Widget für Kanäle in der IBM MQ Console können Kanäle hinzugefügt und gelöscht, Kanäle gestartet und gestoppt, Kanäle zurückgesetzt und aufgelöst sowie Ping-Signale an Kanäle abgesetzt werden. Außerdem können die Eigenschaften eines Kanals angezeigt und bearbeitet und Berechtigungssätze für den Kanal verwaltet werden.

Vorbereitende Schritte





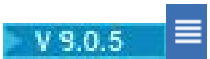

Um ein Widget für Kanäle verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „[Mit IBM MQ-Objekten arbeiten](#)“ auf Seite 100.

Informationen zu diesem Vorgang

Das Widget für Kanäle listet die Kanäle auf, die es für einen bestimmten Warteschlangenmanager gibt. Sie können einzelne Kanäle aus der Liste auswählen, um mit ihnen zu arbeiten.

 Für Kanäle unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

Prozedur

- So fügen Sie einen Kanal hinzu:
 - a) Klicken Sie in der Symbolleiste des Widgets für Kanäle auf das Erstellungssymbol .
 - b) Geben Sie einen Namen für den Kanal ein. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_" und "%".
 - c) Wählen Sie den Typ des Kanals aus, der hinzugefügt werden soll.
 - d) Wenn Sie einen Sender-, Clustersender- oder Requesterkanal erstellen, geben Sie den Verbindungsnamen an. Der Verbindungsname ist der Name des Computers, der den Zielwarteschlangenmanager hostet. Der Name hat das Format *Computername(Portnummer)*. *Computername* steht für den Namen oder die IP-Adresse des Computers, der den Zielwarteschlangenmanager hostet, und *Portnummer* für die Nummer des Ports, den das Empfangsprogramm des Warteschlangenmanagers verwendet.
 - e) Wenn Sie einen Senderkanal oder einen Serverkanal erstellen, geben Sie die Übertragungswarteschlange an, die dem Warteschlangenmanager auf der Empfängerseite des Kanals entspricht.
 - f) Klicken Sie auf **Erstellen**. Der neue Kanal wird erstellt.
- So löschen Sie einen Kanal:
 - a) Wählen Sie den Kanal, den Sie löschen möchten, aus der Liste im Widget für Kanäle aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Löschsymbolsymbol .
 - c) Bestätigen Sie durch Klicken auf **Löschen**, dass der Kanal gelöscht werden soll. Der Kanal wird gelöscht.
- Gehen Sie wie folgt vor, um einen Kanal zu starten:
 - a) Wählen Sie den Kanal, den Sie starten möchten, aus der Liste im Widget für Kanäle aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Startsymbol .
- So stoppen Sie einen Kanal:
 - a) Wählen Sie den Kanal, den Sie stoppen möchten, aus der Liste im Widget für Kanäle aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Stoppsymbol .
 - c) Bestätigen Sie durch Klicken auf **Stoppen**, dass der Kanal gestoppt werden soll.
- So zeigen Sie die Eigenschaften eines Kanals an:
 - a) Wählen Sie den Kanal im Widget für Kanäle aus.
 - b) Klicken Sie in der Symbolleiste des Widgets für Kanäle auf das Eigenschaftensymbol . Alternativ können Sie auf den Kanal doppelklicken.
 - c) 

Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Informationen zu einer Eigenschaft finden Sie in den Eigenschaftsinformationen in [Kanaleigenschaften](#) in der MQ Explorer-Dokumentation.

- So setzen Sie einen Kanal zurück:
 - a) Wählen Sie den Kanal im Widget für Kanäle aus.
 - b) Klicken Sie auf **V 9.0.5** ... > **Zurücksetzen**.
 - c) Geben Sie eine Nachrichtenfolgennummer an.

Sie müssen einen Kanal zurücksetzen, wenn er nicht gestartet wird, weil sich die beiden Enden des Kanals nicht über die Folgennummer der nächsten zu sendenden Nachricht einig sind. Die Nachrichtenfolgennummer gibt diese Nummer an.
 - d) Klicken Sie auf **Kanal zurücksetzen**.
- So lösen Sie einen Kanal auf:
 - a) Wählen Sie den Kanal im Widget für Kanäle aus.
 - b) Klicken Sie auf **V 9.0.5** ... > **Auflösen**.
 - c) Geben Sie an, ob der aktuelle Nachrichtenstapel festgeschrieben oder zurückgestellt werden soll, indem Sie auf **Festschreiben** oder **Zurückstellen** klicken.
- So setzen Sie ein Pingsignal an einen Kanal ab:
 - a) Wählen Sie den Kanal im Widget für Kanäle aus.
 - b) Klicken Sie auf **V 9.0.5** ... > **Ping**.
- So können Sie Berechtigungssätze für einen Kanal anzeigen oder bearbeiten:
 - a) Wählen Sie den Kanal im Widget aus.
 - b) Klicken Sie auf **V 9.0.5** ... > **Berechtigungssätze verwalten**.

Die Berechtigungssätze zeigen die Berechtigungen an, die Benutzer und Administratoren für den ausgewählten Kanal besitzen.

V 9.0.1 Mit Clientverbindungskanälen arbeiten

Mit dem Widget für Clientverbindungskanäle in der IBM MQ Console können Clientverbindungskanäle auf einem Warteschlangenmanager hinzugefügt und gelöscht, die Eigenschaften angezeigt und festgelegt sowie die Berechtigungssätze für den Kanal verwaltet werden.

Vorbereitende Schritte

Um ein Widget für Clientverbindungskanäle verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt [„Mit IBM MQ-Objekten arbeiten“](#) auf Seite 100.

Informationen zu diesem Vorgang

Das Widget für Clientverbindungskanäle listet die Clientverbindungskanäle auf, die es für einen bestimmten Warteschlangenmanager gibt. Sie können einzelne Clientverbindungskanäle aus der Liste auswählen, um mit ihnen zu arbeiten.

z/OS Für Clientverbindungskanäle unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

Prozedur

- So fügen Sie einen Clientverbindungskanal hinzu:

a) Klicken Sie in der Symbolleiste des Widgets für Clientverbindungskanäle auf das Erstellungssymbol




b) Geben Sie einen Namen für den Clientverbindungskanal ein. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ":", "/", "_", und "%".

c) Geben Sie den Verbindungsnamen an. Der Verbindungsname ist der Name des Computers, der den Zielwarteschlangenmanager hostet. Das Format lautet *Computername(Portnummer)*. Datei steht *Computername* für den Namen oder die IP-Adresse des Computers, auf dem sich der Zielwarteschlangenmanager befindet, und *Portnummer* ist der Port, das das Empfangsprogramm des Zielwarteschlangenmanager verwendet.

d) Klicken Sie auf **Erstellen** . Der neue Clientverbindungskanal wird erstellt.

• So löschen Sie einen Clientverbindungskanal:

a) Wählen Sie den Clientverbindungskanal, den Sie löschen möchten, aus der Liste im Widget für Clientverbindungskanäle aus.

b) Klicken Sie in der Widgetsymbolleiste auf das Löschsymbolsymbol  .

c) Bestätigen Sie durch Klicken auf **Löschen**, dass der Clientverbindungskanal gelöscht werden soll. Der Clientverbindungskanal wird gelöscht.

• So können Sie die Eigenschaften eines Clientverbindungskanals anzeigen und bearbeiten:

a) Wählen Sie den Clientverbindungskanal im Clientverbindungskanäle-Widget aus.

b) Klicken Sie in der Symbolleiste des Clientverbindungskanäle-Widgets auf das Eigenschaftensymbol




. Alternativ können Sie auf den Clientverbindungskanal doppelklicken.

c) 

Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Informationen zu einer Eigenschaft finden Sie in den Eigenschaftsinformationen in [Kanaleigenschaften](#) in der MQ Explorer-Dokumentation.

• So können Sie Berechtigungssätze für einen Clientverbindungskanal anzeigen und bearbeiten:

a) Wählen Sie den Clientverbindungskanal im Clientverbindungskanäle-Widget aus.

b) Klicken Sie auf  ... > **Berechtigungssätze verwalten** Die Berechtigungssätze zeigen die Berechtigungen an, die Benutzer und Administratoren für den ausgewählten Clientverbindungskanal besitzen.

Mit Authentifizierungsinformationen arbeiten

Mit dem Widget für Authentifizierungsinformationen in der IBM MQ Console können die Authentifizierungsinformationsobjekte auf einem Warteschlangenmanager hinzugefügt und gelöscht werden. Außerdem können die Eigenschaften angezeigt und festgelegt und die Berechtigungssätze für die Objekte verwaltet werden.


Vorbereitende Schritte

Um ein Widget für Authentifizierungsinformationen verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „[Mit IBM MQ-Objekten arbeiten](#)“ auf Seite 100.



Informationen zu diesem Vorgang

Das Widget für Authentifizierungsinformationen listet die Authentifizierungsinformationen auf, die es für einen bestimmten Warteschlangenmanager gibt. Sie können einzelne Authentifizierungsinformationen aus der Liste auswählen, um mit ihnen zu arbeiten.

Die Warteschlangenmanager-Authentifizierungsinformationen sind Bestandteil der IBM MQ-Unterstützung für Transport Layer Security (TLS). Die Objekte enthalten die Definitionen, die zur Durchführung der Überprüfung des Zertifikatswiderrufs mithilfe von OCSP oder Zertifikatswiderrufslisten (CRLs) auf LDAP-Servern erforderlich sind, sowie die zur Aktivierung der Benutzer-ID- und Kennwortprüfung erforderlichen Definitionen.

 Unter z/OS kann IDPW LDAP nicht verwendet werden und es können auch keine Berechtigungssätze für Authentifizierungsinformationsobjekte angezeigt oder bearbeitet werden.


Prozedur

- So fügen Sie ein Authentifizierungsinformationsobjekt hinzu:
 - a) Klicken Sie auf das Erstellungssymbol  in der Symbolleiste des Widgets für Authentifizierungsinformationen.
 - b) Geben Sie den Namen des Authentifizierungsinformationsobjekts an. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_", "%".
 - c) Geben Sie den Typ des Authentifizierungsinformationsobjekts an.
 - d) Geben Sie zusätzliche objekttypspezifische Informationen an:
 - Geben Sie für **CRL LDAP** den **LDAP-Servernamen** an. Dies ist der Hostname (eine IPv4-Adresse in der Schreibweise mit Trennzeichen oder eine IPv6-Adresse in Hexadezimalschreibweise) des Hosts, auf dem der LDAP-Server aktiv ist. Optional kann zusätzlich eine Portnummer angegeben werden.
 - Geben Sie für **OCSP** die **OCSP-Responder-URL** an. Dies ist die URL, die zur Überprüfung des Zertifikatswiderrufs verwendet wird. Dieser Wert muss eine HTTP-URL mit dem Hostnamen und der Portnummer des OCSP-Responder sein. Wenn der OCSP-Responder den HTTP-Standardport 80 verwendet, muss die Portnummer nicht angegeben werden. HTTP-URLs sind in RFC 1738 definiert.
 - Für **IDPW OS** gelten keine zusätzlichen Anforderungen.
 - Geben Sie für **IDPW LDAP** den **LDAP-Servernamen** und den **Benutzerkurznamen** an. Der LDAP-Servername ist der Hostname (eine IPv4-Adresse in der Schreibweise mit Trennzeichen oder eine IPv6-Adresse in Hexadezimalschreibweise) des Hosts, auf dem der LDAP-Server aktiv ist. Optional kann zusätzlich eine Portnummer angegeben werden. Der Benutzerkurzname ist das Feld im LDAP-Benutzerdatensatz, das als Kurzname für die Verbindung verwendet wird.
 - e) Klicken Sie auf **Erstellen**.
- So löschen Sie ein Authentifizierungsinformationsobjekt:
 - a) Wählen Sie das Authentifizierungsinformationsobjekt, das Sie löschen möchten, aus der Liste im Widget aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Löschsymbolsymbol .
 - c) Bestätigen Sie durch Klicken auf **Löschen**, dass das Authentifizierungsinformationsobjekt gelöscht werden soll. Das Objekt wird gelöscht.
- So können Sie die Eigenschaften eines Authentifizierungsinformationsobjekts anzeigen und bearbeiten:
 - a) Wählen Sie das Authentifizierungsinformationsobjekt im Widget aus.

b) Klicken Sie in der Widgetsymbolleiste auf das Eigenschaftensymbol



. Alternativ können Sie auf das Authentifizierungsinformationsobjekt doppelklicken.

- c) Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden.
- So können Sie Berechtigungssätze für ein Authentifizierungsinformationsobjekt anzeigen und bearbeiten:
 - a) Wählen Sie das Authentifizierungsinformationsobjekt im Widget für Authentifizierungsinformationen aus.
 - b) Klicken Sie auf  ... > **Berechtigungssätze verwalten** Die Berechtigungssätze zeigen die Berechtigungen an, die Benutzer und Administratoren für das ausgewählte Authentifizierungsinformationsobjekt besitzen.

Mit Subskriptionen arbeiten

Mit dem Widget für Subskriptionen in der IBM MQ Console können Subskriptionen auf einem Warteschlangenmanager hinzugefügt und gelöscht, die Eigenschaften angezeigt und festgelegt sowie die Berechtigungssätze für die Subskriptionen verwaltet werden.

Vorbereitende Schritte

Um ein Widget für Subskriptionen verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „[Mit IBM MQ-Objekten arbeiten](#)“ auf Seite 100.

Informationen zu diesem Vorgang

Subskriptionen werden an einen Warteschlangenmanager ausgegeben und enthalten Informationen zu den Veröffentlichungen, die der Subskribent erhalten möchte:

- Die Themenzeichenfolge, an der der Subskribent interessiert ist; dieses Thema kann in mehrere Themenzeichenfolgen aufgelöst werden, wenn Platzhalterzeichen verwendet werden.
- Eine optionale Auswahlzeichenfolge, die auf die veröffentlichten Nachrichten angewendet wird.
- Der Name der Warteschlange, in die ausgewählte Veröffentlichungen gestellt werden.

Weitere Informationen zu Subskriptionen finden Sie unter [Subskribenten und Subskriptionen](#) und [DEFINE SUB](#).

Prozedur



- So fügen Sie ein Subskriptionsobjekt hinzu:

a) Klicken Sie in der Symbolleiste des Subskriptionswidgets auf das Erstellungssymbol



- b) Geben Sie den Namen des Objekts an. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_ " und "%".
- c) Wählen Sie als **Zielklasse** entweder **Verwaltet** oder **Bereitgestellt** aus. Bei Auswahl von **Verwaltet** wird auf dem lokalen Warteschlangenmanager ein Ziel erstellt.
- d) Geben Sie bei Auswahl der Zielklasse **Bereitgestellt** im Feld **Ziel** den Namen der Warteschlange an, an die Nachrichten für diese Subskription weitergeleitet werden.
- e) Geben Sie im Feld **Themenzeichenfolge** die Themenzeichenfolge an, die subskribiert werden soll.
- f) Wählen Sie eine Einstellung für die **Verwendung von Platzhalterzeichen** aus. Wählen Sie **Platzhalter auf Zeichenebene** aus, um anzugeben, das Platzhalterzeichen für Abschnitte von Zeichenfolgen

stehen. Wählen Sie **Platzhalter auf Themenebene** aus, um anzugeben, dass Platzhalterzeichen für Abschnitte der Themenhierarchie stehen.

- g) Wählen Sie einen **Bereich** aus. Wählen Sie **Alle** aus, damit die Subskription an alle Warteschlangenmanager weitergeleitet wird, die durch einen Publish/Subscribe-Brokerverbund oder eine Hierarchie direkt verbunden sind. Wählen Sie **Warteschlangenmanager** aus, damit die Subskription Nachrichten, die für das Thema veröffentlicht werden, nur innerhalb dieses Warteschlangenmanagers weiterleitet.
 - h) Optional: Geben Sie einen **Selektor** an. Eine Auswahlzeichenfolge ist ein Ausdruck, der auf eine Veröffentlichung angewendet wird, um festzustellen, ob sie mit einer Subskription übereinstimmt.
 - i) Klicken Sie auf **Erstellen** .
- So löschen Sie ein Subskriptionsobjekt:
 - a) Wählen Sie die Subskription, die Sie löschen möchten, aus der Liste im Widget für Subskriptionen aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Löschsymboll  .
 - c) Bestätigen Sie durch Klicken auf **Löschen**, dass das Subskriptionsobjekt gelöscht werden soll. Das Objekt wird gelöscht.
 - So können Sie die Eigenschaften eines Subskriptionsobjekts anzeigen und bearbeiten:
 - a) Wählen Sie das Subskriptionsobjekt im Widget aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Eigenschaftensymbol  . Alternativ können Sie auf das Subskriptionsobjekt doppelklicken.
 - c) Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden.

Mit Kanalauthentifizierungsdatensätzen arbeiten

Mit dem Widget für Kanalauthentifizierungsdatensätze in der IBM MQ Console können Kanalauthentifizierungsdatensätze auf einem Warteschlangenmanager hinzugefügt und gelöscht werden. Außerdem können die Eigenschaften für Kanalauthentifizierungsdatensätze angezeigt und festgelegt werden.

Vorbereitende Schritte

Um ein Widget für Kanalauthentifizierungsdatensätze verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt [„Mit IBM MQ-Objekten arbeiten“](#) auf Seite 100.

Informationen zu diesem Vorgang


Die Zugriffsberechtigungen zum Herstellen von Systemverbindungen auf Kanalebene können mithilfe von Kanalauthentifizierungsdatensätzen gezielter gesteuert werden.

Um Sicherheit zu erzwingen, können Sie mithilfe von Blockierungskanalauthentifizierungsdatensätzen den Zugriff auf Ihre Kanäle sperren. Sie können auch mithilfe von Adresszuordnungskanalauthentifizierungsdatensätzen bestimmten Benutzern den Zugriff ermöglichen. Weitere Informationen zu Kanalauthentifizierungsdatensätzen finden Sie unter [Kanalauthentifizierungsdatensätze](#).


Prozedur

- Wie Sie einen Kanalauthentifizierungsdatensatz mit der Identität 'Definierter SSL-/TLS-Name' hinzufügen, wird im Abschnitt [„Kanalauthentifizierungsdatensätze mit Identität 'Definierter SSL-/TLS-Name' erstellen“](#) auf Seite 112 beschrieben.

- Wie Sie einen Kanalauthentifizierungsdatensatz mit der Identität 'Benutzer-ID der Clientanwendung' hinzufügen, wird im Abschnitt „Kanalauthentifizierungsdatensätze mit einer Benutzer-ID für die Clientanwendungsbenutzer-ID erstellen“ auf Seite 113 beschrieben.
- Wie Sie einen Kanalauthentifizierungsdatensatz mit der Identität 'Name des fernen Warteschlangenmanagers' hinzufügen, wird im Abschnitt „Kanalauthentifizierungsdatensätze mit einer fernen WS-Manager-Namensidentität erstellen“ auf Seite 114 beschrieben.
- Wie Sie einen Kanalauthentifizierungsdatensatz mit der Identität 'Adresse' hinzufügen, wird im Abschnitt „Kanalauthentifizierungsdatensätze mit einer IP-Adressidentität erstellen“ auf Seite 115 beschrieben.
- So löschen Sie einen Kanalauthentifizierungsdatensatz:
 - a) Wählen Sie den Kanalauthentifizierungsdatensatz, den Sie löschen möchten, aus der Liste im Widget für Kanalauthentifizierungsdatensätze aus.

b) Klicken Sie in der Widgetsymbolleiste auf das Löschsymboll  .

- c) Bestätigen Sie durch Klicken auf **Löschen**, dass der Kanalauthentifizierungsdatensatz gelöscht werden soll. Der Kanalauthentifizierungsdatensatz wird gelöscht.
- So können Sie die Eigenschaften eines Kanalauthentifizierungsdatensatzes anzeigen und bearbeiten:
 - a) Wählen Sie den Kanalauthentifizierungsdatensatz, den Sie bearbeiten möchten, aus der Liste im Widget für Kanalauthentifizierungsdatensätze aus.
 - b) Klicken Sie in der Widgetsymbolleiste auf das Eigenschaftensymbol

 . Alternativ können Sie auf den Kanalauthentifizierungsdatensatz doppelklicken.

- c) Zeigen Sie die Eigenschaften an und bearbeiten Sie sie je nach Erfordernis. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden.


Kanalauthentifizierungsdatensätze mit Identität 'Definierter SSL-/TLS-Name' erstellen

Mit dem Widget für Kanalauthentifizierungsdatensätze können Zulassungs-, Blockierungs- und Warnungskanalauthentifizierungsdatensätze mit der Identität 'Definierter SSL-/TLS-Name' erstellt werden. Die Identität 'Definierter SSL-/TLS-Name' wird mit Benutzern abgeglichen, die ein persönliches SSL- oder TLS-Zertifikat vorlegen, das einen festgelegten definierten Namen enthält.

Vorbereitende Schritte

Um ein Widget für Kanalauthentifizierungsdatensätze verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „Mit IBM MQ-Objekten arbeiten“ auf Seite 100.

Prozedur

- So fügen Sie einen Kanalauthentifizierungsdatensatz hinzu:
 - a) Klicken Sie in der Symbolleiste des Widgets für Kanalauthentifizierungsdatensätze auf das Erstellungssymbol  .
 - b) Wählen Sie **Regeltyp** aus, um den gewünschten Regeltyp für den Kanalauthentifizierungsdatensatz anzugeben:
 - Wählen Sie **Zulassen** aus, um den Zugriff auf eingehende Verbindungen zuzulassen.
 - Wählen Sie **Blockieren** aus, um den Zugriff auf eingehende Verbindungen zu blockieren.
 - Wählen Sie **Warnung** aus, um beim Zugriff auf eingehende Verbindungen vor möglichen Blockierungen zu warnen. Der Verbindung wird Zugriff ermöglicht und es wird eine Fehlernachricht

zurückgemeldet. Wenn Ereignisse konfiguriert werden, wird eine Ereignisnachricht erstellt, die im Detail anzeigt, was blockiert werden würde. Es werden nur abgegliche Regeln zurückgemeldet.

- c) Wählen Sie den Identitätstyp **Definierter SSL-/TLS-Name** aus der Liste aus.
- d) Klicken Sie auf **Weiter**.
- e) Geben Sie ein **Kanalprofil** an.

Bei dem Kanalprofil handelt es sich um den Namen des Kanals oder der Kanalgruppe, für den bzw. die die Kanalauthentifizierung festgelegt werden soll. Das Profil kann Platzhalterzeichen enthalten, sodass Sie einen Bereich von Kanälen blockieren können. Beispielsweise blockiert das Profil `a1-phade1ta*` Kanäle mit den Namen `alphadelta1`, `alphadelta2`, `alphdelta3` und so weiter.
- f) Geben Sie den Peernamen an. Beispiel: `CN=John Smith, O=IBM, OU=Test, C=GB`. Weitere Informationen zu Peernamen finden Sie im Abschnitt [WebSphere MQ-Regeln für SSLPEER-Werte](#).
- g) Optional: Geben Sie den verwendeten Filter für die **Adresse** an. Die Adresse ist die IP-Adresse, die am anderen Ende des Kanals erwartet wird.
- h) Optional: Geben Sie einen Namen für **Ausstellernamen für SSL-Zertifikat** an. Dies ist der Name der Zertifizierungsstelle, die das SSL-/TLS-Zertifikat ausgestellt haben muss.
- i) Optional: Klicken Sie auf **Weiter**.
- j) Optional: Für den Regeltyp **Zulassen** können Sie optional die **Benutzerquelle** für den Kanalauthentifizierungsdatensatz angeben. Die Benutzerquelle gibt die Quelle der Benutzer-ID an, die verwendet wird, wenn die eingehende Verbindung mit dem definierten SSL-/TLS-Namen übereinstimmt.
 - Die Option **Kanal** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die variable Benutzer-ID oder einen beliebigen Benutzer, der im Kanalobjekt definiert ist, verwenden.
 - Die Option **Zuordnung** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die Benutzer-ID verwenden, die im Feld **MCA-Benutzer-ID** angegeben ist.
- k) Optional: Klicken Sie auf **Weiter**.
- l) Optional: Geben Sie eine **Beschreibung** für den Kanalauthentifizierungsdatensatz an.
- m) Klicken Sie auf **Erstellen**. Der neue Kanalauthentifizierungsdatensatz wird erstellt.



Kanalauthentifizierungsdatensätze mit einer Benutzer-ID für die Clientanwendungsbenutzer-ID erstellen

Mit dem Widget für Kanalauthentifizierungsdatensätze können Zulassungs-, Blockierungs- und Warnungskanalauthentifizierungsdatensätze mit der Identität 'Benutzer-ID der Clientanwendung' erstellt werden. Die Identität 'Benutzer-ID der Clientanwendung' wird mit Clientanwendungs-IDs von einem Clientverbindungskanal abgeglichen.

Vorbereitende Schritte

Um ein Widget für Kanalauthentifizierungsdatensätze verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt [„Mit IBM MQ-Objekten arbeiten“](#) auf Seite 100.

Prozedur

- So fügen Sie einen Kanalauthentifizierungsdatensatz hinzu:
 - a) Klicken Sie in der Symbolleiste des Widgets für Kanalauthentifizierungsdatensätze auf das Erstellungssymbol  **Create**  .
 - b) Wählen Sie **Regeltyp** aus, um den gewünschten Regeltyp für den Kanalauthentifizierungsdatensatz anzugeben:
 - Wählen Sie **Zulassen** aus, um den Zugriff auf eingehende Verbindungen zuzulassen.

- Wählen Sie **Blockieren** aus, um den Zugriff auf eingehende Verbindungen zu blockieren.
 - Wählen Sie **Warnung** aus, um beim Zugriff auf eingehende Verbindungen vor möglichen Blockierungen zu warnen. Der Verbindung wird Zugriff ermöglicht und es wird eine Fehlermeldung zurückgemeldet. Wenn Ereignisse konfiguriert werden, wird eine Ereignisnachricht erstellt, die im Detail anzeigt, was blockiert werden würde. Es werden nur abgegliche Regeln zurückgemeldet.
- c) Wählen Sie den Identitätstyp **Benutzer-ID der Clientanwendung** aus der Liste aus.
- d) Klicken Sie auf **Weiter**.
- e) Geben Sie ein **Kanalprofil** an.
Bei dem Kanalprofilnamen handelt es sich um den Namen des Kanals oder der Kanalgruppe, für den bzw. die die Kanalauthentifizierung festgelegt werden soll. Das Profil kann Platzhalterzeichen enthalten, sodass Sie einen Bereich von Kanälen blockieren können. Beispielsweise blockiert das Profil `alphadelta*` Kanäle mit den Namen `alphadelta1`, `alphadelta2`, `alphadelta3` und so weiter.
- f) Geben Sie die **Client-Benutzer-ID** an. Die Client-Benutzer-ID ist die Benutzer-ID des Clients, den Sie zulassen oder blockieren möchten oder für den eine Warnung ausgegeben werden soll.
- g) Optional: Geben Sie den verwendeten Filter für die **Adresse** an. Die Adresse ist die IP-Adresse, die am anderen Ende des Kanals erwartet wird.
- h) Optional: Klicken Sie auf **Weiter**.
- i) Optional: Für den Regeltyp **Zulassen** können Sie optional die **Benutzerquelle** für den Kanalauthentifizierungsdatensatz angeben. Die Benutzerquelle gibt die Quelle der Benutzer-ID an, die verwendet wird, wenn die eingehende Verbindung mit der Benutzer-ID der Clientanwendung übereinstimmt.
- Die Option **Kanal** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die variable Benutzer-ID oder einen beliebigen Benutzer, der im Kanalobjekt definiert ist, verwenden.
 - Die Option **Zuordnung** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die Benutzer-ID verwenden, die im Feld **MCA-Benutzer-ID** angegeben ist.
- j) Optional: Klicken Sie auf **Weiter**.
- k) Optional: Geben Sie eine **Beschreibung** für den Kanalauthentifizierungsdatensatz an.
- l) Klicken Sie auf **Erstellen**. Der neue Kanalauthentifizierungsdatensatz wird erstellt.


V 9.0.1 Kanalauthentifizierungsdatensätze mit einer fernen WS-Manager-Namensidentität erstellen

Mit dem Widget für Kanalauthentifizierungsdatensätze können Zulassungs-, Blockierungs- und Warnungskanalauthentifizierungsdatensätze mit der Identität 'Name des fernen Warteschlangenmanagers' erstellt werden. Die Identität 'Name des fernen Warteschlangenmanagers' wird mit dem angegebenen Warteschlangenmanager abgeglichen.

Vorbereitende Schritte

Um ein Widget für Kanalauthentifizierungsdatensätze verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „Mit IBM MQ-Objekten arbeiten“ auf Seite 100.

Prozedur

- So fügen Sie einen Kanalauthentifizierungsdatensatz hinzu:
 - a) Klicken Sie in der Symbolleiste des Widgets für Kanalauthentifizierungsdatensätze auf das Erstellungssymbol .
 - b) Wählen Sie **Regeltyp** aus, um den gewünschten Regeltyp für den Kanalauthentifizierungsdatensatz anzugeben:

- Wählen Sie **Zulassen** aus, um den Zugriff auf eingehende Verbindungen zuzulassen.
 - Wählen Sie **Blockieren** aus, um den Zugriff auf eingehende Verbindungen zu blockieren.
 - Wählen Sie **Warnung** aus, um beim Zugriff auf eingehende Verbindungen vor möglichen Blockierungen zu warnen. Der Verbindung wird Zugriff ermöglicht und es wird eine Fehlernachricht zurückgemeldet. Wenn Ereignisse konfiguriert werden, wird eine Ereignisnachricht erstellt, die im Detail anzeigt, was blockiert werden würde. Es werden nur abgeglichene Regeln zurückgemeldet.
- c) Wählen Sie den Identitätstyp **Name des fernen Warteschlangenmanagers** aus der Liste aus.
- d) Klicken Sie auf **Weiter**.
- e) Geben Sie einen Namen für **Profilname** an.
Der Profilname ist der Name des Kanals oder der Kanalgruppe, für den bzw. die Sie die Kanalauthentifizierung einrichten. Das Profil kann Platzhalterzeichen enthalten, sodass Sie einen Bereich von Kanälen blockieren können. Beispielsweise blockiert das Profil `a1phade1ta*` Kanäle mit den Namen `alphadelat1`, `alphadelat2`, `alphadelat3` und so weiter.
- f) Geben Sie einen Namen für **Warteschlangenmanagername** an. Der Warteschlangenmanagername gibt den Namen des fernen Warteschlangenmanagers an, den Sie zulassen oder blockieren möchten oder für den eine Warnung ausgegeben werden soll.
- g) Optional: Geben Sie den verwendeten Filter für die **Adresse** an. Die Adresse ist die IP-Adresse, die am anderen Ende des Kanals erwartet wird.
- h) Optional: Klicken Sie auf **Weiter**.
- i) Optional: Für den Regeltyp **Zulassen** können Sie optional die **Benutzerquelle** für den Kanalauthentifizierungsdatensatz angeben. Die Benutzerquelle gibt die Quelle der Benutzer-ID an, die verwendet wird, wenn die eingehende Verbindung mit dem Namen des fernen Warteschlangenmanagers übereinstimmt.
- Die Option **Kanal** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die variable Benutzer-ID oder einen beliebigen Benutzer, der im Kanalobjekt definiert ist, verwenden.
 - Die Option **Zuordnung** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die Benutzer-ID verwenden, die im Feld **MCA-Benutzer-ID** angegeben ist.
- j) Optional: Klicken Sie auf **Weiter**.
- k) Optional: Geben Sie eine **Beschreibung** für den Kanalauthentifizierungsdatensatz an.
- l) Klicken Sie auf **Erstellen**. Der neue Kanalauthentifizierungsdatensatz wird erstellt.


V 9.0.1 Kanalauthentifizierungsdatensätze mit einer IP-Adressidentität erstellen

Mit dem Widget für Kanalauthentifizierungsdatensätze können Zulassungs-, Blockierungs- und Warnungskanalauthentifizierungsdatensätze mit der Identität 'IP-Adresse' erstellt werden. Die Identität 'IP-Adresse' wird mit den angegebenen IP-Adressen abgeglichen.

Vorbereitende Schritte

Um ein Widget für Kanalauthentifizierungsdatensätze verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt [„Mit IBM MQ-Objekten arbeiten“](#) auf Seite 100.

Prozedur

- So fügen Sie einen Kanalauthentifizierungsdatensatz hinzu:
 - a) Klicken Sie in der Symbolleiste des Widgets für Kanalauthentifizierungsdatensätze auf das Erstellungssymbol .
 - b) Wählen Sie **Regeltyp** aus, um den gewünschten Regeltyp für den Kanalauthentifizierungsdatensatz anzugeben:

- Wählen Sie **Zulassen** aus, um den Zugriff auf eingehende Verbindungen zuzulassen.
 - Wählen Sie **Blockieren** aus, um den Zugriff auf eingehende Verbindungen zu blockieren.
 - Wählen Sie **Warnung** aus, um beim Zugriff auf eingehende Verbindungen vor möglichen Blockierungen zu warnen. Der Verbindung wird Zugriff ermöglicht und es wird eine Fehlernachricht zurückgemeldet. Wenn Ereignisse konfiguriert werden, wird eine Ereignisnachricht erstellt, die im Detail anzeigt, was blockiert werden würde. Es werden nur abgegliche Regeln zurückgemeldet.
- c) Wählen Sie den Identitätstyp **Adresse** aus der Liste aus.
- d) Klicken Sie auf **Weiter**.
- e) Optional: Geben Sie für den Regeltyp **Blockieren** oder **Warnung** den **Zeitpunkt des Abgleichs** an. Die folgenden Optionen stehen zur Auswahl:
- **Beim Empfangsprogramm**. Bei Auswahl dieser Option wird versucht, die Regel beim Empfangsprogramm abzugleichen.
 - **Beim Kanal**. Bei Auswahl dieser Option wird versucht, die Regel beim Kanal abzugleichen.
- f) Geben Sie einen Namen für **Profilname** an.
Der Profilname ist der Name des Kanals oder der Kanalgruppe, für den bzw. die Sie die Kanalauthentifizierung einrichten. Das Profil kann Platzhalterzeichen enthalten, sodass Sie einen Bereich von Kanälen blockieren können. Beispielsweise blockiert das Profil `a1phade1ta*` Kanäle mit den Namen `alphadelta1`, `alphadelta2`, `alphadelta3` und so weiter.
- g) Geben Sie eine **Adresse** an. Die Adresse ist die IP-Adresse oder eine durch Kommas getrennte Liste von IP-Adressen, die zugelassen oder blockiert werden.
- h) Optional: Klicken Sie auf **Weiter**.
- i) Optional: Für den Regeltyp **Zulassen** können Sie optional die **Benutzerquelle** für den Kanalauthentifizierungsdatensatz angeben. Die Benutzerquelle gibt die Quelle der Benutzer-ID an, die verwendet wird, wenn die eingehende Verbindung mit dem Namen des fernen Warteschlangenmanagers übereinstimmt.
- Die Option **Kanal** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die variable Benutzer-ID oder einen beliebigen Benutzer, der im Kanalobjekt definiert ist, verwenden.
 - Die Option **Zuordnung** gibt an, dass eingehende Verbindungen, die mit der Zuordnung übereinstimmen, die Benutzer-ID verwenden, die im Feld **MCA-Benutzer-ID** angegeben ist.
- j) Optional: Klicken Sie auf **Weiter**.
- k) Optional: Geben Sie eine **Beschreibung** für den Kanalauthentifizierungsdatensatz an.
- l) Klicken Sie auf **Erstellen**. Der neue Kanalauthentifizierungsdatensatz wird erstellt.

V 9.0.1 Kanalauthentifizierungsdatensätze mit einer endgültig zugeordneten Benutzer-ID erstellen

Mit dem Widget für Kanalauthentifizierungsdatensätze können Blockierungs- und Warnungskanalauthentifizierungsdatensätze mit der Identität 'Endgültig zugeordnete Benutzer-ID' erstellt werden. Die Identität 'Endgültig zugeordnete Benutzer-ID' wird mit der Liste angegebener Benutzer-IDs von einem Serverkanal abgeglichen.

Vorbereitende Schritte

Um ein Widget für Kanalauthentifizierungsdatensätze verwenden zu können, müssen Sie es zunächst erstellen. Weitere Informationen zum Erstellen von Widgets für IBM MQ-Objekte finden Sie im Abschnitt „Mit IBM MQ-Objekten arbeiten“ auf Seite 100.

Prozedur

- So fügen Sie einen Kanalauthentifizierungsdatensatz hinzu:


- a) Klicken Sie in der Symbolleiste des Widgets für Kanalauthentifizierungsdatensätze auf das Erstellungssymbol  .
- b) Wählen Sie **Regeltyp** aus, um den gewünschten Regeltyp für den Kanalauthentifizierungsdatensatz anzugeben:
- Wählen Sie **Blockieren** aus, um den Zugriff auf eingehende Verbindungen zu blockieren.
 - Wählen Sie **Warnung** aus, um beim Zugriff auf eingehende Verbindungen vor möglichen Blockierungen zu warnen. Der Verbindung wird Zugriff ermöglicht und es wird eine Fehlernachricht zurückgemeldet. Wenn Ereignisse konfiguriert werden, wird eine Ereignisnachricht erstellt, die im Detail anzeigt, was blockiert werden würde. Es werden nur abgeglichene Regeln zurückgemeldet.
- c) Wählen Sie den Identitätstyp **Endgültig zugeordnete Benutzer-ID** aus der Liste aus.
- d) Klicken Sie auf **Weiter**.
- e) Geben Sie einen Namen für **Profilname** an.
Der Profilname ist der Name des Kanals oder der Kanalgruppe, für den bzw. die Sie die Kanalauthentifizierung einrichten. Das Profil kann Platzhalterzeichen enthalten, sodass Sie einen Bereich von Kanälen blockieren können. Beispielsweise blockiert das Profil `alphaDelta*` Kanäle mit den Namen `alphaDelta1`, `alphaDelta2`, `alphaDelta3` und so weiter.
- f) Geben Sie die **Benutzerliste** an. Die Benutzerliste ist eine durch Kommas getrennte Liste von Benutzer-IDs vom Kanal, die blockiert werden sollen.
- g) Optional: Klicken Sie auf **Weiter**.
- h) Optional: Geben Sie eine **Beschreibung** für den Kanalauthentifizierungsdatensatz an.
- i) Klicken Sie auf **Erstellen** . Der neue Kanalauthentifizierungsdatensatz wird erstellt.

Mit Berechtigungsdatensätzen arbeiten


Sie können den Zugriff von Gruppen auf Warteschlangenmanager und IBM MQ-Objekte steuern, indem Sie für jede Gruppe einen Berechtigungssatz angeben.

Informationen zu diesem Vorgang

Sie können den Zugriff, den eine Gruppe von Messaging-Benutzern auf einen bestimmten Warteschlangenmanager oder ein bestimmtes IBM MQ-Objekt hat, mithilfe von Berechtigungssätzen feinabstimmen. Sie konfigurieren den Berechtigungssatz für alle Objekttypen zwar auf dieselbe Weise mithilfe derselben Prozedur, aber welche Berechtigungen Sie tatsächlich konfigurieren, ist vom jeweiligen Objekttyp abhängig.

Vergleichen Sie beispielsweise die verschiedenen Berechtigungen, die für einen WS-Manager und eine Warteschlange verfügbar sind, wie in den folgenden Abbildungen dargestellt: 

Authority records for 'qm3'

Delete  1 item selected [Cancel](#)

▲ Entity name	Entity type
mqm	Group
mqsystem	User

Total: 2 Last updated: 3:40:18 PM

Administration

- Change
- Delete
- Display
- Ctrl

Context

- Set all context
- Set identity context

MQI

- Alternate user authority
- Connect
- Inquire
- Set
- System

Check all


Uncheck all

Close

Save

V 9.0.5

Authority records for 'q1' on qm3

Delete 		1 item selected	Cancel
▲ Entity name	Entity type		
mqm	Group		
mqsystem	User		



Administration	Context	MQI
<input checked="" type="checkbox"/> Change	<input checked="" type="checkbox"/> Pass all context	<input checked="" type="checkbox"/> Browse
<input checked="" type="checkbox"/> Clear	<input checked="" type="checkbox"/> Pass identity context	<input checked="" type="checkbox"/> Inquire
<input checked="" type="checkbox"/> Delete	<input checked="" type="checkbox"/> Set all context	<input checked="" type="checkbox"/> Get
<input checked="" type="checkbox"/> Display	<input checked="" type="checkbox"/> Set identity context	<input checked="" type="checkbox"/> Put
		<input checked="" type="checkbox"/> Set



Unter z/OS kann nicht mit Berechtigungssätzen gearbeitet werden.

Prozedur

- So können Sie einen Berechtigungssatz für ein IBM MQ-Objekt anzeigen oder bearbeiten:
 - a) Wählen Sie das Objekt in einem Widget im Dashboard aus. Der zugeordnete Warteschlangenmanager muss aktiv sein.
 - b) Wählen Sie in der entsprechenden Widgetsymbolleiste ... > **Berechtigungssätze verwalten**.
 - c) Wählen Sie die Gruppe aus, deren Berechtigungssatz angezeigt werden soll. Die Berechtigungen für die betreffende Gruppe werden angezeigt.
 - d) Wählen Sie erforderliche Berechtigungen aus oder löschen Sie Berechtigungen. Abhängig vom Typ des Objekts, für das Sie einen Berechtigungssatz erstellen, sind unterschiedliche Berechtigungen verfügbar.
 - e) Klicken Sie auf **Speichern**.
- So können Sie eine Berechtigungssatz für Erstellungen für einen Warteschlangenmanager anzeigen oder bearbeiten:
 - a) Wählen Sie den Warteschlangenmanager in einem Widget für Warteschlangenmanager im Dashboard aus. Der WS-Manager muss aktiv sein.
 - b) Wählen Sie in der Widgetsymbolleiste ... > **Berechtigungssätze erstellen verwalten**.
 - c) Wählen Sie die Gruppe aus, deren Berechtigungssatz für Erstellungen angezeigt werden soll. Die Berechtigungen für die betreffende Gruppe werden angezeigt.

- d) Wählen Sie erforderliche Berechtigungen für Erstellungen aus oder löschen Sie Berechtigungen.
- e) Klicken Sie auf **Speichern**.
- So erstellen Sie einen Berechtigungssatz für ein IBM MQ-Objekt:
 - a) Wählen Sie das IBM MQ-Objekt in einem Widget im Dashboard aus. Der zugeordnete Warteschlangenmanager muss aktiv sein.
 - b) Wählen Sie in der Widgetsymbolleiste ... > **Berechtigungsdatensätze verwalten**.
 - c) Klicken Sie auf das Plusymbol .
 - d) Geben Sie den Namen des Benutzers oder der Gruppe an, für den bzw. die Sie den Berechtigungssatz erstellen. Der Benutzer oder die Gruppe muss existieren.
 - e) Wählen Sie den **Entitätstyp** aus, um anzugeben, ob die Entität ein Benutzer oder eine Gruppe ist.
 - f) Klicken Sie auf **Erstellen**.
 - g) Legen Sie durch Auswählen oder Löschen fest, welche Berechtigungen der Benutzer oder die Gruppe besitzen soll. Es sind für jeden Objekttyp unterschiedliche Berechtigungen verfügbar.
 - h) Klicken Sie auf **Speichern**.
- So erstellen Sie einen Berechtigungssatz für die Erstellung von Objekten auf einem Warteschlangenmanager:
 - a) Wählen Sie den Warteschlangenmanager in einem Widget im Dashboard aus. Der WS-Manager muss aktiv sein.
 - b) Wählen Sie in der Widgetsymbolleiste ... > **Berechtigungsdatensätze erstellen verwalten**.
 - c) Klicken Sie auf das Symbol 'Erstellen' .
 - d) Geben Sie den Namen des Benutzers oder der Gruppe an, für den bzw. die Sie den Berechtigungssatz erstellen. Der Benutzer oder die Gruppe muss existieren.
 - e) Wählen Sie den **Entitätstyp** aus, um anzugeben, ob die Entität ein Benutzer oder eine Gruppe ist.
 - f) Klicken Sie auf **Erstellen**.
 - g) Legen Sie durch Auswählen oder Löschen fest, welche Erstellungsrechte der Benutzer oder die Gruppe besitzen soll.
 - h) Klicken Sie auf **Speichern**.

Überwachen der Systemressourcennutzung

Mit dem Widget für Diagramme in der IBM MQ Console können Überwachungsdaten für Warteschlangenmanager angezeigt werden.

Informationen zu diesem Vorgang

Sie fügen ein Widget für Diagramme zu Ihrem Dashboard hinzu und konfigurieren es dann, um einen bestimmten Aspekt der Ressourcennutzung zu überwachen. Sie können viele Instanzen des Widget für Diagramme erstellen, um unterschiedliche Daten anzuzeigen. Die Daten werden in einem Diagrammformat angezeigt.

Daten werden in 10-Sekunden-Intervallen gesammelt. Die X-Achse des Diagramms zeigt eine Zeitleiste an. Die Y-Achse zeigt Einheiten passend zu der überwachten Ressource an. Die Skala auf der Y-Achse wird dynamisch geändert, um sie an die zurückgegebenen Daten anzupassen.

Es muss mindestens ein Warteschlangenmanager aktiv sein, damit Sie ein Widget für Diagramme konfigurieren können.

 Unter z/OS kann die Systemressourcennutzung nicht überwacht werden.

Vorgehensweise

1. Fügen Sie ein Widget für Diagramme zum Dashboard hinzu:

a) Klicken Sie auf das Symbol 'Widget hinzufügen'

V 9.0.5

Add widget

b) Wählen Sie **Diagramme** aus.

2. Konfigurieren Sie das Widget für Diagramme für die Anzeige von Daten:

a) Klicken Sie auf das Konfigurationssymbol

V 9.0.5



in der Titelleiste des Diagrammwidgets.

b) Optional: Geben Sie einen **Widgettitel** ein. Dieser Titel wird in der Titelleiste des Widgets angezeigt.

c) Wählen Sie die zu überwachende **Ressourcenklasse** aus:

Plattform-CPUs

Überwachen Sie die Nutzung der CPUs.

Plattformpersistente Datenspeicher

Überwachen Sie die Nutzung der Plattenressource.

API-Nutzungsstatistik

Überwachen Sie API-Aufrufe.

API-Nutzungsstatistik pro Warteschlange

Überwachen Sie API-Aufrufe durch einzelne Warteschlangen. Bei Auswahl dieser Klasse geben Sie den Namen der zu überwachenden Warteschlange im Feld **Objekt** ein.

d) Wählen Sie den zu überwachenden **Ressourcentyp** aus.

Welche Ressourcentypen zur Auswahl verfügbar sind, hängt von der ausgewählten Ressourcenklasse ab. In der folgenden Tabelle werden die Ressourcentypen aufgeführt:

Klasse	Typ	Beschreibung
Plattform-CPUs	CPU-Leistung - plattformweit	Wählen Sie diesen Typ aus, um Leistungsdaten für die CPUs und den Hauptspeicher anzuzeigen.
	CPU-Leistung - aktiver Warteschlangenmanager	Wählen Sie diesen Typ aus, um Leistungsdaten für die CPUs und den Hauptspeicher anzuzeigen, die sich auf die Warteschlangenmanager beziehen, die Sie überwachen. Damit ein Warteschlangenmanager überwacht werden kann, muss er aktiv sein. Wenn Sie Ergebnisse von mehreren Warteschlangenmanagern überwachen, werden zur Unterscheidung der Leistungsdaten im Diagramm verschiedene Farben verwendet.
Plattformpersistente Datenspeicher	Plattenbelegung - plattformweit	Wählen Sie diesen Typ aus, um Leistungsdaten für die globale Plattenbelegung anzuzeigen.

<i>Tabelle 6. Ressourcentypen (Forts.)</i>		
Klasse	Typ	Beschreibung
	Plattenbelegung - aktive Warteschlangenmanager	Wählen Sie diesen Typ aus, um Leistungsdaten für die Plattenbelegung anzuzeigen, die sich auf die Warteschlangenmanager bezieht, die Sie überwachen. Damit ein Warteschlangenmanager überwacht werden kann, muss er aktiv sein. Wenn Sie Ergebnisse von mehreren Warteschlangenmanagern überwachen, werden zur Unterscheidung der Leistungsdaten im Diagramm verschiedene Farben verwendet.
	Plattenbelegung - Wiederherstellungsprotokoll für Warteschlangenmanager	Wählen Sie diesen Typ aus, um Daten zur Plattenspeichernutzung für das Wiederherstellungsprotokoll jedes Warteschlangenmanagers, den Sie überwachen, anzuzeigen.
API-Nutzungsstatistik	MQCONN und MQDISC	Wählen Sie diesen Typ aus, um Daten zu MQCONN- und MQDISC-Aufrufen anzuzeigen.
	MQOPEN und MQCLOSE	Wählen Sie diesen Typ aus, um Daten zu MQOPEN- und MQCLOSE-Aufrufen anzuzeigen.
	MQINQ und MQSET	Wählen Sie diesen Typ aus, um Daten zu MQINQ- und MQSET-Aufrufen anzuzeigen.
	MQPUT	Wählen Sie diesen Typ aus, um Daten zu MQPUT-bezogenen Aufrufen anzuzeigen.
	MQGET	Wählen Sie diesen Typ aus, um Daten zu MQGET-bezogenen Aufrufen anzuzeigen.
	Commit und Rollback	Wählen Sie diesen Typ aus, um Informationen zur Verwendung von Synchronisationspunkten durch den Warteschlangenmanager anzuzeigen.
	Abonnieren	Wählen Sie diesen Typ aus, um Daten anzuzeigen, die sich auf MQSUB-Aufrufe beziehen.
	Veröffentlichen	Wählen Sie diesen Typ aus, um Daten zu veröffentlichten Nachrichten anzuzeigen.
API-Nutzungsstatistik pro Warteschlange	MQOPEN und MQCLOSE	Wählen Sie diesen Typ aus, um Daten zu MQOPEN- und MQCLOSE-Aufrufen für die angegebene Warteschlange anzuzeigen.
	MQINQ und MQSET	Wählen Sie diesen Typ aus, um Daten zu MQINQ- und MQSET-Aufrufen für die angegebene Warteschlange anzuzeigen.

Tabelle 6. Ressourcentypen (Forts.)		
Klasse	Typ	Beschreibung
	MQPUT und MQPUT1	Wählen Sie diesen Typ aus, um Daten zu MQPUT-bezogenen und MQPUT1-bezogenen Aufrufen für die angegebene Warteschlange anzuzeigen.
	MQGET	Wählen Sie diesen Typ aus, um Daten zu MQGET-bezogenen Aufrufen für die angegebene Warteschlange anzuzeigen.

e) Wählen Sie das zu überwachende **Ressourcenelement** aus:

Welche Ressourcenelemente zur Auswahl verfügbar sind, hängt von der ausgewählten Ressourcenklasse und vom ausgewählten Ressourcentyp ab. In den folgenden Tabellen werden die Ressourcenelemente aufgeführt:

Tabelle 7. Ressourcenelemente für Plattform-CPUs		
Typ	Element	Beschreibung
CPU-Leistung - plattformweit	Prozentsatz der Benutzer-CPU-Zeit	Zeigt den Prozentsatz der CPU-Auslastung im Benutzerstatus an.
	Prozentsatz der System-CPU-Zeit	Zeigt den Prozentsatz der CPU-Auslastung im Systemstatus an.
	CPU-Belastung – durchschnittlich eine Minute	Zeigt die durchschnittliche Belastung für einen Zeitraum von einer Minute an.
	CPU-Belastung – durchschnittlich fünf Minuten	Zeigt die durchschnittliche Belastung für einen Zeitraum von fünf Minuten an.
	CPU-Belastung – durchschnittlich 15 Minuten	Zeigt die durchschnittliche Belastung für einen Zeitraum von fünfzehn Minuten an.
	Prozentsatz des freien RAM	Zeigt den Prozentsatz des freien RAM-Speichers an.
CPU-Leistung - aktiver Warteschlangenmanager	RAM-Gesamtbytes	Zeigt die Gesamtzahl Bytes des konfigurierten RAM an.
	Benutzer-CPU-Zeit - geschätzter Prozentsatz für Warteschlangenmanager	Schätzt den Prozentsatz der CPU-Nutzung im Benutzerstatus für Prozesse, die sich auf die überwachten Warteschlangenmanager beziehen.
	System-CPU-Zeit - geschätzter Prozentsatz für Warteschlangenmanager	Schätzt den Prozentsatz der CPU-Nutzung im Systemstatus für Prozesse, die sich auf die überwachten Warteschlangenmanager beziehen.
	RAM-Gesamtbytes - Schätzung für Warteschlangenmanager	Schätzt die Gesamtzahl RAM-Bytes, die von den überwachten Warteschlangenmanagern belegt werden.

<i>Tabelle 8. Ressourcenelemente für plattformpersistente Datenspeicher</i>		
Typ	Element	Beschreibung
Plattenbelegung - plattformweit	MQ-Tracedateisystem - belegte Bytes	Zeigt die Anzahl Bytes des Plattenspeichers an, die vom Tracedateisystem belegt werden.
	MQ-Tracedateisystem - freier Speicherplatz	Zeigt den freien Speicherplatz des für das Tracedateisystem reservierten Plattenspeichers an.
	MQ-Fehlerdateisystem - belegte Bytes	Zeigt die Anzahl Bytes des Plattenspeichers an, die von Fehlerdaten belegt werden.
	MQ-Fehlerdateisystem - freier Speicherplatz	Zeigt den freien Speicherplatz des für Fehlerdaten reservierten Plattenspeichers an.
	MQ-FDC-Dateizähler	Zeigt die aktuelle Anzahl von FDC-Dateien an.
Plattenbelegung - aktive Warteschlangenmanager	Dateisystem des Warteschlangenmanagers - belegte Bytes	Zeigt die Anzahl Bytes des Plattenspeichers an, die von Warteschlangenmanagerdateien für die Warteschlangenmanager, die Sie überwachen, belegt werden.
	Dateisystem des Warteschlangenmanagers - freier Speicherplatz	Zeigt den freien Speicherplatz des für Warteschlangenmanagerdateien reservierten Plattenspeichers an.
Plattenbelegung - Wiederherstellungsprotokoll für Warteschlangenmanager	Protokoll - belegte Bytes	Zeigt die Anzahl Bytes des Plattenspeichers an, die für die Wiederherstellungsprotokolle der Warteschlangenmanager, die Sie überwachen, verwendet werden.
	Protokoll - max. Bytes	Zeigt die maximale Anzahl Bytes des Plattenspeichers an, die zur Verwendung für Warteschlangenmanagerwiederherstellungsprotokolle konfiguriert sind.
	Protokolldateisystem - belegte Bytes	Zeigt die Gesamtzahl der für das Protokolldateisystem verwendeten Plattenspeicherbytes an.
	Protokolldateisystem - max. Bytes	Zeigt die Anzahl der für das Protokolldateisystem konfigurierten Plattenspeicherbytes an.
	Protokoll - geschriebene physische Bytes	Zeigt die Anzahl der in die Wiederherstellungsprotokolle geschriebenen Bytes an.

<i>Tabelle 8. Ressourcenelemente für plattformpersistente Datenspeicher (Forts.)</i>		
Typ	Element	Beschreibung
	Protokoll - geschriebene logische Bytes	Zeigt die logische Anzahl der in die Wiederherstellungsprotokolle geschriebenen Bytes an.
	Protokoll - Schreiblatenz	Zeigt beim synchronen Schreiben in das Warteschlangenmanagerwiederherstellungsprotokoll eine Kennzahl für die Latenz an.

<i>Tabelle 9. Ressourcenelemente für API-Nutzungsstatistik</i>		
Typ	Element	Beschreibung
MQCONN und MQDISC	MQCONN/MQCONN-ANZAHL	Zeigt die Anzahl der Aufrufe von MQCONN und MQCONN an.
	Anzahl fehlgeschlagener MQCONN/MQCONN-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQCONN und MQCONN an.
	Gleichzeitige Verbindungen - oberer Grenzwert	Zeigt die maximale Anzahl gleichzeitiger Verbindungen im aktuellen Statistikintervall an.
	MQDISC-Anzahl	Zeigt die Anzahl der Aufrufe von MQDISC an.
MQOPEN und MQCLOSE	MQOPEN-Anzahl	Zeigt die Anzahl der Aufrufe von MQOPEN an.
	Anzahl fehlgeschlagener MQOPEN-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQOPEN an.
	MQCLOSE-Anzahl	Zeigt die Anzahl der Aufrufe von MQCLOSE an.
	Anzahl fehlgeschlagener MQCLOSE-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQCLOSE an.
MQINQ und MQSET	MQINQ-Anzahl	Zeigt die Anzahl der Aufrufe von MQINQ an.
	Anzahl fehlgeschlagener MQINQ-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQINQ an.
	MQSET-Anzahl	Zeigt die Anzahl der Aufrufe von MQSET an.
	Anzahl fehlgeschlagener MQSET-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQSET an.

<i>Tabelle 9. Ressourcenelemente für API-Nutzungsstatistik (Forts.)</i>		
Typ	Element	Beschreibung
MQPUT	MQPUT/MQPUT1-Gesamtanzahl in Intervall	Zeigt die Anzahl der Aufrufe von MQPUT und MQPUT1 an.
	MQPUT/MQPUT1-Byteanzahl in Intervall	Zeigt die Gesamtzahl der Datenbytes an, die von Aufrufen von MQPUT und MQPUT1 eingereicht werden.
	Anzahl nicht persistenter MQPUT-Nachrichten	Zeigt die Anzahl nicht persistenter Nachrichten an, die von MQPUT eingereicht werden.
	Anzahl persistenter MQPUT-Nachrichten	Zeigt die Anzahl persistenter Nachrichten an, die von MQPUT eingereicht werden.
	Anzahl fehlgeschlagener MQPUT-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQPUT an.
	Anzahl nicht persistenter MQPUT1-Nachrichten	Zeigt die Anzahl nicht persistenter Nachrichten an, die von MQPUT1 eingereicht werden.
	Anzahl persistenter MQPUT1-Nachrichten	Zeigt die Anzahl persistenter Nachrichten an, die von MQPUT1 eingereicht werden.
	Anzahl fehlgeschlagener MQPUT1-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQPUT1 an.
	Einreihung nicht persistenter Nachrichten - Byteanzahl	Zeigt die Anzahl der in nicht persistenten Nachrichten eingereichten Bytes an.
	Einreihung persistenter Nachrichten - Byteanzahl	Zeigt die Anzahl der in persistenten Nachrichten eingereichten Bytes an.
	MQSTAT-Anzahl	Zeigt die Anzahl der Aufrufe von MQSTAT an.
	Anzahl fehlgeschlagener MQSTAT-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQSTAT an.
MQGET	Gesamtabrufe mit Löschen in Intervall - Anzahl	Anzahl der Nachrichten, die von MQGET aus Warteschlangen entfernt werden.
	Gesamtabrufe mit Löschen in Intervall - Byteanzahl	Anzahl der Datenbytes, die von MQGET aus Warteschlangen entfernt werden.

<i>Tabelle 9. Ressourcenelemente für API-Nutzungsstatistik (Forts.)</i>		
Typ	Element	Beschreibung
	Abruf nicht persistenter Nachrichten mit Löschen - Anzahl	Anzahl nicht persistenter Nachrichten, die von MQGET aus Warteschlangen entfernt werden.
	Abruf persistenter Nachrichten mit Löschen - Anzahl	Anzahl persistenter Nachrichten, die von MQGET aus Warteschlangen entfernt werden.
	Fehlgeschlagene MQGET-Vorgänge - Anzahl	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQGET an.
	Abruf nicht persistenter Nachrichten - Byteanzahl	Zeigt eine Anzahl Bytes von nicht persistenten Nachrichten an, die an MQGET zurückgegeben werden.
	Abruf persistenter Nachrichten - Byteanzahl	Zeigt eine Anzahl Bytes von persistenten Nachrichten an, die an MQGET zurückgegeben werden.
	Anzeige nicht persistenter Nachrichten - Anzahl	Zeigt eine Anzahl von nicht persistenten Nachrichten an, die angezeigt wurden.
	Anzeige persistenter Nachrichten - Anzahl	Zeigt eine Anzahl von persistenten Nachrichten an, die angezeigt wurden.
	Anzahl fehlgeschlagener Anzeigevorgänge	Zeigt eine Anzahl Nachrichten an, deren Anzeige fehlgeschlagen ist.
	Anzeige nicht persistenter Nachrichten - Byteanzahl	Zeigt die Anzahl Bytes von nicht persistenten Nachrichten an, die angezeigt wurden.
	Anzeige persistenter Nachrichten - Byteanzahl	Zeigt die Anzahl Bytes von persistenten Nachrichten an, die angezeigt wurden.
	Anzahl abgelaufener Nachrichten	Zeigt eine Anzahl der abgelaufenen Nachrichten an.
	Anzahl bereinigter Warteschlangen	Zeigt eine Anzahl von Warteschlangen an, die bereinigt wurden.
	MQCB-Anzahl	Zeigt die Anzahl der Aufrufe von MQCB an.

<i>Tabelle 9. Ressourcenelemente für API-Nutzungsstatistik (Forts.)</i>		
Typ	Element	Beschreibung
	Anzahl fehlgeschlagener MQCBs	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQCB an.
	MQCTL-Anzahl	Zeigt die Anzahl der Aufrufe von MQCTL an.
	Anzahl fehlgeschlagener MQCTL-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQCTL an.
Commit und Rollback	Festschreibungszähler	Zeigt die Anzahl der Aufrufe von MQCMIT an.
	Anzahl fehlgeschlagener Festschreibungen	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQCMIT an.
	Rollback-Anzahl	Zeigt die Anzahl der Aufrufe von MQBACK an.
Abonnieren	Anzahl der Erstellungen permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Erstellen permanenter Subskriptionen an.
	Anzahl der Änderungen permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Ändern permanenter Subskriptionen an.
	Anzahl der Wiederaufnahmen permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Wiederaufnehmen permanenter Subskriptionen an.
	Anzahl der Erstellungen nicht permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Erstellen nicht permanenter Subskriptionen an.
	Anzahl der Änderungen nicht permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Ändern nicht permanenter Subskriptionen an.
	Anzahl der Wiederaufnahmen nicht permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Wiederaufnehmen nicht permanenter Subskriptionen an.
	Anzahl fehlgeschlagener Erstellungen/Änderungen/Wiederaufnahmen von Subskriptionen	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQSUBRQ zum Erstellen, Ändern oder Wiederaufnehmen von Subskriptionen an.

<i>Tabelle 9. Ressourcenelemente für API-Nutzungsstatistik (Forts.)</i>		
Typ	Element	Beschreibung
	Anzahl der Löschungen permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Löschen permanenter Subskriptionen an.
	Anzahl der Löschungen nicht permanenter Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Löschen nicht permanenter Subskriptionen an.
	Anzahl fehlgeschlagener Löschungen von Subskriptionen	Zeigt die Anzahl der Aufrufe von MQSUB zum Löschen von Subskriptionen an.
	MQSUBRQ-Anzahl	Zeigt die Anzahl der Aufrufe von MQSUBRQ an.
	Anzahl fehlgeschlagener MQSUBRQ-Vorgänge	Zeigt die Anzahl der fehlgeschlagenen Aufrufe von MQSUBRQ an.
	Permanenter Subskribent - oberer Grenzwert	Zeigt die maximale Anzahl permanenter Subskriptionen im aktuellen Statistikintervall an.
	Permanenter Subskribent - unterer Grenzwert	Zeigt die minimale Anzahl permanenter Subskriptionen im aktuellen Statistikintervall an.
	Nicht permanenter Subskribent - oberer Grenzwert	Zeigt die maximale Anzahl nicht permanenter Subskriptionen im aktuellen Statistikintervall an.
	Nicht permanenter Subskribent - unterer Grenzwert	Zeigt die minimale Anzahl nicht permanenter Subskriptionen im aktuellen Statistikintervall an.
Veröffentlichen	Gesamtzahl der MQPUT/MQPUT1-Vorgänge für Themen in Intervall	Die Anzahl der Nachrichten, die für Themen eingereicht werden.
	Gesamtbytes eingereichter Themen in Intervall	Die Anzahl der Nachrichtenbytes, die für Themen eingereicht werden.
	Für Subskribenten veröffentlicht - Nachrichtenanzahl	Zeigt die Anzahl der Nachrichten an, die für Subskribenten veröffentlicht werden.
	Für Subskribenten veröffentlicht - Byteanzahl	Zeigt die Byteanzahl von Nachrichten an, die für Subskribenten veröffentlicht werden.

<i>Tabelle 9. Ressourcenelemente für API-Nutzungsstatistik (Forts.)</i>		
Typ	Element	Beschreibung
	Nicht persistent - Anzahl der MQPUT/MQPUT1-Vorgänge für Themen	Zeigt die Anzahl nicht persistenter Nachrichten an, die für Themen eingereicht werden.
	Persistent - Anzahl der MQPUT/MQPUT1-Vorgänge für Themen	Zeigt die Anzahl persistenter Nachrichten an, die für Themen eingereicht werden.
	Anzahl fehlgeschlagener MQPUT/MQPUT1-Vorgänge für Themen	Zeigt die Anzahl fehlgeschlagener Versuche zur Nachrichtenübergabe für ein Thema an.

<i>Tabelle 10. Ressourcenelemente für API-Nutzungsstatistik pro Warteschlange</i>		
Typ	Element	Beschreibung
MQOPEN und MQCLOSE	MQOPEN-Anzahl	Zeigt die Anzahl der Aufrufe von MQOPEN an.
	MQCLOSE-Anzahl	Zeigt die Anzahl der Aufrufe von MQCLOSE an.
MQINQ und MQSET	MQINQ-Anzahl	Zeigt die Anzahl der Aufrufe von MQINQ an.
	MQSET-Anzahl	Zeigt die Anzahl der Aufrufe von MQSET an.
MQPUT und MQPUT1	MQPUT/MQPUT1-Anzahl	Zeigt die Anzahl der Aufrufe von MQPUT und MQPUT1 an.
	MQPUT-Byteanzahl	Zeigt die Gesamtzahl der Datenbytes an, die von Aufrufen von MQPUT und MQPUT1 eingereicht werden.
	MQPUT - Anzahl nicht persistenter Nachrichten	Zeigt die Anzahl nicht persistenter Nachrichten an, die von MQPUT eingereicht werden.
	MQPUT - Anzahl persistenter Nachrichten	Zeigt die Anzahl persistenter Nachrichten an, die von MQPUT eingereicht werden.
	MQPUT1 - Anzahl nicht persistenter Nachrichten	Zeigt die Anzahl nicht persistenter Nachrichten an, die von MQPUT1 eingereicht werden.
	MQPUT1 - Anzahl persistenter Nachrichten	Zeigt die Anzahl persistenter Nachrichten an, die von MQPUT1 eingereicht werden.
	Anzahl nicht persistenter Bytes	Zeigt die Anzahl der in nicht persistenten Nachrichten eingereichten Bytes an.

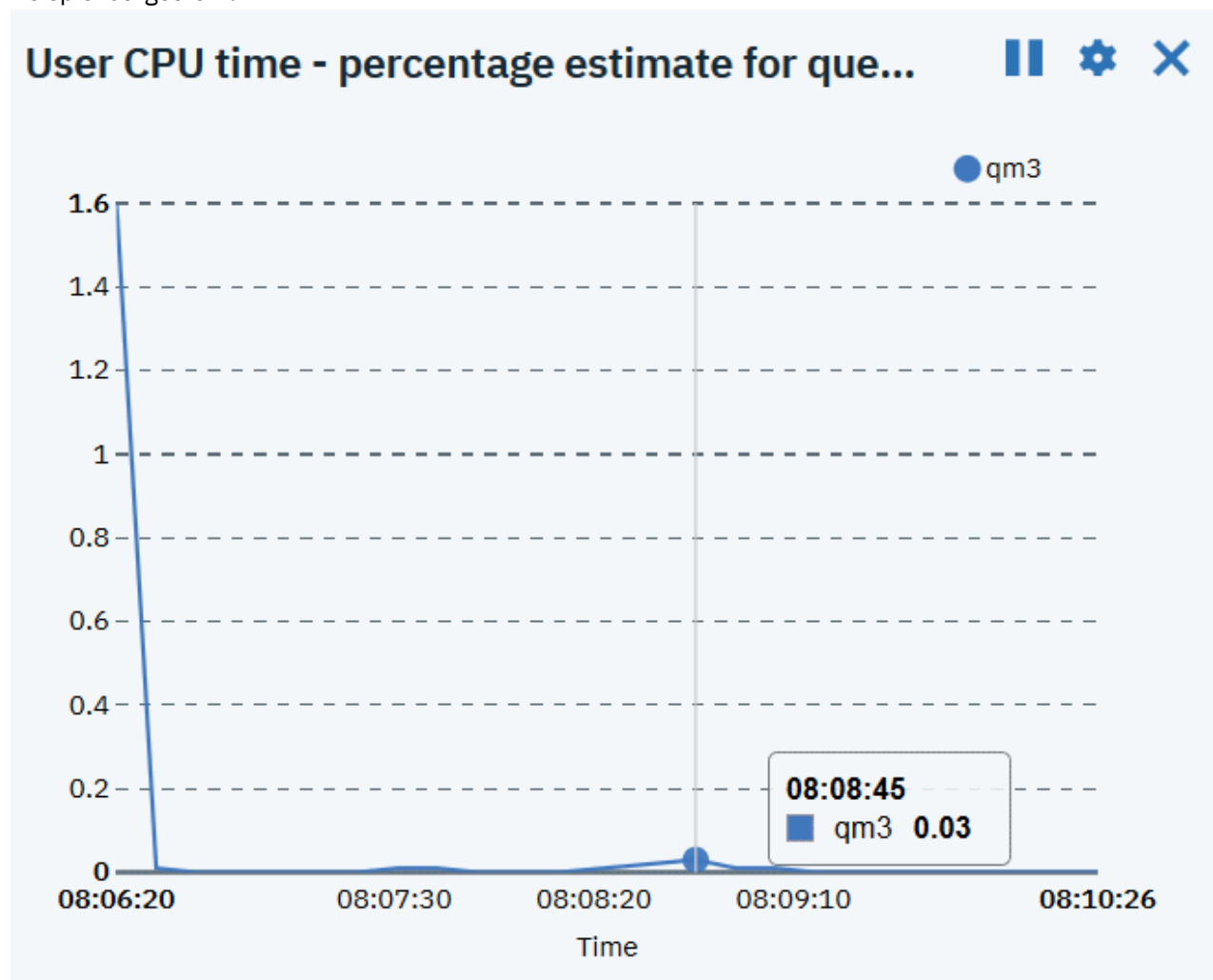
<i>Tabelle 10. Ressourcenelemente für API-Nutzungsstatistik pro Warteschlange (Forts.)</i>		
Typ	Element	Beschreibung
	Anzahl persistenter Bytes	Zeigt die Anzahl der in persistenten Nachrichten eingereiheten Bytes an.
	Vermiedene Einreihungen in Warteschlange	
	Vermiedene Bytes in Warteschlange	
	Sperrkonflikt	
MQGET	MQGET-Anzahl	
	MQGET-Byteanzahl	
	Anzahl zerstörerischer MQGET-Vorgänge für nicht persistente Nachrichten	Anzahl nicht persistenter Nachrichten, die von MQGET aus der Warteschlange entfernt werden.
	Anzahl zerstörerischer MQGET-Vorgänge für persistente Nachrichten	Anzahl persistenter Nachrichten, die von MQGET aus der Warteschlange entfernt werden.
	Anzahl nicht persistenter Bytes bei zerstörerischen MQGET-Vorgängen	Zeigt eine Anzahl Bytes von nicht persistenten Nachrichten an, die an MQGET zurückgegeben werden.
	Anzahl persistenter Bytes bei zerstörerischen MQGET-Vorgängen	Zeigt eine Anzahl Bytes von persistenten Nachrichten an, die an MQGET zurückgegeben werden.
	Anzahl der MQGET-Vorgänge zur Anzeige für nicht persistente Nachrichten	Zeigt eine Anzahl von nicht persistenten Nachrichten an, die angezeigt wurden.
	Anzahl der MQGET-Vorgänge zur Anzeige für persistente Nachrichten	Zeigt eine Anzahl von persistenten Nachrichten an, die angezeigt wurden.
	Byteanzahl der MQGET-Vorgänge zur Anzeige (nicht persistent)	Zeigt die Anzahl Bytes von nicht persistenten Nachrichten an, die angezeigt wurden.
	Byteanzahl der MQGET-Vorgänge zur Anzeige (persistent)	Zeigt die Anzahl Bytes von persistenten Nachrichten an, die angezeigt wurden.
	Abgelaufene Nachrichten	Zeigt eine Anzahl der abgelaufenen Nachrichten an.
	Anzahl bereinigter Warteschlangen	Zeigt eine Anzahl von Warteschlangen an, die bereinigt wurden.

Tabelle 10. Ressourcenelemente für API-Nutzungsstatistik pro Warteschlange (Forts.)		
Typ	Element	Beschreibung
	Durchschnittszeit in Warteschlange	
	Wartezeit	

- f) Wählen Sie einen zu überwachenden Warteschlangenmanager aus und geben Sie die Farbe an, in der Informationen für diesen Warteschlangenmanager angezeigt werden sollen. Klicken Sie auf **Hinzufügen**, um weitere Warteschlangenmanager hinzuzufügen. Sie können bis zu fünf Warteschlangenmanager angeben.
- g) Klicken Sie auf **Speichern**.

Ergebnisse

Nachdem Sie das Widget konfiguriert haben, gibt es eine kurze Verzögerung, bevor Daten im Diagramm angezeigt werden. Daten werden entlang einer Zeitachse angezeigt. Jeder Datenpunkt repräsentiert das Ende des 10-Sekunden-Zeitraums, über den die Daten gesammelt werden. Sie können den Mauszeiger über Datenpunkte im Diagramm bewegen, um detaillierte Informationen anzuzeigen, wie im folgenden Beispiel dargestellt: V 9.0.5




V 9.0.1 Dashboard-Layouts konfigurieren

Ein Dashboard ist ein Container in der IBM MQ Console, in dem Widgets angezeigt werden. Sie können mehrere Dashboardregisterkarten erstellen, um eine unterschiedliche Auswahl von Informationen anzuzeigen.

Informationen zu diesem Vorgang

Sie können jede Dashboardregisterkarte konfigurieren, indem Sie auf den Pfeil neben dem Registerkar-

tennamen klicken . Sie können den Namen der Registerkarte ändern und eine Beschreibung für die Registerkarte hinzufügen. Außerdem können Sie die Anzahl Spalten der Registerkarte konfigurieren.

Sie können das Layout der Widgets innerhalb einer Dashboardregisterkarte konfigurieren, indem Sie die Anordnung der Widgets per Drag-and-drop ändern.

Prozedur

- „Dashboardregisterkarten erstellen und löschen“ auf Seite 133
- „Dashboard-Layouts importieren und exportieren“ auf Seite 134

V 9.0.1 Dashboardregisterkarten erstellen und löschen

Sie können automatisch eine Dashboardregisterkarte erstellen, die Informationen zu einem bestimmten lokalen Warteschlangenmanager anzeigt. Sie können Dashboardregisterkarten manuell erstellen und löschen.

Informationen zu diesem Vorgang

Wenn Sie automatisch eine Dashboardregisterkarte erstellen, die Informationen zu einem bestimmten lokalen Warteschlangenmanager anzeigen soll, werden folgende Widgets automatisch hinzugefügt:

- Warteschlangenwidget
- Widget für Clientverbindungskanäle
- Widget für Kanäle
- Widget für Empfangsprogramme
- Widget für Subskriptionen
- Themenwidget
- Widget für Authentifizierungsinformationen

Prozedur

- So erstellen Sie eine Dashboardregisterkarte:
 - a) Klicken Sie auf das Plusymbol neben Ihren vorhandenen Dashboardregisterkarten



- b) Geben Sie einen Namen für die neue Registerkarte ein.
 - c) Optional: Geben Sie eine Beschreibung für die neue Registerkarte ein.
 - d) Klicken Sie auf **Hinzufügen**.
- So erstellen Sie automatisch eine Dashboardregisterkarte für einen bestimmten Warteschlangenmanager:
 - a) Wählen Sie den Warteschlangenmanager im Widget für lokale Warteschlangenmanager aus.

b) Auswählen ... > **Neue Registerkarte 'Dashboard' hinzufügen**

Es wird eine neue Dashboardregisterkarte erstellt. Die Registerkarte erhält den Namen des Warteschlangenmanagers.

- So löschen Sie eine Dashboardregisterkarte:

a) Klicken Sie neben dem Namen der Dashboardregisterkarte auf den Pfeil  .

b) Wählen Sie **Registerkarte löschen** aus.

c) Bestätigen Sie durch Klicken auf **Löschen**, dass die Dashboardregisterkarte gelöscht werden soll. Die Registerkarte wird gelöscht.

Dashboard-Layouts importieren und exportieren


Sie können ein Dashboardlayout speichern, indem Sie aus der IBM MQ Console exportieren. Sie können ein gespeichertes Dashboardlayout in die IBM MQ Console importieren.

Informationen zu diesem Vorgang

Wenn Sie ein Dashboard exportieren, wird auf der lokalen Festplatte eine JSON-Datei erstellt. Anschließend können Sie die JSON-Datei in ein Dashboard importieren, um das Layout erneut zu erstellen. Beim Importieren eines Dashboardlayouts können Sie angeben, dass die importierten Registerkarten zu einem vorhandenen Dashboardlayout hinzugefügt werden sollen. Alternativ können Sie das vorhandene Dashboardlayout durch das importierte Layout ersetzen.

Prozedur

- So exportieren Sie ein Dashboardlayout:

a) Klicken Sie auf das Dashboardmenüsymbol  .

b) Wählen Sie **Dashboard exportieren** aus.

Die Datei wird im Downloadordner Ihres Browsers gespeichert.

- So importieren Sie ein Dashboardlayout:

a) Klicken Sie auf das Dashboardmenüsymbol  .

b) Wählen Sie **Dashboard importieren** aus.

Das Fenster 'Dashboardkonfiguration importieren' wird geöffnet.

c) Klicken Sie auf **Durchsuchen** und suchen Sie nach der Position der Datei, die Ihre Konfiguration enthält.

d) Wählen Sie aus, wie die Dashboardregisterkarten importiert werden sollen:

Sie können zwischen folgenden Optionen wählen:

- **Registerkarten des importierten Dashboards an vorhandenes Dashboard anfügen**
- **Vorhandenes Dashboard durch Registerkarten des importierten Dashboards ersetzen**

e) Klicken Sie auf **Import** (Importieren).

Die Dashboardregisterkarten werden importiert.





Steuerelemente der Statusübersicht

Über die Steuerelemente oben im Dashboard können Sie die Tracefunktion in der IBM MQ Console aktivieren, auf die Onlinehilfe zugreifen, Informationen zu IBM MQ Console anzeigen und sich von der IBM MQ Console abmelden.

Informationen zu diesem Vorgang

Die Steuerelemente entsprechen allgemeinen Funktionen der IBM MQ Console.

Prozedur

- Greifen Sie auf die IBM Documentation für IBM MQ zu, indem Sie auf das Hilfesymbol klicken .
- Klicken Sie auf das Menüsymbol , um das Dashboard zu importieren, zu exportieren oder zurückzusetzen. Weitere Informationen finden Sie unter „[Dashboard-Layouts konfigurieren](#)“ auf Seite 133.
- Klicken Sie auf das Symbol für Einstellungen , um den Diagnosetrace für die IBM MQ Console zu aktivieren oder zu inaktivieren.
- Sie können Informationen zur IBM MQ Console, einschließlich den Informationen zum angemeldeten Benutzer, anzeigen, indem Sie auf das Personensymbol  klicken und **Informationen** auswählen.
- Klicken Sie auf **Abmelden**, um sich bei der IBM MQ Console abzumelden.
Diese Option ist nicht verfügbar, wenn Sie über ein Clientzertifikat bei IBM MQ Console angemeldet sind oder wenn das Sicherheitskonfigurationsbeispiel `no_security.xml` verwendet wird. Weitere Informationen zu den Sicherheitskonfigurationen finden Sie im Abschnitt [Sicherheit der IBM MQ Console konfigurieren](#).

Tastaturkurzbefehle

Sie können in der IBM MQ Console mit Direktaufrufen arbeiten.

In der folgenden Tabelle sind die verfügbaren Direktaufrufe aufgeführt. Sie werden alle in einem Widget verwendet.

Tabelle 11. In der Konsole verfügbare Direktaufrufe

Schlüssel	Action
p	Eigenschaften des ausgewählten Objekts anzeigen
c	Neues Objekt erstellen
Umschalttaste+d	Ausgewähltes Objekt löschen
a	Alle Objekts in einem Widget auswählen
Umschalttaste+a	Auswahl von Objekten aufheben

Windows

Linux

Verwaltung über die IBM MQ Explorer

Mit dem IBM MQ Explorer können Sie die lokale oder ferne Verwaltung Ihres Netzes von einem Computer aus ausführen, auf dem nur Windows oder Linux x86-64 ausgeführt wird.

IBM MQ for Windows und IBM MQ for Linux x86-64 stellen eine Verwaltungsschnittstelle mit dem Namen IBM MQ Explorer bereit, um Verwaltungstasks als Alternative zur Verwendung von Steuer- oder MQSC-Befehlen auszuführen. [Befehlssätze vergleichen](#) zeigt Ihnen, was Sie mit IBM MQ Explorer tun können.

Der IBM MQ Explorer ermöglicht Ihnen die lokale oder ferne Verwaltung Ihres Netzes von einem Computer, auf dem Windows oder Linux x86-64 ausgeführt wird, indem Sie auf den IBM MQ Explorer auf den Warteschlangenmanagern und Clustern zeigen, die Sie interessieren. Es ist eine Fernverbindung zu Warteschlangenmanagern möglich, die auf einer beliebigen unterstützten Plattform, einschließlich z/OS, ausgeführt werden. Auf diese Weise kann Ihr gesamter Messaging-Backbone über die Konsole angezeigt, durchsucht und geändert werden.

Informationen, wie ferne IBM MQ-Warteschlangenmanager konfiguriert werden müssen, damit sie mit IBM MQ Explorer verwaltet werden können, finden Sie im Abschnitt „Softwarevoraussetzungen und Definitionen für IBM MQ Explorer“ auf Seite 138.

Es ermöglicht Ihnen die Ausführung von Tasks, die in der Regel mit der Einrichtung und Optimierung der Arbeitsumgebung für IBM MQ verbunden sind, entweder lokal oder über Fernzugriff in einer Windows- oder Linux x86-64-Systemdomäne.

Unter Linux kann IBM MQ Explorer möglicherweise nicht gestartet werden, wenn Sie über mehrere Eclipse-Installationen verfügen. In diesem Fall müssen Sie IBM MQ Explorer mit einer anderen Benutzer-ID als der starten, die für die andere Eclipse-Installation verwendet wird.

Damit IBM MQ Explorer unter Linux gestartet werden kann, muss es möglich sein, eine Datei in Ihr Ausgangsverzeichnis zu schreiben; außerdem muss das Ausgangsverzeichnis vorhanden sein.

Windows

Linux

Was Sie mit dem IBM MQ Explorer tun können

Sie können mit dem IBM MQ Explorer Verwaltungstasks unter Verwendung einer Reihe von Inhaltsansichten und Eigenschaftendialogen ausführen. Sie können den IBM MQ Explorer auch erweitern, indem Sie eine oder mehrere Eclipse-Plug-ins schreiben.

IBM MQ Explorer-Tasks

Mit dem IBM MQ Explorer können Sie folgende Tasks ausführen:

- Erstellen und löschen Sie einen WS-Manager (nur auf Ihrer lokalen Maschine).
- Starten und stoppen Sie einen WS-Manager (nur auf Ihrer lokalen Maschine).
- Definieren, Anzeigen und Ändern der Definitionen von IBM MQ-Objekten, wie z. B. Warteschlangen und Kanäle.
- Durchsuchen Sie die Nachrichten in einer Warteschlange.
- Starten und stoppen Sie einen Kanal.
- Statusinformationen zu einem Kanal, einem Listener, einer Warteschlange oder Serviceobjekten anzeigen.
- Anzeigen von Warteschlangenmanagern in einem Cluster.
- Überprüfen Sie, welche Anwendungen, Benutzer oder Kanäle eine bestimmte Warteschlange geöffnet haben.
- Erstellen Sie mit dem Assistenten 'Neuen Cluster erstellen' einen neuen WS-Manager-Cluster.
- Fügen Sie einen WS-Manager mit Hilfe des Assistenten zum Hinzufügen von WS-Manager zu Cluster einem Cluster hinzu.
- Verwalten Sie das Authentifizierungsinformationsobjekt, das mit der TLS-Kanalsicherheit (Transport Layer Security) verwendet wird.
- Kanalinitiatoren, Auslösemonitore und Empfangsprogramme erstellen und löschen.
- Starten oder stoppen Sie die Befehlsserver, Kanalinitiatoren, Auslösemonitore und Empfangsprogramme.
- Legen Sie bestimmte Services fest, die beim Start eines Warteschlangenmanagers automatisch gestartet werden sollen.
- Ändern der Merkmale von Warteschlangenmanagern.
- Ändern Sie den lokalen Standardwarteschlangenmanager.
- Rufen Sie die **stzmqikm** -GUI (ikeyMan) auf, um TLS-Zertifikate zu verwalten, Zertifikate Warteschlangenmanagern zuzuordnen und Zertifikatsspeicher zu konfigurieren und einzurichten (nur auf Ihrer lokalen Maschine).
- Erstellen Sie JMS-Objekte aus IBM MQ-Objekten und IBM MQ-Objekte aus JMS-Objekten.
- Erstellen Sie eine JMS-Verbindungsfactory für jeden der derzeit unterstützten Typen.

- Ändern Sie die Parameter für jeden Service, wie z. B. die TCP-Portnummer für einen Listener oder einen Namen für die Kanalinitiatorwarteschlange.
- Den Service-Trace starten oder stoppen.

Inhaltsansichten und Eigenschaftendialoge

Sie führen Verwaltungstasks mit einer Reihe von Inhaltsansichten und Eigenschaftendialogen aus.

Inhaltsansicht

Eine Inhaltsansicht ist eine Anzeige, in der Folgendes angezeigt werden kann:

- Attribute und Verwaltungsoptionen, die sich auf IBM MQ selbst beziehen.
- Attribute und Verwaltungsoptionen, die sich auf ein oder mehrere zugehörige Objekte beziehen.
- Attribute und Verwaltungsoptionen für einen Cluster.

Eigenschaftendialoge

Ein Eigenschaftendialog ist eine Anzeige, in der Attribute angezeigt werden, die sich auf ein Objekt in einer Reihe von Feldern beziehen, von denen einige bearbeitet werden können.

Sie navigieren mithilfe der Navigatorsicht durch den IBM MQ Explorer. Im Navigator können Sie die von Ihnen benötigten Inhaltsansicht auswählen.

IBM MQ Explorer erweitern

Der IBM MQ Explorer stellt Informationen in einem Stil dar, der mit dem des Eclipse-Frameworks und den anderen Plug-in-Anwendungen, die von Eclipse unterstützt werden, konsistent ist.

Durch eine Erweiterung von IBM MQ Explorer haben Systemadministratoren die Möglichkeit, den IBM MQ Explorer anzupassen, um die Art und Weise, wie sie IBM MQ verwalten, zu verbessern.

Weitere Informationen hierzu finden Sie im Abschnitt [MQ Explorer erweitern](#).

Entscheiden, ob der IBM MQ Explorer verwendet werden soll

Beachten Sie bei der Entscheidung, ob der IBM MQ Explorer bei Ihrer Installation verwendet werden soll, die in diesem Abschnitt aufgeführten Informationen.

Sie müssen die folgenden Punkte beachten:

Objektnamen

Wenn Sie Namen von Warteschlangenmanagern und anderen Objekten mit IBM MQ Explorer in Kleinbuchstaben verwenden, müssen Sie die Objektnamen bei Verwendung von MQSC-Befehlen in einfache Anführungszeichen setzen, oder IBM MQ erkennt sie nicht.

Große Warteschlangenmanager

Der IBM MQ Explorer funktioniert am besten mit kleinen Warteschlangenmanagern. Wenn Sie über eine große Anzahl von Objekten auf einem einzigen Warteschlangenmanager verfügen, können Verzögerungen auftreten, während der IBM MQ Explorer die erforderlichen Informationen zur Darstellung in einer Ansicht extrahiert.

Cluster

IBM MQ-Cluster können potenziell Hunderte oder Tausende von Warteschlangenmanagern enthalten. Der IBM MQ Explorer stellt die Warteschlangenmanager in einem Cluster in einer Baumstruktur dar. Die physische Größe eines Clusters hat keine Auswirkungen auf die Geschwindigkeit des IBM MQ Explorers, da der IBM MQ Explorer erst dann eine Verbindung zu den Warteschlangenmanagern im Cluster herstellt, wenn Sie sie auswählen.

IBM MQ Explorer einrichten

In diesem Abschnitt werden die erforderlichen Schritte zum Einrichten des IBM MQ Explorers beschrieben.

- [„Softwarevoraussetzungen und Definitionen für IBM MQ Explorer“ auf Seite 138](#)
- [„Sicherheit für IBM MQ Explorer“ auf Seite 138](#)
- [„Warteschlangenmanager und Cluster in IBM MQ Explorer ein- und ausblenden“ auf Seite 142](#)
- [„Clusterzugehörigkeit und IBM MQ Explorer“ auf Seite 143](#)
- [„Datenkonvertierung für IBM MQ Explorer“ auf Seite 144](#)

Softwarevoraussetzungen und Definitionen für IBM MQ Explorer

Stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind, bevor Sie versuchen, den IBM MQ Explorer zu verwenden.

Der IBM MQ Explorer kann nur über das TCP/IP-Kommunikationsprotokoll eine Verbindung zu fernen Warteschlangenmanagern herstellen.

Überprüfen Sie Folgendes:

1. Ein Befehlsserver wird auf jedem über Remotezugriff verwalteten Warteschlangenmanager ausgeführt.
2. Ein geeignetes TCP/IP-Listener-Objekt muss auf jedem fernen WS-Manager ausgeführt werden. Dieses Objekt kann der IBM MQ-Listener oder, auf Systemen unter UNIX and Linux, der Dämon Inetd sein.
3. Ein Serververbindungskanal, der standardmäßig mit dem Namen SYSTEM.ADMIN.SVRCONN bezeichnet wird, ist auf allen fernen Warteschlangenmanagern vorhanden.

Sie können den Kanal mit dem folgenden MQSC-Befehl erstellen:

```
DEFINE CHANNEL (SYSTEM.ADMIN.SVRCONN) CHLTYPE (SVRCONN)
```

Mit diesem Befehl wird eine Basiskanaldefinition erstellt. Wenn Sie eine komplexere Definition wünschen (z. B. die Sicherheit konfigurieren), benötigen Sie zusätzliche Parameter. Weitere Informationen finden Sie in `DEFINE CHANNEL`.

4. Die Systemwarteschlange (SYSTEM.MQEXPLORER.REPLY.MODEL) muss vorhanden sein.

Sicherheit für IBM MQ Explorer

Wenn Sie IBM MQ in einer Umgebung verwenden, in der es für Sie wichtig ist, den Benutzerzugriff auf bestimmte Objekte zu steuern, sollten Sie gegebenenfalls die Sicherheitsaspekte bei der Verwendung von IBM MQ Explorer berücksichtigen.

Berechtigung zur Verwendung von IBM MQ Explorer

Jeder Benutzer kann den IBM MQ Explorer verwenden, allerdings sind bestimmte Berechtigungen für die Verbindung, den Zugriff und die Verwaltung von Warteschlangenmanagern erforderlich.

Um lokale Verwaltungstasks mit dem IBM MQ Explorer ausführen zu können, muss ein Benutzer über die erforderliche Berechtigung zum Ausführen der Verwaltungstasks verfügen. Wenn der Benutzer Mitglied der Gruppe mqm ist, hat der Benutzer die Berechtigung, alle lokalen Verwaltungstasks auszuführen.

Um eine Verbindung zu einem fernen Warteschlangenmanager herstellen und ferne Verwaltungstasks mit dem IBM MQ Explorer ausführen zu können, muss der Benutzer, der den IBM MQ Explorer ausführt, folgende Berechtigungen besitzen:

- CONNECT-Berechtigung für das Ziel-WS-Manager-Objekt
- Berechtigung INQUIRE für das Ziel-WS-Manager-Objekt
- Berechtigung DISPLAY für das Ziel-WS-Manager-Objekt
- Berechtigung INQUIRE für die Warteschlange, SYSTEM.MQEXPLORER.REPLY.MODEL
- Berechtigung DISPLAY für die Warteschlange SYSTEM.MQEXPLORER.REPLY.MODEL
- Berechtigung INPUT (get) für die Warteschlange, SYSTEM.MQEXPLORER.REPLY.MODEL
- Berechtigung OUTPUT (put) für die Warteschlange, SYSTEM.MQEXPLORER.REPLY.MODEL
- Berechtigung OUTPUT (put) für die Warteschlange, SYSTEM.ADMIN.COMMAND.QUEUE

- Berechtigung INQUIRE in der Warteschlange, SYSTEM.ADMIN.COMMAND.QUEUE
- Berechtigung zum Ausführen der ausgewählten Aktion

Anmerkung: Die INPUT-Berechtigung bezieht sich auf die Eingabe für den Benutzer aus einer Warteschlange (eine get-Operation). Die OUTPUT-Berechtigung bezieht sich auf die Ausgabe des Benutzers in eine Warteschlange (eine PUT-Operation).

Wenn Sie eine Verbindung zu einem fernen Warteschlangenmanager unter IBM MQ for z/OS herstellen und über den IBM MQ Explorer ferne Verwaltungstasks ausführen möchten, müssen Sie Folgendes bereitstellen:

- RACF-Profil für die Systemwarteschlange SYSTEM.MQEXPLORER.REPLY.MODEL
- RACF-Profil für die Warteschlangen AMQ.MQEXPLORER.*

Darüber hinaus muss der Benutzer, der den IBM MQ Explorer ausführt, über folgende Berechtigungen verfügen:

- RACF-Berechtigung UPDATE für die Systemwarteschlange SYSTEM.MQEXPLORER.REPLY.MODEL
- RACF-Berechtigung UPDATE für die Warteschlangen AMQ.MQEXPLORER.*
- CONNECT-Berechtigung für das Ziel-WS-Manager-Objekt
- Berechtigung zum Ausführen der ausgewählten Aktion
- Berechtigung READ für alle hlq.DISPLAY.Objekt-Profile in der Klasse MQCMD5

Informationen zum Erteilen von Berechtigungen für IBM MQ-Objekte finden Sie im Abschnitt [Zugriff auf ein IBM MQ-Objekt auf UNIX oder Linux-Systemen](#) und unter [Windows](#) erteilen.

Wenn ein Benutzer versucht, eine Operation auszuführen, die er nicht ausführen darf, ruft der Zielwarteschlangenmanager Berechtigungsfehler auf, und die Operation schlägt fehl.

Gemäß dem Standardfilter im IBM MQ Explorer werden alle IBM MQ-Objekte angezeigt. Wenn es IBM MQ-Objekte gibt, für die ein Benutzer nicht die Berechtigung DISPLAY besitzt, werden Berechtigungsfehler generiert. Wenn Berechtigungsereignisse aufgezeichnet werden, schränken Sie den Bereich der Objekte ein, die für die Objekte angezeigt werden, für die der Benutzer die Berechtigung DISPLAY für hat.

Sicherheit für die Verbindung von IBM MQ Explorer zu fernen Warteschlangenmanagern

Sie müssen den Kanal zwischen dem IBM MQ Explorer und jedem fernen Warteschlangenmanager sichern.

Der IBM MQ Explorer stellt als MQI-Clientanwendung eine Verbindung zu fernen Warteschlangenmanagern her. Dies bedeutet, dass jeder ferne WS-Manager über eine Definition eines Serververbindungskanals und eines geeigneten TCP/IP-Listeners verfügen muss. Wenn Sie Ihren Serververbindungskanal nicht sichern, ist es möglich, dass eine heimtückische Anwendung eine Verbindung zum selben Serververbindungskanal herstellt und Zugriff auf die WS-Manager-Objekte mit unbegrenzter Berechtigung erhält. Um Ihren Serververbindungskanal zu sichern, geben Sie für das MCAUSER-Attribut des Kanals einen Wert an, der nicht leer ist, verwenden Sie Kanalauthentifizierungsdatensätze oder verwenden Sie einen Sicherheitsexit.

Der Standardwert für das Attribut MCAUSER ist die lokale Benutzer-ID . Wenn Sie einen nicht leeren Benutzernamen als MCAUSER-Attribut des Serververbindungskanals angeben, werden alle Programme, die mit diesem Kanal eine Verbindung zum WS-Manager herstellen, mit der Identität des benannten Benutzers ausgeführt und haben dieselbe Berechtigungsstufe. Dies ist nicht der Fall, wenn Sie Kanalauthentifizierungsdatensätze verwenden.

Sicherheitsexit mit IBM MQ Explorer verwenden

Sie können bei Verwendung von IBM MQ Explorer einen Standardsicherheitsexit sowie für Warteschlangenmanager spezifische Sicherheitsexits angeben.

Sie können einen Standardsicherheitsexit definieren, der für alle neuen, vom IBM MQ Explorer ausgehenden Clientverbindungen verwendet werden kann. Dieser Standardexit kann zum Zeitpunkt der Verbindungsabschlussverbindung überschrieben werden. Sie können auch einen Sicherheitsexit für einen ein-

zelen Warteschlangenmanager oder eine Gruppe von Warteschlangenmanagern definieren, die wirksam werden, wenn eine Verbindung hergestellt wird. Sie geben die Exits mit dem IBM MQ Explorer an. Weitere Informationen finden Sie in der IBM MQ Explorer-Hilfe.

Mit IBM MQ Explorer über TLS-fähige MQI-Kanäle eine Verbindung zu einem fernen Warteschlangenmanager herstellen

Der IBM MQ Explorer stellt über einen MQI-Kanal Verbindungen zu fernen Warteschlangenmanagern her. Wenn Sie den MQI-Kanal mit der TLS-Sicherheit sichern wollen, müssen Sie den Kanal mit Hilfe einer Definitionstabelle für Clientkanäle einrichten.

Informationen zum Einrichten eines MQI-Kanals unter Verwendung einer Definitionstabelle für Clientkanäle finden Sie unter [Übersicht über IBM MQ MQI clients](#).

Wenn Sie den Kanal über eine Definitionstabelle für Clientkanäle eingerichtet haben, können Sie den IBM MQ Explorer verwenden, um eine Verbindung zu einem fernen Warteschlangenmanager über TLS-fähigen MQI-Kanal herzustellen, wie in „Tasks auf dem System, auf dem sich der ferne WS-Manager befindet“ auf Seite 140 und „Tasks auf dem System, das den IBM MQ Explorer hostet“ auf Seite 140 beschrieben.

Tasks auf dem System, auf dem sich der ferne WS-Manager befindet

Führen Sie auf dem System, auf dem sich der ferne WS-Manager befindet, die folgenden Tasks aus:

1. Definieren Sie eine Serververbindung und ein Clientverbindungspaar und geben Sie den entsprechenden Wert für das Attribut *SSLCIPH* in der Serververbindung auf beiden Kanälen an. Weitere Informationen zum Attribut *SSLCIPH* finden Sie im Abschnitt [Kanäle mit TLS schützen](#).
2. Senden Sie die Kanaldefinitionstabelle *AMQCLCHL.TAB*, die sich im Verzeichnis *@ipcc* des Warteschlangenmanagers befindet, an das System, auf dem sich die IBM MQ Explorer befindet.
3. Starten Sie einen TCP/IP-Listener an einem angegebenen Port.
4. Stellen Sie sowohl die CA-als auch die persönlichen TLS-Zertifikate in das Verzeichnis *SSL* des Warteschlangenmanagers ein:
 - */var/mqm/qmgrs/+QMNAME+/SSL* für Systeme unter UNIX and Linux
 - *C:\Programme\IBM\MQ\qmgrs\+QMNAME+\SSL* für Systeme unter WindowsDabei ist *+QMNAME+* ein Token, das den Namen des Warteschlangenmanagers darstellt.
5. Erstellen Sie eine Schlüsseldatenbankdatei vom Typ *CMS* mit dem Namen *key.kdb*. Speichern Sie das Kennwort verdeckt in einer Datei, indem Sie die Option in der grafischen Benutzerschnittstelle in der Datei **strmqikm** (iKeyman) überprüfen oder indem Sie die Option *-stash* mit den Befehlen **runmqckm** verwenden.
6. Fügen Sie die CA-Zertifikate der Schlüsseldatenbank hinzu, die im vorherigen Schritt erstellt wurde.
7. Importieren Sie das persönliche Zertifikat für den WS-Manager in die Schlüsseldatenbank.

Weitere ausführliche Informationen zum Arbeiten mit TLS auf Windows-Systemen finden Sie unter [Mit TLS auf UNIX, Linux, and Windows arbeiten](#).

Tasks auf dem System, das den IBM MQ Explorer hostet

Führen Sie auf dem System, das den IBM MQ Explorer hostet, folgende Tasks aus:

1. Erstellen Sie eine Schlüsseldatenbankdatei vom Typ *JKS* mit dem Namen *key.jks*. Legen Sie ein Kennwort für diese Schlüsseldatenbankdatei fest.

Der IBM MQ Explorer verwendet Java-Keystore-Dateien (*JKS*) für die TLS-Sicherheit. Daher muss die Keystore-Datei, die für die Konfiguration von TLS für den IBM MQ Explorer erstellt wird, damit übereinstimmen.

2. Fügen Sie die CA-Zertifikate der Schlüsseldatenbank hinzu, die im vorherigen Schritt erstellt wurde.
3. Importieren Sie das persönliche Zertifikat für den WS-Manager in die Schlüsseldatenbank.

4. Auf Windows- und Linux-Systemen können Sie IBM MQ Explorer starten, indem Sie das Systemmenü, die ausführbare Datei von MQExplorer oder den Befehl **strmqcfcg** verwenden.
5. Klicken Sie in der Symbolleiste von IBM MQ Explorer auf **Fenster -> Einstellungen**, erweitern Sie **IBM MQ Explorer** und klicken Sie auf **SSL Client Certificate Stores** (SSL-Clientzertifikatsspeicher). Geben Sie den Namen und das Kennwort für die in Schritt 1 von „Tasks auf dem System, das den IBM MQ Explorer hostet“ auf Seite 140 erstellte JKS-Datei sowohl im vertrauenswürdigen Zertifikatsspeicher als auch im persönlichen Zertifikatsspeicher ein und klicken Sie anschließend auf **OK**.
6. Schließen Sie das Fenster **Benutzervorgaben**, und klicken Sie auf **Warteschlangenmanager**. Klicken Sie auf **Warteschlangenmanager anzeigen/verdecken** und anschließend in der Anzeige **Warteschlangenmanager anzeigen/verdecken** auf **Hinzufügen**.
7. Geben Sie den Namen des Warteschlangenmanagers ein, und wählen Sie die Option **Direkt verbinden** aus. Klicken Sie auf 'Weiter
8. Wählen Sie die Option **Clientkanaldefinitionstabelle (CCDT) verwenden** aus und geben Sie die Position der Kanaltabellendatei an, die Sie vom fernen Warteschlangenmanager in Schritt 2 in „Tasks auf dem System, auf dem sich der ferne WS-Manager befindet“ auf Seite 140 auf dem System, auf dem sich der ferne WS-Manager befindet, übertragen haben.
9. Klicken Sie auf **Fertig stellen**. Sie können jetzt aus dem IBM MQ Explorer auf den fernen Warteschlangenmanager zugreifen.

Mit IBM MQ Explorer eine Verbindung über einen anderen Warteschlangenmanager herstellen

Mit dem IBM MQ Explorer können Sie über einen temporären Warteschlangenmanager, mit dem der IBM MQ Explorer bereits verbunden ist, eine Verbindung zu einem Warteschlangenmanager herstellen.

In diesem Fall übergibt der IBM MQ Explorer PCF-Befehlsnachrichten an den temporären Warteschlangenmanager, wobei er Folgendes angibt:

- Den Parameter *ObjectQMgrName* im Objektdeskriptor (MQOD) als Name des Zielwarteschlangenmanagers. Weitere Informationen zur Auflösung von Warteschlangennamen finden Sie in der [Namensauflösung](#).
- Den Parameter *UserIdentifier* im Nachrichtendeskriptor (MQMD) als lokale Benutzer-ID (userId).

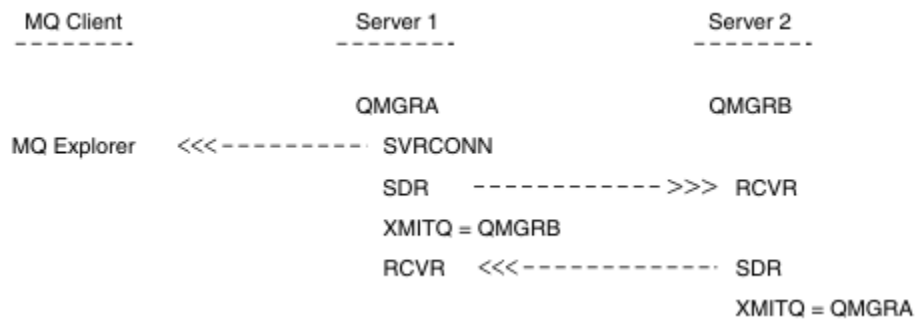
Wenn die Verbindung dann zum Herstellen der Verbindung zum Zielwarteschlangenmanager über einen temporären Warteschlangenmanager verwendet wird, wird die Benutzer-ID erneut im Parameter *UserIdentifier* des Nachrichtendeskriptors (MQMD) ausgegeben. Damit der MCA-Listener auf dem Zielwarteschlangenmanager diese Nachricht akzeptieren kann, muss entweder das Attribut MCAUSER festgelegt werden, oder die Benutzer-ID muss bereits mit der Berechtigung 'put' vorhanden sein.

Der Befehlsserver auf dem Zielwarteschlangenmanager reiht Nachrichten in die Übertragungswarteschlange ein, die die userId im Parameter *UserIdentifier* im Nachrichtendeskriptor (MQMD) angibt. Damit dies erfolgreich ist, muss die userId bereits mit der Berechtigung 'put' auf dem Zielwarteschlangenmanager vorhanden sein.

Im folgenden Beispiel wird gezeigt, wie ein Warteschlangenmanager über einen temporären Warteschlangenmanager mit dem IBM MQ Explorer verbunden wird.

Stellen Sie eine Fernverwaltungsverbindung zu einem Warteschlangenmanager her. Überprüfen Sie Folgendes:

- Der WS-Manager auf dem Server ist aktiv und hat einen Serververbindungskanal (SVRCONN) definiert.
- Listener ist aktiv.
- Der Befehlsserver ist aktiv.
- Die Warteschlange SYSTEM.MQ EXPLORER.REPLY.MODEL wurde erstellt, und Sie verfügen über ausreichende Berechtigungen.
- WS-Manager-Listener, Befehlsserver und Senderkanäle werden gestartet.



In diesem Beispiel gilt Folgendes:

- IBM MQ Explorer ist über eine Clientverbindung mit dem Warteschlangenmanager QMGRA (auf Server1) verbunden.
- Warteschlangenmanager QMGRB auf Server2 kann jetzt über einen temporären Warteschlangenmanager (QMGRA) mit IBM MQ Explorer verbunden werden
- Wenn Sie eine Verbindung zu QMGRB mit IBM MQ Explorer herstellen, wählen Sie QMGRA als temporären Warteschlangenmanager aus.

In dieser Situation besteht keine direkte Verbindung zu QMGRB von IBM MQ Explorer. Die Verbindung zu QMGRB erfolgt über QMGRA.

Der WS-Manager QMGRB auf Server2 ist mit QMGRA auf Server1 mit Sender-Empfänger-Kanälen verbunden. Der Kanal zwischen QMGRA und QMGRB muss so konfiguriert werden, dass die Fernverwaltung möglich ist (siehe [„Kanäle und Übertragungswarteschlangen für die Fernverwaltung vorbereiten“](#) auf Seite 203).

Warteschlangenmanager und Cluster in IBM MQ Explorer ein- und ausblenden

Im IBM MQ Explorer können mehrere Warteschlangenmanager gleichzeitig angezeigt werden. In der Anzeige 'Warteschlangenmanager ein-/ausblenden' können Sie auswählen, ob Sie Informationen zu einer anderen (fernen) Maschine anzeigen können (im Menü können Sie die Option für den Warteschlangenmanagerknoten auswählen). Lokale WS-Manager werden automatisch erkannt.

So zeigen Sie einen fernen Warteschlangenmanager an:

1. Klicken Sie mit der rechten Maustaste auf den Baumknoten **Queue Managers** (Warteschlangenmanager) und wählen Sie anschließend **Show/Hide Queue Managers** (Warteschlangenmanager anzeigen/verdecken) aus.
2. Klicken Sie auf **Hinzufügen**. Die Anzeige 'WS-Manager anzeigen/verdecken' wird angezeigt.
3. Geben Sie den Namen des fernen Warteschlangenmanagers und den Hostnamen oder die IP-Adresse in die angegebenen Felder ein.

Der Hostname oder die IP-Adresse wird verwendet, um eine Clientverbindung zum fernen Warteschlangenmanager mit dem Standard-Serververbindungskanal SYSTEM.ADMIN.SVRCONN oder einem benutzerdefinierten Serververbindungskanal herzustellen.

4. Klicken Sie auf **Fertigstellen**.

In der Anzeige 'WS-Manager anzeigen/verdecken' wird auch eine Liste aller sichtbaren Warteschlangenmanager angezeigt. Sie können diese Anzeige verwenden, um Warteschlangenmanager in der Navigationsansicht zu verdecken.

Wenn im IBM MQ Explorer ein Warteschlangenmanager angezeigt wird, der Mitglied eines Clusters ist, wird der Cluster erkannt und automatisch angezeigt.

Gehen Sie wie folgt vor, um die Liste der fernen WS-Manager aus dieser Anzeige zu exportieren:

1. Schließen Sie die Anzeige 'WS-Manager anzeigen/verdecken'.

2. Klicken Sie mit der rechten Maustaste auf den höchsten **IBM MQ**-Baumknoten im Navigationsfenster des IBM MQ Explorer und wählen Sie anschließend **Export IBM MQ Explorer Settings** (IBM MQ-Einstellungen exportieren) aus.
3. Klicken Sie auf **IBM MQ Explorer > IBM MQ Explorer Settings**.
4. Wählen Sie **Verbindungsdaten > Ferne Warteschlangenmanager** aus.
5. Wählen Sie eine Datei aus, in der die exportierten Einstellungen gespeichert werden sollen.
6. Klicken Sie abschließend auf **Fertig stellen** , um die Verbindungsinformationen des fernen Warteschlangenmanagers in die angegebene Datei zu exportieren.

Gehen Sie wie folgt vor, um eine Liste der fernen Warteschlangenmanager zu importieren

1. Klicken Sie mit der rechten Maustaste auf den höchsten **IBM MQ**-Baumknoten im Navigationsfenster des IBM MQ Explorer und wählen Sie anschließend **Import IBM MQ Explorer Settings** (IBM MQ-Einstellungen importieren) aus.
2. Klicken Sie auf **IBM MQ Explorer > IBM MQ Explorer Settings**.
3. Klicken Sie auf **Durchsuchen** und navigieren Sie zu dem Pfad der Datei, die die Verbindungsinformationen des fernen Warteschlangenmanagers enthält.
4. Klicken Sie auf **Öffnen** . Wenn die Datei eine Liste mit fernen Warteschlangenmanagern enthält, ist das Kästchen **Verbindungsdaten > Ferne Warteschlangenmanager** aktiviert.
5. Klicken Sie abschließend auf **Fertigstellen**, um die Verbindungsinformationen des fernen Warteschlangenmanagers in den IBM MQ Explorer zu importieren.

Clusterzugehörigkeit und IBM MQ Explorer

IBM MQ Explorer benötigt Informationen zu Warteschlangenmanagern, die Mitglied eines Clusters sind.

Wenn ein Warteschlangenmanager Mitglied eines Clusters ist, wird der Clusterknoten automatisch gefüllt.

Wenn Warteschlangenmanager zu Mitgliedern von Clustern werden, während der IBM MQ Explorer ausgeführt wird, müssen Sie den IBM MQ Explorer mit aktuellen Verwaltungsdaten zu Clustern versorgen, damit er effizient mit ihnen kommunizieren und bei einer entsprechenden Anforderung die richtigen Clusterinformationen anzeigen kann. Damit dies möglich ist, benötigt der IBM MQ Explorer folgende Informationen:

- Name eines Repository-Warteschlangenmanagers
- Verbindungsname des Repository-Warteschlangenmanagers, wenn er sich auf einem fernen Warteschlangenmanager befindet

Mit diesen Informationen kann der IBM MQ Explorer Folgendes tun:

- Den Repository-Warteschlangenmanager verwenden, um eine Liste der Warteschlangenmanager im Cluster abzurufen.
- Die Warteschlangenmanager verwalten, die Mitglieder des Clusters sind und sich auf unterstützten Plattformen und Befehlsebenen befinden.

Eine Verwaltung ist nicht möglich, wenn:

- Das ausgewählte Repository ist nicht verfügbar. Der IBM MQ Explorer wechselt nicht automatisch zu einem alternativen Repository.
- Das ausgewählte Repository kann nicht über TCP/IP kontaktiert werden.
- Das ausgewählte Repository wird auf einem Warteschlangenmanager ausgeführt, der auf einer Plattform und Befehlsebene aktiv ist, die nicht vom IBM MQ Explorer unterstützt werden.

Die Clustermitglieder, die verwaltet werden können, können lokal sein, oder sie können fern sein, wenn sie über TCP/IP kontaktiert werden können. Der IBM MQ Explorer stellt zu lokalen Warteschlangenmanagern, die Mitglieder eines Clusters sind, direkte Verbindungen her, ohne eine Clientverbindung zu verwenden.

Datenkonvertierung für IBM MQ Explorer

Der IBM MQ Explorer arbeitet in CCSID 1208 (UTF-8). Dies ermöglicht es dem IBM MQ Explorer, die Daten von fernen Warteschlangenmanagern richtig anzuzeigen. Unabhängig davon, ob die Verbindung zu einem Warteschlangenmanager direkt oder über einen temporären Warteschlangenmanager hergestellt wird, erfordert IBM MQ Explorer, dass alle ankommenden Nachrichten nach CCSID 1208 (UTF-8) konvertiert werden.

Es wird eine Fehlernachricht ausgegeben, wenn Sie versuchen, eine Verbindung zwischen dem IBM MQ Explorer und einem Warteschlangenmanager mit einer CCSID herzustellen, die der IBM MQ Explorer nicht erkennt.

Unterstützte Konvertierungen werden im Abschnitt [Codepagekonvertierung](#) beschrieben.

Windows IBM MQ-Taskleistanwendung verwenden (nur Windows)

Die Taskleistanwendung von IBM MQ zeigt ein Symbol in der Windows-Taskleiste auf dem Server an. Das Symbol gibt den aktuellen Status von IBM MQ an und stellt ein Menü bereit, aus dem Sie einfache Aktionen ausführen können.

Unter Windows befindet sich das IBM MQ-Symbol in der Taskleiste auf dem Server und wird mit einem farbcodierten Statussymbol überlagert, das eine der folgenden Bedeutungen haben kann:

Grün

Ordnungsgemäßer Betrieb; keine Alerts vorhanden.

Blau

Unbestimmt; IBM MQ wird gerade gestartet oder beendet.

Gelb

Alert: Ein oder mehrere Services sind fehlgeschlagen oder sind bereits fehlgeschlagen.

Klicken Sie zum Anzeigen des Menüs mit der rechten Maustaste auf das IBM MQ-Symbol. Über das Menü können Sie folgende Aktionen ausführen:

- Klicken Sie auf **Öffnen**, um den IBM MQ-Alertmonitor zu öffnen.
- Klicken Sie auf **Beenden**, um die IBM MQ-Taskleistanwendung zu beenden.
- Klicken Sie auf **IBM MQ Explorer**, um den IBM MQ Explorer zu starten.
- Klicken Sie auf **IBM MQ stoppen**, um IBM MQ zu stoppen.
- Klicken Sie auf **Informationen zu IBM MQ**, um Informationen zum IBM MQ-Alertmonitor anzuzeigen.

Windows Die IBM MQ -Alertmonitoranwendung (nur Windows)

Der IBM MQ-Alertmonitor ist ein Fehlererkennungstool, das Probleme mit IBM MQ auf einer lokalen Maschine identifiziert und erfasst.

Der Alertmonitor zeigt Informationen zum aktuellen Status der lokalen Installation eines IBM MQ-Servers an. Außerdem überwacht er das Windows Advanced Configuration and Power Interface (ACPI) und stellt sicher, dass die ACPI-Einstellungen durchgesetzt werden.

Über den IBM MQ-Alertmonitor können Sie folgende Aktionen ausführen:

- Direkt zugreifen auf den IBM MQ Explorer
- Anzeigen von Informationen zu allen ausstehenden Alerts
- Beenden des IBM MQ-Service auf der lokalen Maschine
- Weiterleiten von Alernachrichten über das Netz an ein konfigurierbares Benutzerkonto oder an eine Windows-Workstation oder einen Windows-Server

Lokale IBM MQ-Objekte verwalten

Sie können lokale IBM MQ-Objekte verwalten, um Anwendungsprogramme zu unterstützen, die die Schnittstelle für Nachrichtenwarteschlangen (Message Queue Interface, MQI) verwenden.

Informationen zu diesem Vorgang

In diesem Kontext bedeutet lokale Verwaltung das Erstellen, Anzeigen, Ändern, Kopieren und Löschen von IBM MQ-Objekten.

Zusätzlich zu den in diesem Abschnitt beschriebenen Methoden können Sie lokale IBM MQ-Objekte auch mit dem IBM MQ Explorer verwalten. Weitere Informationen finden Sie im Abschnitt [„Verwaltung über die IBM MQ Explorer“](#) auf Seite 135.

Prozedur

- Die Informationen in den folgenden Abschnitten helfen Ihnen bei der Verwaltung lokaler IBM MQ-Objekte.
 - [Anwendungsprogramme, die MQI verwenden](#)
 - [„MQ mit Hilfe von MQSC-Befehlen verwalten“](#) auf Seite 10
 - [„Mit Warteschlangenmanagern arbeiten“](#) auf Seite 149
 - [„Mit lokalen Warteschlangen arbeiten“](#) auf Seite 151
 - [„Mit Aliaswarteschlangen arbeiten“](#) auf Seite 156
 - [„Mit Modellwarteschlangen arbeiten“](#) auf Seite 179
 - [„Mit Services arbeiten“](#) auf Seite 187
 - [„Objekte zum Auslösen verwalten“](#) auf Seite 194



Warteschlangenmanager starten und stoppen


Eine Einführung zum Stoppen und Starten eines Warteschlangenmanagers mit Links zu weiteren Informationen.

Informationen zu diesem Vorgang

Sie können einen WS-Manager mit Hilfe von Befehlen starten und stoppen:

- Verwenden Sie zum Starten eines Warteschlangenmanagers den Befehl **strmqm**.
- Um einen Warteschlangenmanager zu stoppen, verwenden Sie den Befehl **endmqm**, der drei Möglichkeiten zum Stoppen eines Warteschlangenmanagers bietet: eine gesteuerte oder ein Quiesce, ein Herunterfahren, ein sofortiges Herunterfahren und ein präventives Herunterfahren.

  Alternativ können Sie unter Windows und Linux auch den IBM MQ Explorer zum Starten und Stoppen eines Warteschlangenmanagers verwenden.

 Unter Windows können Sie einen Warteschlangenmanager so konfigurieren, dass er automatisch gestartet wird, wenn das System mit dem IBM MQ Explorer gestartet wird.

Vorgehensweise

1. Weitere Informationen zum Starten eines Warteschlangenmanagers finden Sie unter [WS-Manager starten](#).
2. Weitere Informationen zum Stoppen eines Warteschlangenmanagers finden Sie im Abschnitt [Warteschlangenmanager stoppen](#).

Wenn die Standardmethoden zum Stoppen und Entfernen eines Warteschlangenmanagers fehlschlagen, können Sie versuchen, den WS-Manager manuell zu stoppen.

Informationen zu diesem Vorgang

Die Standardweise zum Stoppen von Warteschlangenmanagern ist die Verwendung des Befehls **endmqm**, wie im Abschnitt [Warteschlangenmanager stoppen](#) beschrieben. Wenn Sie einen Warteschlangenmanager nicht in der Standardweise stoppen können, können Sie versuchen, einen WS-Manager manuell zu stoppen. Die Art und Weise, in der Sie dies tun, hängt von der verwendeten Plattform ab.

Prozedur

- **Windows**
Informationen zum Stoppen eines Warteschlangenmanagers unter Windows finden Sie im Abschnitt [„Warteschlangenmanager unter Windows manuell stoppen“](#) auf Seite 146.
- **Linux** **UNIX**
Wie Sie einen Warteschlangenmanager auf UNIX oder Linux anhalten können, erfahren Sie unter [„Warteschlangenmanager unter UNIX und Linux manuell stoppen“](#) auf Seite 147.

Zugehörige Informationen

[Warteschlangenmanager auf Multiplatforms erstellen und verwalten](#)
[endmqm](#)

Windows

Warteschlangenmanager unter Windows manuell stoppen

Wenn Sie einen Warteschlangenmanager unter Windows mit dem Befehl **endmqm** nicht stoppen können, können Sie versuchen, den Warteschlangenmanager manuell zu stoppen, indem Sie alle Prozesse beenden, die aktiv sind, und den IBM MQ -Service stoppen.

Informationen zu diesem Vorgang

Tipp: Der Task-Manager von Windows und der Befehl **tasklist** liefern eingeschränkte Informationen zu Tasks. Weitere Informationen zur Ermittlung der Prozesse, die sich auf einen bestimmten Warteschlangenmanager beziehen, finden Sie unter Verwendung eines Tools wie *Process Explorer* (`procexp.exe`), das von der Microsoft -Website unter <https://www.microsoft.com> heruntergeladen werden kann.

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager unter Windows zu stoppen.

Vorgehensweise

1. Listen Sie die Namen (IDs) der aktiven Prozesse auf, indem Sie den Task-Manager von Windows verwenden.
2. Beenden Sie die Prozesse im Task-Manager von Windows oder mit dem Befehl **taskkill** in der folgenden Reihenfolge (sofern sie aktiv sind):

Tabelle 12. Windows-Prozesse, die gestoppt werden sollen, falls sie aktiv sind

Prozessname	Beschreibung
AMQZMUC0	Kritischer Prozessmanager
AMQZXMA0	Ausführungscontroller
AMQZFUMA	OAM-Prozess
AMQZLAA0	LQM-Agenten
AMQZLSA0	LQM-Agenten

<i>Tabelle 12. Windows-Prozesse, die gestoppt werden sollen, falls sie aktiv sind (Forts.)</i>	
Prozessname	Beschreibung
AMQZMUFO	Utility-Manager
AMQZMGR0	Prozesscontroller
AMQZMUR0	Wiederanlauffähiger Prozessmanager
AMQFQPUB	Publish/Subscribe-Prozess
AMQFCXBA	Brokerworker-Prozess
AMQRMPPA	Prozess-Pooling-Prozess
AMQCRSTA	Nicht-Thread-Responder-Jobprozess
AMQCRS6B	LU62 -Empfängerkanal und Clientverbindung
AMQRRMFA	Der Repositoryprozess (für Cluster)
AMQPCSEA	Befehlsserver
RUNMQTRM	Auslösermonitor für einen Server aufrufen
RUNMQDLQ	Handler für dead-letter-Warteschlangen aufrufen
RUNMQCHI	Der Kanalinitiatorprozess
RUNMQLSR	Der Kanal-Listener-Prozess
AMQXSSVN	Gemeinsam genutzte Speicherserver

3. Stoppen Sie den IBM MQ-Service über **Verwaltungstools** > **Dienste** in der Windows-Systemsteuerung.
4. Wenn Sie alle Methoden ausprobiert haben und der WS-Manager nicht gestoppt wurde, führen Sie einen Warmstart Ihres Systems durch.

Warteschlangenmanager unter UNIX und Linux manuell stoppen

Wenn Sie einen Warteschlangenmanager unter UNIX und Linux nicht mit dem Befehl **endmqm** stoppen können, können Sie versuchen, den Warteschlangenmanager manuell zu stoppen, indem Sie alle aktiven Prozesse beenden und den IBM MQ-Service stoppen.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager unter UNIX und Linux zu stoppen.

Wenn Sie den Warteschlangenmanager manuell stoppen, kommt möglicherweise FFST zum Einsatz und werden FDC-Dateien in `/var/mqm/errors` abgelegt. Dies sollte nicht als Fehler im WS-Manager betrachtet werden.

Der WS-Manager wird normal erneut gestartet, auch wenn er mit dieser Methode gestoppt wurde, um ihn manuell zu stoppen.

Vorgehensweise

1. Suchen Sie die Prozess-IDs der Warteschlangenmanagerprogramme, die noch mit dem Befehl **ps** ausgeführt werden.

Wenn der WS-Manager beispielsweise QMNAME genannt wird, verwenden Sie den folgenden Befehl:

```
ps -ef | grep QMNAME
```

2. Beenden Sie alle Warteschlangenmanagerprozesse, die noch aktiv sind, mit dem Befehl **kill** und geben Sie dabei die PIDs an, die mit dem Befehl **ps** erkannt wurden.

Verwenden Sie zum Beenden eines Prozesses entweder den Befehl **kill -KILL <pid>** oder den entsprechenden Befehl **kill -9 <pid>**.

Sie müssen die PIDs, die Sie töten möchten, nacheinander durcharbeiten und diesen Befehl jedes Mal ausgeben.

Wichtig: Wenn Sie ein anderes Signal als **9 (SIGKILL)** verwenden, wird der Prozess wahrscheinlich nicht gestoppt und Sie erhalten unvorhersehbare Ergebnisse.

Beenden Sie die Prozesse in der folgenden Reihenfolge:

Tabelle 13. UNIX- und Linux-Prozesse, die gestoppt werden sollen, falls sie aktiv sind

Prozessname	Beschreibung
amqzmuc0	Kritischer Prozessmanager
amqzxma0	Ausführungscontroller
amqzfuma	OAM-Prozess
amqzlaa0	LQM-Agenten
amqzlsa0	LQM-Agenten
amqzmuf0	Utility-Manager
amqzmur0	Wiederanlauffähiger Prozessmanager
amqzmgr0	Prozesscontroller
amqfqpub	Publish/Subscribe-Prozess
amqfcxba	Brokerworker-Prozess
amqrmppa	Prozess-Pooling-Prozess
amqcrsta	Nicht-Thread-Responder-Jobprozess
amqcrs6b	LU62 -Empfängerkanal und Clientverbindung
amqrrmfa	Der Repositoryprozess (für Cluster)
amqpcsea	Befehlsserver
runmqtrm	Auslösermonitor für einen Server aufrufen
runmqdlq	Handler für dead-letter-Warteschlangen aufrufen
runmqchi	Der Kanalinitiatorprozess
runmqlsr	Der Kanal-Listener-Prozess

Anmerkung: Mit dem Befehl **kill -9** können Sie Prozesse beenden, die nicht gestoppt werden können.

MQI-Kanäle stoppen

Wenn Sie einen STOP CHANNEL-Befehl für einen Serververbindungskanal ausgeben, können Sie auswählen, welche Methode zum Stoppen des Clientverbindungskanals verwendet werden soll. Dies bedeutet, dass ein Clientkanal, der einen MQGET-Warteaufruf ausgibt, gesteuert werden kann, und Sie können entscheiden, wie und wann der Kanal gestoppt werden soll.

Der Befehl STOP CHANNEL kann mit drei Modi ausgeführt werden, die angeben, wie der Kanal gestoppt werden soll:

QUIESCE

Stoppt den Kanal, nachdem alle aktuellen Nachrichten verarbeitet wurden.

Wenn die gemeinsame Nutzung von Konversationen aktiviert ist, wird der IBM MQ MQI client die Stoppanforderung rechtzeitig erkennen; diese Zeit hängt von der Geschwindigkeit des Netzes ab. Die Clientanwendung wird durch die Ausgabe eines nachfolgenden Aufrufs an IBM MQ auf die Stoppanforderung aufmerksam.

Erzwingen

Stoppt den Kanal sofort.

Beenden

Stoppt den Kanal sofort. Wenn der Kanal als Prozess ausgeführt wird, kann er den Kanalprozess beenden oder wenn der Kanal als Thread ausgeführt wird, sein Thread.

Dies ist ein mehrstufiger Prozess. Wird der Terminate-Modus verwendet, wird versucht, den Serververbindungskanal zu stoppen, zunächst mit dem Quiesce-Modus, dann mit dem Force-Modus und, falls erforderlich, mit dem Terminate-Modus. Der Client kann während der verschiedenen Phasen der Beendigung unterschiedliche Rückkehrcodes empfangen. Wenn der Prozess oder Thread beendet wird, empfängt der Client einen Kommunikationsfehler.

Die an die Anwendung zurückgegebenen Rückkehrcodes variieren je nach dem ausgegebenen MQI-Aufruf und dem ausgegebenen Befehl STOP CHANNEL. Der Client empfängt entweder einen MQRC_CONNECTION_QUIESCING-oder einen MQRC_CONNECTION_BROKEN-Rückkehrcode. Wenn ein Client MQRC_CONNECTION_QUIESCING erkennt, sollte er versuchen, die aktuelle Transaktion abzuschließen und zu beenden. Dies ist mit MQRC_CONNECTION_BROKEN nicht möglich. Wenn der Client die Transaktion nicht beendet und schnell genug beendet wird, wird CONNECTION_BROKEN nach ein paar Sekunden abgerufen. Ein STOP CHANNEL-Befehl mit MODE (FORCE) oder MODE (TERMINATE) führt eher zu einer CONNECTION_BROKEN als mit MODE (QUIESCE).

Zugehörige Informationen

[Kanäle](#)

Mit Warteschlangenmanagern arbeiten

Beispiele für MQSC-Befehle, die Sie zum Anzeigen oder Ändern von Warteschlangenmanagerattributen verwenden können.

WS-Manager-Attribute anzeigen

Verwenden Sie den MQSC-Befehl **DISPLAY QMGR** , um die Attribute des Warteschlangenmanagers anzuzeigen, der im Befehl **runmqsc** angegeben ist:

```
DISPLAY QMGR
```

Eine typische Ausgabe dieses Befehls wird in [Abbildung 16 auf Seite 150](#) gezeigt.

```

DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408: Display Queue Manager details.
QMNAME(QM1)
ACCTINT(1800)
ACCTQ(OFF)
ACTVCONO (DISABLED)
ALTDATE(2012-05-27)
AUTHOREV(DISABLED)
CHAD(DISABLED)
CHADEXIT( )
CLWLDATA( )
CLWLLEN(100)
CLWLUSEQ(LOCAL)
CMDLEVEL(800)
CONFIGEV(DISABLED)
CRTIME(16.14.01)
DEFXMITQ( )
DISTL(YES)
IPADDRV(IPV4)
LOGGEREV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWSNT)
PSNPMMSG(DISCARD)
PSSYNCP(1FPER)
PSMODE(ENABLED)
REPOS( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRYP( )
SSLFIPS(NO)
MQ\Data\qmgrs\QM1\ssl\key)
SSLKEYC(0)
STATCHL(OFF)
STATMQI(OFF)
STRSTPEV(ENABLED)
TREELIFE(1800)
ACCTCONO(DISABLED)
ACCTMQI(OFF)
ACTIVREC(MSG)
ACTVTRC(OFF)
ALTTIME(16.14.01)
CCSID(850)
CHADEV(DISABLED)
CHLEV(DISABLED)
CLWLXIT( )
CLWLMRUC(999999999)
CMDEV(DISABLED)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
CRDATE(2011-05-27)
DEADQ( )
DESCR( )
INHIBTEV(DISABLED)
LOCALEV(DISABLED)
MARKINT(5000)
MAXMSGL(4194304)
MAXPRTY(9)
MONACLS(QMGR)
MONQ(OFF)
PERFMEV(DISABLED)
  PSRTCNT(5)
PSNPRES(NORMAL)
QMID(QM1_2011-05-27_16.14.01)
REMOETEVEV(DISABLED)
REPOSNL( )
SCHINIT(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
SSLKEYR(C:\Program Files\IBM\WebSphere
STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SYNCP
TRIGINT(999999999)

```

Abbildung 16. Typische Ausgabe des Befehls `DISPLAY QMGR`

Anmerkung: SYNCP ist ein schreibgeschütztes Warteschlangenmanagerattribut.

Der Parameter **ALL** ist der Standardwert für den Befehl **DISPLAY QMGR**. Es zeigt alle Attribute des Warteschlangenmanagers an. In der Ausgabe werden insbesondere der Name des Standardwarteschlangenmanagers, der Name der Warteschlange für dead-Mail und der Name der Befehlswarteschlange angegeben.

Sie können bestätigen, dass diese Warteschlangen vorhanden sind, indem Sie den folgenden Befehl eingeben:

```
DISPLAY QUEUE (SYSTEM.*)
```

Daraufhin wird eine Liste der Warteschlangen angezeigt, die mit dem Stammverzeichnis `SYSTEM.*` übereinstimmen. Die runden Klammern sind erforderlich.

Ändern von WS-Managerattributen

Um die Attribute des im Befehl `runmqsc` angegebenen WS-Managers zu ändern, verwenden Sie den MQSC-Befehl **ALTER QMGR** und geben Sie die Attribute und Werte an, die Sie ändern möchten. Verwenden Sie z. B. die folgenden Befehle, um die Attribute von `jupiter.queue.manager` zu ändern:

```
runmqsc jupiter.queue.manager
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

Mit dem Befehl **ALTER QMGR** wird die verwendete Warteschlange für dead-letter geändert und die Sperrereignisse aktiviert.

Zugehörige Informationen

[Warteschlangenmanager auf Multiplatforms erstellen und verwalten](#)

[Attribute für den Warteschlangenmanager](#)

[runmqsc \(MQSC-Befehle ausführen\)](#)

[ANZEIGEN QMGR](#)

[ALTER QMGR](#)

Mit lokalen Warteschlangen arbeiten

Dieser Abschnitt enthält Beispiele für einige MQSC-Befehle, die Sie zum Verwalten von lokalen, Modell- und Aliaswarteschlangen verwenden können.

Detaillierte Informationen zu diesen Befehlen finden Sie in den [MQSC-Befehlen](#).

Zugehörige Informationen

[Benennungseinschränkungen für Warteschlangen](#)

[Benennungseinschränkungen für andere Objekte](#)

Lokale Warteschlange definieren

Bei einer Anwendung ist der lokale WS-Manager der Warteschlangenmanager, mit dem die Anwendung verbunden ist. Warteschlangen, die vom lokalen WS-Manager verwaltet werden, werden als lokale Warteschlangen für diesen Warteschlangenmanager angegeben.

Verwenden Sie den MQSC-Befehl **DEFINE QLOCAL**, um eine lokale Warteschlange zu erstellen. Sie können auch den Standardwert verwenden, der in der lokalen Standardwarteschlangendefinition definiert ist, oder Sie können die Warteschlangenkenndaten aus denen der lokalen Standardwarteschlange ändern.

Anmerkung: Die lokale Standardwarteschlange hat den Namen SYSTEM.DEFAULT.LOCAL.QUEUE, und sie wurde bei der Systeminstallation erstellt.

Der folgende Befehl **DEFINE QLOCAL** definiert beispielsweise eine Warteschlange mit dem Namen ORANGE.LOCAL.QUEUE mit folgenden Merkmalen:

- Sie ist aktiviert für Abrufe, aktiviert für die Option "puts" und arbeitet mit einer Prioritätsauftragsbasis.
- Es handelt sich um eine *normale* Warteschlange; es handelt sich nicht um eine Initialisierungswarteschlange oder eine Übertragungswarteschlange, und es werden keine Auslösenachrichten generiert.
- Die maximale Warteschlangenlänge beträgt 5000 Nachrichten; die maximale Nachrichtenlänge beträgt 4194304 Byte.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT(ENABLED) +
  GET(ENABLED) +
  NOTRIGGER +
  MSGDLVSQ(PRIORITY) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  USAGE(NORMAL)
```

Anmerkung:

1. Mit Ausnahme des Werts für die Beschreibung sind alle Attributwerte, die in dem Beispiel dargestellt sind, die Standardwerte. Diese Beispiele sind für die Illustration purposes definitiv eingeschlossen. Sie können sie übergehen, wenn Sie sicher sind, dass die Standardwerte die gewünschten Werte sind oder nicht geändert wurden. Weitere Informationen hierzu finden Sie im Abschnitt [„Standardobjektattribute anzeigen“](#) auf Seite 152.
2. **USAGE(NORMAL)** gibt an, dass es sich bei dieser Warteschlange nicht um eine Übertragungswarteschlange handelt.

3. Wenn bereits eine lokale Warteschlange auf demselben WS-Manager mit dem Namen ORANGE.LOCAL.QUEUE vorhanden ist, schlägt dieser Befehl fehl. Verwenden Sie das Attribut **REPLACE**, wenn die vorhandene Warteschlangendefinition überschrieben werden soll. Lesen Sie dazu auch [„Attribute der lokalen Warteschlange ändern“](#) auf Seite 153.

Zugehörige Informationen

[QLOCAL DEFINIER](#)

Standardobjektattribute anzeigen

Mit dem Befehl **DISPLAY QUEUE** können Attribute angezeigt werden, die bei der Definition eines IBM MQ-Objekts vom Standardobjekt übernommen wurden.

Wenn Sie ein IBM MQ-Objekt definieren, werden alle Attribute, die Sie nicht angeben, aus dem Standardobjekt verwendet. Wenn Sie beispielsweise eine lokale Warteschlange definieren, übernimmt die Warteschlange alle Attribute, die Sie in der Definition aus der lokalen Standardwarteschlange weglassen, die als SYSTEM.DEFAULT.LOCAL.QUEUE bezeichnet wird. Um genau zu sehen, welche Attribute diese Attribute haben, verwenden Sie den folgenden Befehl:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Die Syntax dieses Befehls unterscheidet sich von der Syntax des entsprechenden Befehls **DEFINE**. Im Befehl **ANZEIGEN** können Sie nur den Namen der Warteschlange angeben, während Sie im Befehl **DEFINE** den Typ der Warteschlange angeben müssen, d. h. QLOCAL, QALIAS, QMODEL oder QREMOTE.

Sie können Attribute selektiv anzeigen, indem Sie sie einzeln angeben. Beispiel:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH;
```

Mit diesem Befehl werden die drei angegebenen Attribute wie folgt angezeigt:

```
AMQ8409: Display Queue details.  
QUEUE(ORANGE.LOCAL.QUEUE)      TYPE(QLOCAL)  
CURDEPTH(0)                     MAXDEPTH(5000)  
MAXMSGL(4194304)
```

CURDEPTH ist die aktuelle Warteschlangenlänge, d. h. die Anzahl der Nachrichten in der Warteschlange. Dies ist ein nützliches Attribut, das angezeigt werden kann, da durch die Überwachung der Warteschlangenlänge sichergestellt werden kann, dass die Warteschlange nicht voll wird.

Zugehörige Informationen

[ANZEIGEN QUEUE](#)

[Warteschlangen definieren \(DEFINE\)](#)

Lokale Warteschlangendefinition kopieren

Sie können eine Warteschlangendefinition mit dem Attribut **LIKE** im Befehl **DEFINE** kopieren.

Beispiel:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

Mit diesem Befehl wird eine Warteschlange mit denselben Attributen wie die ursprüngliche Warteschlange ORANGE.LOCAL.QUEUE erstellt, und nicht die Attribute der lokalen Standardwarteschlange. Geben Sie den Namen der Warteschlange ein, die exakt so kopiert werden soll, wie sie beim Erstellen der

Warteschlange eingegeben wurde. Wenn der Name Kleinbuchstaben enthält, schließen Sie den Namen in einfache Anführungszeichen ein.

Sie können diese Form des Befehls **DEFINE** auch verwenden, um eine Warteschlangendefinition zu kopieren, jedoch eine oder mehrere Änderungen an den Attributen des Originals zu ersetzen. Beispiel:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

Dieser Befehl kopiert die Attribute der Warteschlange ORANGE.LOCAL.QUEUE in die Warteschlange THIRD.QUEUE, gibt jedoch an, dass die maximale Nachrichtenlänge in der neuen Warteschlange 1024 Byte und nicht 4194304 betragen soll.

Anmerkung:

1. Wenn Sie das Attribut **LIKE** in einem **DEFINE** -Befehl verwenden, kopieren Sie die Warteschlangenattribute nur. Die Nachrichten werden nicht in die Warteschlange kopiert.
2. Wenn Sie eine lokale Warteschlange definieren, ohne **LIKE** anzugeben, ist sie mit der Angabe von **DEFINE LIKE (SYSTEM.DEFAULT.LOCAL.QUEUE)** identisch.

Zugehörige Informationen

[Warteschlangen definieren \(DEFINE\)](#)

Attribute der lokalen Warteschlange ändern

Sie können die Warteschlangenattribute auf zwei Arten ändern, indem Sie entweder den Befehl **ALTER QLOCAL** oder den Befehl **DEFINE QLOCAL** mit dem Attribut **REPLACE** verwenden.

In „[Lokale Warteschlange definieren](#)“ auf Seite 151 wurde die Warteschlange mit dem Namen ORANGE.LOCAL.QUEUE definiert. Nehmen wir zum Beispiel an, dass Sie die maximale Nachrichtenlänge in dieser Warteschlange auf 10.000 Byte reduzieren möchten.

- Mit dem Befehl **ALTER** :

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Mit diesem Befehl wird ein einzelnes Attribut geändert, das der maximalen Nachrichtenlänge entspricht. Alle anderen Attribute bleiben unverändert.

- Verwenden Sie den Befehl **DEFINE** mit der Option **REPLACE**, z. B.:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Mit diesem Befehl werden nicht nur die maximale Nachrichtenlänge, sondern auch alle anderen Attribute geändert, die ihre Standardwerte erhalten. Die Warteschlange ist jetzt aktiviert, während sie zuvor gesperrt wurde. Put aktiviert ist der Standardwert, wie er in der Warteschlange SYSTEM.DEFAULT.LOCAL.QUEUE angegeben ist.

Wenn Sie die maximale Nachrichtenlänge in einer vorhandenen Warteschlange verringern, sind die vorhandenen Nachrichten nicht betroffen. Alle neuen Nachrichten müssen jedoch die neuen Kriterien erfüllen.

Zugehörige Informationen

[ALTER-Warteschlangen](#)

[ALTER QLOCAL](#)

[Warteschlangen definieren \(DEFINE\)](#)

[QLOCAL DEFINIER](#)

Löschen einer lokalen Warteschlange

Mit dem Befehl **CLEAR** können Sie den Inhalt einer lokalen Warteschlange löschen.

Wenn Sie z. B. alle Nachrichten aus einer lokalen Warteschlange mit dem Namen MAGENTA.QUEUE löschen möchten, verwenden Sie den folgenden Befehl:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

Anmerkung: Es gibt keine Eingabeaufforderung, mit der Sie Ihre Meinung ändern können. Sobald Sie die Eingabetaste drücken, gehen die Nachrichten verloren.

Sie können eine Warteschlange nicht löschen, wenn:

- Es sind nicht festgeschriebene Nachrichten vorhanden, die unter Synchronisationspunkt in die Warteschlange gestellt wurden.
- Eine Anwendung hat momentan die Warteschlange geöffnet.

Zugehörige Informationen

CLEAR QLOCAL

Lokale Warteschlange löschen

Sie können den MQSC-Befehl **DELETE QLOCAL** verwenden, um eine lokale Warteschlange zu löschen.

Eine Warteschlange kann nicht gelöscht werden, wenn sie nicht festgeschriebene Nachrichten enthält. Wenn die Warteschlange jedoch eine oder mehrere festgeschriebene Nachrichten und keine nicht festgeschriebenen Nachrichten enthält, kann sie nur gelöscht werden, wenn Sie die Option **PURGE** angeben. Beispiel:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

Wenn Sie **NOPURGE** anstelle von **PURGE** angeben, wird sichergestellt, dass die Warteschlange nicht gelöscht wird, wenn sie festgeschriebene Nachrichten enthält.

Zugehörige Informationen

DELETE QLOCAL

Warteschlangen durchsuchen

IBM MQ stellt einen Beispielwarteschlangenbrowser bereit, mit dem Sie den Inhalt der Nachrichten in einer Warteschlange anzeigen können. Der Browser wird sowohl in der Quellen- als auch in der ausführbaren Datei bereitgestellt.

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

Windows Unter Windows lauten die Dateinamen und Pfade für den Beispielwarteschlangenbrowser wie folgt:

Quelle

MQ_INSTALLATION_PATH\tools\c\samples\

Ausführbare Datei

MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcbg.exe

Linux **UNIX** Unter UNIX und Linux lauten die Dateinamen und Pfade wie folgt:

Quelle

MQ_INSTALLATION_PATH/samp/amqsbcbg0.c

Ausführbare Datei

MQ_INSTALLATION_PATH/samp/bin/amqsbcbg

Windows Auf Windows-Systemen steht die Unterstützung für große Dateien ohne zusätzliche Unterstützung zur Verfügung.

Linux **UNIX** Auf UNIX- und Linux-Systemen müssen Sie die Unterstützung großer Dateien explizit aktivieren, bevor Sie Warteschlangendateien mit einer Größe von mehr als 2 GB erstellen können. Weitere Informationen hierzu finden Sie in der Dokumentation zum Betriebssystem.

Einige Dienstprogramme, wie z. B. 'tar', können Dateien mit mehr als 2 GB nicht bewältigen. Überprüfen Sie vor der Aktivierung der Unterstützung für große Dateien die Dokumentation zum Betriebssystem auf Informationen zu Einschränkungen bei den Dienstprogrammen, die Sie verwenden.

Informationen zur Planung der erforderlichen Speichermenge für Warteschlangen finden Sie in [IBM MQ Family-Performance Reports](#) für plattformspezifische Leistungsberichte.

Mit Aliaswarteschlangen arbeiten

Sie können eine Aliaswarteschlange definieren, um indirekt auf eine andere Warteschlange oder ein anderes Thema zu verweisen.

V9.0.0.1 **V9.0.1**



Achtung: Verteilerlisten unterstützen nicht die Verwendung von Aliaswarteschlangen, die auf Topic-Objekte verweisen. Ab IBM MQ 9.0.1 und IBM MQ 9.0.0 Fix Pack 1 gibt IBM MQ MQRC_ALIAS_BASE_Q_TYPE_ERROR zurück, wenn eine Aliaswarteschlange auf ein Themenobjekt in einer Verteilerliste verweist.

Für die Warteschlange, auf die sich eine Aliaswarteschlange bezieht, kann es sich um eine der folgenden Warteschlangen handeln:

- Eine lokale Warteschlange (siehe „[Lokale Warteschlange definieren](#)“ auf Seite 151).
- Eine lokale Definition einer fernen Warteschlange (siehe „[Lokale Definition einer fernen Warteschlange erstellen](#)“ auf Seite 208).
- Ein Thema.

Eine Aliaswarteschlange ist keine echte Warteschlange, sondern eine Definition, die zur Ausführungszeit in eine reale Warteschlange (oder Zielwarteschlange) aufgelöst wird. Die Definition der Aliaswarteschlange gibt die Zielwarteschlange an. Wenn eine Anwendung einen MQOPEN -Aufruf in eine Aliaswarteschlange aufruft, löst der Warteschlangenmanager den Aliasnamen in den Namen der Zielwarteschlange auf.

Eine Aliaswarteschlange kann nicht in eine andere lokal definierte Aliaswarteschlange aufgelöst werden. Eine Aliaswarteschlange kann jedoch in Aliaswarteschlangen aufgelöst werden, die an anderer Stelle in Clustern definiert sind, in denen der lokale WS-Manager Mitglied ist. Weitere Informationen finden Sie in der [Namensauflösung](#).

Aliaswarteschlangen sind nützlich für:

- verschiedenen Anwendungen unterschiedliche Ebenen von Zugriffsberechtigungen für die Zielwarteschlange geben.
- Es können verschiedene Anwendungen auf unterschiedliche Weise mit derselben Warteschlange arbeiten. (Möglicherweise möchten Sie andere Standardprioritäten oder unterschiedliche Standardpersistenzwerte zuordnen.)
- Vereinfachung der Wartung, Migration und Lastverteilung. (Möglicherweise möchten Sie den Namen der Zielwarteschlange ändern, ohne Ihre Anwendung ändern zu müssen, die weiterhin den Aliasnamen verwendet.)

Angenommen, es wurde eine Anwendung entwickelt, um Nachrichten in eine Warteschlange mit dem Namen MY.ALIAS.QUEUE zu stellen. Sie gibt den Namen dieser Warteschlange an, wenn sie eine MQOPEN -Anforderung und indirekt eine Nachricht in diese Warteschlange eingibt. Der Anwendung ist nicht bekannt, dass es sich bei der Warteschlange um eine Aliaswarteschlange handelt. Für jeden MQI-Aufruf, der diesen Aliasnamen verwendet, löst der Warteschlangenmanager den Namen der realen Warteschlange

auf, die entweder eine lokale Warteschlange oder eine ferne Warteschlange sein kann, die in diesem WS-Manager definiert ist.

Wenn Sie den Wert des Attributs TARGQ ändern, können Sie MQI-Aufrufe in eine andere Warteschlange umleiten, möglicherweise auf einem anderen Warteschlangenmanager. Dies ist hilfreich bei der Wartung, Migration und Lastverteilung.

Definieren einer Aliaswarteschlange

Mit dem folgenden Befehl wird eine Aliaswarteschlange erstellt:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Mit diesem Befehl werden MQI-Aufrufe umgeleitet, die MY.ALIAS.QUEUE in der Warteschlange YELLOW.QUEUE angeben. Der Befehl erstellt die Zielwarteschlange nicht; die MQI-Aufrufe schlagen fehl, wenn die Warteschlange YELLOW.QUEUE zur Ausführungszeit nicht vorhanden ist.

Wenn Sie die Aliasdefinition ändern, können Sie die MQI-Aufrufe an eine andere Warteschlange umleiten. Beispiel:

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

Mit diesem Befehl werden MQI-Aufrufe an eine andere Warteschlange, MAGENTA.QUEUE, umgeleitet.

Sie können auch Aliaswarteschlangen verwenden, um eine einzelne Warteschlange (die Zielwarteschlange) zu erstellen, um unterschiedliche Attribute für verschiedene Anwendungen zu haben. Definieren Sie dies, indem Sie zwei Aliasnamen definieren, eine für jede Anwendung. Angenommen, es gibt zwei Anwendungen:

- Die Anwendung ALPHA kann Nachrichten in YELLOW.QUEUE einlegen, aber es ist nicht zulässig, Nachrichten von ihr abzurufen.
- Die Anwendung BETA kann Nachrichten von YELLOW.QUEUE abrufen, aber es ist nicht zulässig, Nachrichten in diese Datei zu stellen.

Mit dem folgenden Befehl wird ein Aliasname definiert, der aktiviert ist und für die Anwendung ALPHA inaktiviert wird:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (ENABLED) +  
GET (DISABLED)
```

Mit dem folgenden Befehl wird ein Alias definiert, der inaktiviert ist und für die Anwendung BETA aktiviert wird:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (DISABLED) +  
GET (ENABLED)
```

ALPHA verwendet in seinen MQI-Aufrufen den Warteschlangennamen ALPHAS.ALIAS.QUEUE; BETA verwendet den Warteschlangennamen BETAS.ALIAS.QUEUE. Beide greifen auf die gleiche Warteschlange zu, aber auf unterschiedliche Weise.

Sie können die Attribute LIKE und REPLACE verwenden, wenn Sie Warteschlangenaliasnamen auf die gleiche Weise definieren, wie Sie diese Attribute mit lokalen Warteschlangen verwenden.

Andere Befehle mit Aliaswarteschlangen verwenden

Sie können die entsprechenden MQSC-Befehle verwenden, um Aliaswarteschlangenattribute anzuzeigen oder zu ändern oder um das Aliaswarteschlangenobjekt zu löschen. Beispiel:

Verwenden Sie den Befehl **DISPLAY QALIAS**, um die Attribute der Aliaswarteschlange anzuzeigen:

```
DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE)
```

Verwenden Sie den Befehl **ALTER QALIAS**, um den Namen der Basiswarteschlange zu ändern, in den der Aliasname aufgelöst wird, wobei die Option `force` die Änderung auch dann erzwingt, wenn die Warteschlange geöffnet ist:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGQ(ORANGE.LOCAL.QUEUE) FORCE
```

Verwenden Sie den Befehl **DELETE QALIAS**, um diesen Warteschlangenalias zu löschen:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Sie können eine Aliaswarteschlange nicht löschen, wenn eine Anwendung momentan die Warteschlange geöffnet hat.

Zugehörige Informationen

[ALTER QALIAS](#)

[DEFINE QALIAS](#)

[DELETE QALIAS](#)

[Verteilerlisten](#)

Mit dead-letter-Warteschlangen arbeiten

Jeder WS-Manager verfügt in der Regel über eine lokale Warteschlange, die als Warteschlange für einen dead-letter verwendet wird, sodass Nachrichten, die nicht an ihr korrektes Ziel zugestellt werden können, für einen späteren Abruf gespeichert werden können. Sie informieren den Warteschlangenmanager über die Warteschlange für nicht zustellbare Nachrichten und geben an, wie Nachrichten in einer Warteschlange für dead-letter verarbeitet werden sollen. Die Verwendung von Warteschlangen mit dead-Mail kann sich auf die Reihenfolge auswirken, in der Nachrichten zugestellt werden, so dass Sie sie nicht verwenden können.

Geben Sie zum Erzählen des WS-Managers in der Warteschlange für dead-Mail den Namen einer Warteschlange für dead-letter auf dem Befehl **crtmqm** (z. B. `crtmqm -u DEAD.LETTER.QUEUE`) an, oder verwenden Sie das Attribut **DEADQ** im Befehl **ALTER QMGR**, um eine spätere Angabe zu machen. Sie müssen die Warteschlange für den dead-letter definieren, bevor Sie sie verwenden.

Es ist eine Beispiel-Deadletter-Warteschlange mit dem Namen `SYSTEM.DEAD.LETTER.QUEUE` mit dem Produkt verfügbar. Diese Warteschlange wird bei der Erstellung des Warteschlangenmanagers automatisch erstellt. Sie können diese Definition bei Bedarf ändern und sie umbenennen.

Eine Warteschlange für nicht zustellbare Nachrichten hat keine besonderen Anforderungen mit Ausnahme der folgenden:

- Es muss sich um eine lokale Warteschlange handeln.
- Das Attribut **MAXMSGL** (maximale Nachrichtenlänge) muss die Warteschlange aktivieren, damit die größten Nachrichten, die der Warteschlangenmanager verarbeiten muss, **plus** die Größe des Headers "dead-letter" (`MQDLH`) erfüllen müssen.

Die Verwendung von Warteschlangen mit dead-Mail kann sich auf die Reihenfolge auswirken, in der Nachrichten zugestellt werden, so dass Sie sie nicht verwenden können. Sie legen das Kanalattribut `USEDLQ` fest, um zu bestimmen, ob die Warteschlange für dead-letter verwendet wird, wenn Nachrichten nicht zugestellt werden können. Dieses Attribut kann so konfiguriert werden, dass einige Funktionen des Warteschlangenmanagers die Warteschlange für dead-letter verwenden, während andere Funktionen dies nicht tun. Weitere Informationen zur Verwendung des Kanalattributs `USEDLQ` für verschiedene `MQSC`-Befehle finden Sie in [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) und [DISPLAY CLUSQMGR](#).

IBM MQ stellt einen DLQ-Handler (Steuerroutine der Warteschlange für nicht zustellbare Nachrichten) bereit, mit dem Sie angeben können, wie Nachrichten, die in einer Warteschlange für nicht zustellbare Nachrichten gefunden werden, verarbeitet oder entfernt werden sollen. Weitere Informationen finden Sie im Abschnitt „[Nachrichten in einer IBM MQ-Warteschlange für nicht zustellbare Nachrichten verarbeiten](#)“ auf Seite 159.

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

ALTER QMGR

[crtmqm \(WS-Manager erstellen\)](#)

Nachrichten in einer IBM MQ-Warteschlange für nicht zustellbare Nachrichten verarbeiten

Für die Verarbeitung von Nachrichten in einer Warteschlange für nicht zustellbare Nachrichten (DLQ) stellt IBM MQ einen Standard-DLQ-Handler zur Verfügung. Der Handler gleicht Nachrichten in der DLQ mit Einträgen in einer Regeltabelle ab, die Sie definieren.

Nachrichten können von Warteschlangenmanagern, Nachrichtenkanalagenten (MCAs) und Anwendungen in eine DLQ gestellt werden. Alle Nachrichten in der DLQ müssen mit einer *Headerstruktur einer nicht zustellbaren Nachricht* (MQDLH) als Präfix vorangesetzt werden. Nachrichten, die von einem Warteschlangenmanager oder einem Nachrichtenkanalagenten in die DLQ gestellt werden, haben immer diesen Header; Anwendungen, die Nachrichten in die DLQ stellen, müssen diese Kopfzeile angeben. Das Feld *Reason* der MQDLH-Struktur enthält einen Ursachencode, der angibt, warum sich die Nachricht in der DLQ befindet.

Alle IBM MQ-Umgebungen benötigen eine Routine, um Nachrichten in der DLQ regelmäßig zu verarbeiten. IBM MQ stellt eine Standardroutine bereit, die als *DLQ-Handler* (Steuerroutine der Warteschlange für nicht zustellbare Nachrichten) bezeichnet wird und die Sie mit dem Befehl `runmqdlq` aufrufen.

Anweisungen für die Verarbeitung von Nachrichten in der DLQ werden dem DLQ-Handler mit Hilfe einer vom Benutzer geschriebenen *Regeltabelle* bereitgestellt. Das heißt, der DLQ-Handler gleicht Nachrichten in der Warteschlange für DLQ mit Einträgen in der Regeltabelle ab; wenn eine DLQ-Nachricht mit einem Eintrag in der Regeltabelle übereinstimmt, führt der DLQ-Handler die Aktion aus, die diesem Eintrag zugeordnet ist.

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

Steuerroutine der Warteschlange für nicht zustellbare Nachrichten (DLQ-Handler) unter IBM i

Was ist die Steuerroutine der Warteschlange für nicht zustellbare Nachrichten unter IBM i und wie rufen Sie sie auf?

Eine *Warteschlange für nicht zustellbare Nachrichten* (DLQ), die manchmal auch als *Warteschlange für nicht zugestellte Nachrichten* bezeichnet wird, ist eine Haltewarteschlange für Nachrichten, die nicht an ihre Zielwarteschlangen zugestellt werden können. Jedem Warteschlangenmanager in einem Netz sollte eine DLQ zugeordnet sein.

Anmerkung: Es ist häufig vorzuziehen, Nachrichten in einer DLQ zu vermeiden. Informationen zur Verwendung und Vermeidung von DLQs finden Sie im Abschnitt „[Mit dead-letter-Warteschlangen arbeiten](#)“ auf Seite 158.

Warteschlangenmanager, Nachrichtenkanalagenten und Anwendungen können Nachrichten in den DLQ stellen. Alle Nachrichten in der DLQ müssen mit einer *Headerstruktur einer nicht zustellbaren Nachricht* (MQDLH) als Präfix vorangesetzt werden. Nachrichten, die von einem Warteschlangenmanager oder von einem Nachrichtenkanalagenten in die DLQ gestellt werden, verfügen immer über einen MQDLH. Für An-

wendungen, die Nachrichten in die DLQ stellen, immer einen MQDLH-Wert angeben. Das Feld *Reason* der MQDLH-Struktur enthält einen Ursachencode, der angibt, warum sich die Nachricht in der DLQ befindet.

In allen IBM MQ-Umgebungen muss es eine Routine geben, die regelmäßig ausgeführt wird, um Nachrichten in der DLQ zu verarbeiten. IBM MQ stellt eine Standardroutine bereit, die als *DLQ-Handler* (Steuerroutine der Warteschlange für nicht zustellbare Nachrichten) bezeichnet wird und die Sie mit dem Befehl STRMQMDLQ aufrufen. Eine vom Benutzer geschriebene *Regeltabelle* stellt Anweisungen an den DLQ-Handler zur Verarbeitung von Nachrichten in der Warteschlange für nicht zustellbare Nachrichten (DLQ) bereit. Das heißt, der DLQ-Handler gleicht Nachrichten in der Warteschlange für DLQ mit Einträgen in der Regeltabelle ab. Wenn eine DLQ-Nachricht mit einem Eintrag in der Regeltabelle übereinstimmt, führt der DLQ-Handler die Aktion aus, die diesem Eintrag zugeordnet ist.

DLQ-Handler aufrufen

Verwenden Sie den Befehl STRMQMDLQ, um den DLQ-Handler aufzurufen. Sie können die DLQ, die Sie verarbeiten wollen, und den Warteschlangenmanager, den Sie verwenden möchten, auf zwei Arten benennen:

- Als Parameter für STRMQMDLQ über die Eingabeaufforderung. Zum Beispiel:

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QTXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- In der Regeltabelle. Zum Beispiel:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE)
```

Anmerkung: Die Regeltabelle ist eine Teildatei in einer physischen Quellendatei, die einen beliebigen Namen annehmen kann.

Die Beispiele gelten für die DLQ ABC1.DEAD.LETTER.QUEUE, deren Eigner der Standardwarteschlangenmanager ist.

Wenn Sie die DLQ oder den Warteschlangenmanager nicht wie gezeigt angeben, wird der Standardwarteschlangenmanager für die Installation zusammen mit der DLQ, die zu diesem Warteschlangenmanager gehört, verwendet.


Der Befehl STRMQMDLQ verwendet seine Eingabe aus der Regeltabelle.

Sie müssen berechtigt sein, sowohl auf die DLQ selbst als auch auf alle Nachrichtenwarteschlangen zuzugreifen, an die Nachrichten in der DLQ weitergeleitet werden, damit der DLQ-Handler ausgeführt werden kann. Sie müssen auch berechtigt sein, die Identität anderer Benutzer zu übernehmen, damit die DLQ Nachrichten in Warteschlangen mit der Berechtigung der Benutzer-ID im Nachrichtenkontext einreicht.

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

 *Regeltabelle des DLQ-Handlers unter IBM i*

Die Regeltabelle des DLQ-Handlers definiert, wie der DLQ-Handler Nachrichten verarbeitet, die in der DLQ von IBM i ankommen.

Die Regeltabelle des DLQ-Handlers definiert, wie der DLQ-Handler Nachrichten verarbeiten soll, die in der DLQ ankommen. Es gibt zwei Typen von Einträgen in einer Regeltabelle:

- Der erste Eintrag in der Tabelle, der optional ist, enthält *Steuerdaten*.
- Alle anderen Einträge in der Tabelle sind *Regeln*, damit der DLQ-Handler folgen kann. Jede Regel besteht aus einem *Muster* (eine Gruppe von Nachrichtenmerkmalen), mit denen eine Nachricht abgeglichen wird, und eine *Aktion*, die ausgeführt werden soll, wenn eine Nachricht in der DLQ mit dem

angegebenen Muster übereinstimmt. Es muss mindestens eine Regel in einer Regeltabelle vorhanden sein.

Jeder Eintrag in der Regeltabelle enthält ein oder mehrere Schlüsselwörter.

Steuerdaten

In diesem Abschnitt werden die Schlüsselwörter beschrieben, die Sie in einen Steuerdateneintrag in eine DLQ-Handler-Regeltabelle aufnehmen können. Beachten Sie Folgendes:

- Der Standardwert für ein Schlüsselwort (falls vorhanden) ist unterstrichen.
- Die vertikale Linie (|) trennt Alternativen. Sie können nur eine dieser Angaben angeben.
- Alle Schlüsselwörter sind optional.

INPUTQ (*QueueName* | ')

Der Name der DLQ, die verarbeitet werden soll:

1. Jeder Wert für UDLMMSGQ (oder *DFT), den Sie als Parameter für den Befehl **STRMQMDLQ** angeben, überschreibt alle INPUTQ-Werte in der Regeltabelle.
2. Wenn Sie einen leeren UDLMMSGQ-Wert als Parameter für den Befehl **STRMQMDLQ** angeben, wird der INPUTQ-Wert in der Regeltabelle verwendet.
3. Wird ein leerer UDLMMSGQ-Wert als Parameter für den Befehl **STRMQMDLQ** und ein leerer Wert für INPUTQ in der Regeltabelle angegeben, wird die standardmäßige Warteschlange für nicht zustellbare Nachrichten des Systems verwendet.

INPUTQM (*QueueManagerName* | ')

Der Name des Warteschlangenmanagers, der Eigner der in dem Schlüsselwort INPUTQ genannten DLQ ist.

Wenn Sie keinen Warteschlangenmanager angeben oder INPUTQM (') in der Regeltabelle angeben, verwendet das System den Standardwarteschlangenmanager für die Installation.

RETRYINT (*Interval* | 60)

Das Intervall (in Sekunden), in dem der DLQ-Handler versuchen soll, Nachrichten in der DLQ erneut zu verarbeiten, die beim ersten Versuch nicht verarbeitet werden konnten und für die wiederholte Versuche angefordert wurden. Das Wiederholungsintervall beträgt standardmäßig 60 Sekunden.

WAIT (**YES | **NO** | *nnn*)**

Gibt an, ob der DLQ-Handler warten soll, bis weitere Nachrichten in der DLQ ankommen, wenn er feststellt, dass keine weiteren Nachrichten mehr verarbeitet werden können.

JA

Bewirkt, dass der DLQ-Handler unbegrenzt wartet.

NO

Bewirkt, dass der DLQ-Handler beendet wird, wenn er feststellt, dass der DLQ entweder leer ist oder keine Nachrichten enthält, die er verarbeiten kann.

nnn

Bewirkt, dass der DLQ-Handler auf *nnn* Sekunden wartet, bevor er neue Arbeit eintrifft, bevor er beendet wird, nachdem er festgestellt hat, dass die Warteschlange leer ist oder keine Nachrichten enthält, die er verarbeiten kann.

Geben Sie WAIT (YES) für ausgelastet-DLQs und WAIT (NO) oder WAIT (*nnn*) für DLQs an, die über eine niedrige Aktivität verfügen. Wenn der DLQ-Handler beendet werden darf, rufen Sie ihn unter Verwendung von Triggern erneut auf.

Sie können den Namen der DLQ als Eingabeparameter für den Befehl **STRMQMDLQ** als Alternative zum Einschluss von Steuerdaten in die Regeltabelle angeben. Wenn ein beliebiger Wert sowohl in der Regeltabelle als auch in der Eingabe für den Befehl **STRMQMDLQ** angegeben wird, hat der im Befehl **STRMQMDLQ** angegebene Wert Vorrang.

Anmerkung: Wenn ein Steuerdateneintrag in der Regeltabelle enthalten ist, muss er der erste Eintrag in der Tabelle sein.

IBM i DLQ-Regeln (Muster und Aktionen) unter IBM i

Eine Beschreibung der Muster und Aktionen für jede der Regeln der Warteschlange für nicht zustellbare Nachrichten unter IBM i.

Nachfolgend ist eine Beispielregel aus einer Regeltabelle für DLQ-Handler-Regeln:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Diese Regel weist den DLQ-Handler an, dass 3 versucht, eine beliebige persistente Nachricht an die Zielwarteschlange zu übergeben, die in den DLQ gestellt wurde, da MQPUT und MQPUT1 unterdrückt wurden.

In diesem Abschnitt werden die Schlüsselwörter beschrieben, die Sie in eine Regel aufnehmen können. Dabei ist Folgendes zu beachten:

- Der Standardwert für ein Schlüsselwort (falls vorhanden) ist unterstrichen. Bei den meisten Schlüsselwörtern ist der Standardwert * (Stern), der mit einem beliebigen Wert übereinstimmt.
- Die vertikale Linie (|) trennt Alternativen. Sie können nur eine dieser Angaben angeben.
- Alle Schlüsselwörter mit Ausnahme von ACTION sind optional.

Dieser Abschnitt beginnt mit einer Beschreibung der Schlüsselwörter für die Mustererkennung (diejenigen, für die Nachrichten in der DLQ abgeglichen werden sollen). Anschließend werden die Aktionsschlüsselwörter beschrieben (die Schlüsselwörter, die bestimmen, wie der DLQ-Handler eine übereinstimmende Nachricht verarbeiten soll).

IBM i DLQ-Musterabgleichschlüsselwörter unter IBM i

Die Musterabgleichschlüsselwörter sind in einem Beispiel beschrieben. Verwenden Sie diese Schlüsselwörter, um Werte anzugeben, mit denen Nachrichten in der IBM i-Warteschlange für nicht zustellbare Nachrichten abgeglichen werden. Alle Schlüsselwörter für die Mustererkennung sind optional.

APPLIDAT (*ApplIdentityData* | *)

Der *ApplIdentityData*-Wert der Nachricht in der DLQ, die im Nachrichtendeskriptor (MQMD) angegeben ist.

APPLNAME (*PutAppName* | *)

Der Name der Anwendung, die den MQPUT- oder MQPUT1-Aufruf ausgegeben hat, wie im Feld *PutAppName* des Nachrichtendeskriptors, MQMD, der Nachricht in der DLQ angegeben ist.

APPLTYPE (*PutApplType* | *)

Der *PutApplType*-Wert, der im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

DESTQ (*QueueName* | *)

Der Name der Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

DESTQM (*QueueManagerName* | *)

Der Name des Warteschlangenmanagers für die Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

FEEDBACK (*Feedback* | *)

Wenn der *MsgType*-Wert MQMT_REPORT ist, beschreibt *Feedback* die Art des Berichts.

Sie können symbolische Namen verwenden. Sie können beispielsweise den symbolischen Namen MQFB_COA verwenden, um die Nachrichten in der DLQ zu identifizieren, die eine Bestätigung ihrer Ankunft in ihren Zielwarteschlangen erfordern.

FORMAT (*Format* | *)

Der Name, den der Absender der Nachricht verwendet, um das Format der Nachrichtendaten zu beschreiben.

MSGTYPE (*MsgType* | *)

Der Nachrichtentyp der Nachricht in der DLQ.

Sie können symbolische Namen verwenden. Sie können beispielsweise den symbolischen Namen MQMT_REQUEST verwenden, um die Nachrichten in der DLQ zu identifizieren, die Antworten erfordern.

PERSIST (Persistence | *)

Der Persistenzwert der Nachricht. (Die Persistenz einer Nachricht bestimmt, ob sie die Neustarts des Warteschlangenmanagers überlebt.)

Sie können symbolische Namen verwenden. Sie können z. B. den symbolischen Namen MQPER_PERSISTENT verwenden, um die Nachrichten in der DLQ zu identifizieren, die persistent sind.

REASON (ReasonCode | *)

Der Ursachencode, der beschreibt, warum die Nachricht in die DLQ gestellt wurde.

Sie können symbolische Namen verwenden. Sie können beispielsweise den symbolischen Namen MQRC_Q_FULL verwenden, um die Nachrichten zu identifizieren, die in die Warteschlange für nicht zugehenden_Nachrichten gestellt wurden, weil ihre Zielwarteschlangen voll waren.

REPLYQ (QueueName | *)

Der Name der Antwortwarteschlange, die im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

REPLYQM (QueueManagerName | *)

Der Name des Warteschlangenmanagers für die Warteschlange für Antwortwarteschlangen, die im Schlüsselwort REPLYQ angegeben ist.

USERID (UserIdentifier | *)

Die Benutzer-ID des Benutzers, der die Nachricht in der DLQ erstellt hat, wie im Nachrichtendeskriptor MQMD angegeben.

IBM i DLQ-Aktionsschlüsselwörter unter IBM i

Verwenden Sie diese Aktionsschlüsselwörter für die Warteschlange für nicht zustellbare Nachrichten, um festzulegen, wie eine übereinstimmende Nachricht in der IBM i-Warteschlange für nicht zustellbare Nachrichten verarbeitet wird.

ACTION (DISCARD | IGNORE | RETRY | FWD)

Die Aktion, die für jede Nachricht in der DLQ ausgeführt wird, die mit dem Muster übereinstimmt, das in dieser Regel definiert ist.

KARTE

Bewirkt, dass die Nachricht aus der Warteschlange für DLQ gelöscht wird.

IGNORE

Bewirkt, dass die Nachricht in der Warteschlange für nicht zustellbare Nachrichten bleibt.

WIEDERHO

Bewirkt, dass der DLQ-Handler erneut versucht, die Nachricht in die Zielwarteschlange zu stellen.

FWD

Bewirkt, dass der DLQ-Handler die Nachricht an die Warteschlange weiterleitet, die im Schlüsselwort FWDQ angegeben ist.

Sie müssen das Schlüsselwort ACTION angeben. Die Anzahl der Versuche, eine Aktion zu implementieren, wird durch das Schlüsselwort RETRY gesteuert. Das Schlüsselwort RETRYINT in den Steuerdaten steuert das Intervall zwischen den Versuchen.

FWDQ (Warteschlangenname | & DESTQ | & REPLYQ)

Der Name der Nachrichtenwarteschlange, an die die Nachricht weitergeleitet wird, wenn Sie das Schlüsselwort ACTION auswählen.

Warteschlangenname

Der Name einer Nachrichtenwarteschlange. FWDQ (") ist nicht gültig.

& DESTQ

Geben Sie den Warteschlangennamen aus dem Feld *DestQName* in der MQDLH-Struktur ein.

& REPLYQ

Geben Sie den Warteschlangennamen aus dem Feld *ReplyToQ* im Nachrichtendeskriptor MQMD ein.

Sie können REPLYQ (? *) in dem Nachrichtenmuster angeben, um Fehlernachrichten zu vermeiden, wenn eine Regel, die FWDQ (& REPLYQ) angibt, mit einer Nachricht mit einem leeren *ReplyToQ*-Feld übereinstimmt.

FWDQM (*QueueManagerName* | & DESTQM | & REPLYQM | ' ')

Der Warteschlangenmanager der Warteschlange, an die eine Nachricht weitergeleitet wird.

QueueManagerName

Der Warteschlangenmanagername für die Warteschlange, an die die Nachricht weitergeleitet wird, wenn Sie das Schlüsselwort ACTION (FWD) auswählen.

& DESTQM

Geben Sie den Namen des WS-Managers aus dem Feld *DestQMGrName* in der MQDLH-Struktur ein.

& REPLYQM

Geben Sie den Namen des WS-Managers aus dem Feld *ReplyToQMGr* im Nachrichtendeskriptor MQMD ein.

..

FWDQM (''), der der Standardwert ist, gibt den lokalen Warteschlangenmanager an.

HEADER (YES | NO)

Gibt an, ob der MQDLH in einer Nachricht verbleiben soll, für die ACTION (FWD) angefordert wird. Standardmäßig bleibt der MQDLH in der Nachricht. Das Schlüsselwort HEADER ist für andere Aktionen als FWD nicht gültig.

PUTAUT (DEF | CTX)

Die Berechtigung, mit der Nachrichten vom DLQ-Handler gestellt werden sollen:

DEF

Versetzt Nachrichten mit der Berechtigung des DLQ-Handlers selbst.

CTX

Bewirkt, dass die Nachrichten mit der Berechtigung der Benutzer-ID in den Nachrichtenkontext gestellt werden. Wenn Sie PUTAUT (CTX) angeben, müssen Sie berechtigt sein, die Identität der anderen Benutzer zu übernehmen.

WIEDERHOLUNG (*RetryCount* | 1)

Die Anzahl der Versuche im Bereich von 1 bis 999.999.999, eine Aktion auszuführen (in dem im Schlüsselwort RETRYINT der Steuerdaten angegebenen Intervall).

Anmerkung: Die Anzahl der Versuche des DLQ-Handlers, eine bestimmte Regel zu implementieren, ist spezifisch für die aktuelle Instanz des DLQ-Steuerroutrinen; die Anzahl bleibt nicht über Neustarts hinweg bestehen. Wenn Sie den DLQ-Handler erneut starten, wird die Anzahl der Versuche, eine Regel anwenden zu lassen, auf Null zurückgesetzt.

IBM i *DLQ-Regeltabellenkonventionen unter IBM i*

Die Regeltabelle der IBM i-Warteschlange für nicht zustellbare Nachrichten muss bestimmte Konventionen bezüglich ihrer Syntax, ihrer Struktur und ihres Inhalts einhalten.

- Eine Regeltabelle muss mindestens eine Regel enthalten.
- Schlüsselwörter können in beliebiger Reihenfolge vorkommen.
- Ein Schlüsselwort kann nur einmal in eine beliebige Regel eingeschlossen werden.
- Bei Schlüsselwörtern wird die Groß-/Kleinschreibung nicht beachtet
- Ein Schlüsselwort und sein Parameterwert müssen durch mindestens ein Leerzeichen oder Komma von anderen Schlüsselwörtern getrennt werden.

- Eine beliebige Anzahl von Leerzeichen kann am Anfang oder Ende einer Regel und zwischen Schlüsselwörtern, Interpunktionszeichen und Werten auftreten.
- Jede Regel muss in einer neuen Zeile beginnen.
- Für die Portierbarkeit darf die signifikante Länge einer Zeile nicht größer als 72 Zeichen sein.
- Verwenden Sie das Pluszeichen (+) als letztes nicht leeres Zeichen in einer Zeile, um anzuzeigen, dass die Regel aus dem ersten nicht leeren Zeichen in der nächsten Zeile fortgesetzt wird. Verwenden Sie das Minuszeichen (-) als letztes nicht leeres Zeichen in einer Zeile, um anzuzeigen, dass die Regel ab dem Anfang der nächsten Zeile fortgesetzt wird. Fortsetzungszeichen können innerhalb von Schlüsselwörtern und Parametern vorkommen.

Zum Beispiel:

```
APPLNAME(' ABC+
D')
```

führt zu 'ABCD'.

```
APPLNAME(' ABC-
D')
```

ergibt ' ABC D'.

- Kommentarzeilen, die mit einem Stern (*) beginnen, können an einer beliebigen Position in der Regeltabelle vorkommen.
- Leerzeilen werden ignoriert.
- Jeder Eintrag in der Tabelle der DLQ-Handler-Regeln enthält ein oder mehrere Schlüsselwörter und die zugehörigen Parameter. Die Parameter müssen den folgenden Syntaxregeln entsprechen:
 - Jeder Parameterwert muss mindestens ein signifikantes Zeichen enthalten. Die Anführungszeichen in Anführungszeichen, die in Anführungszeichen eingeschlossen sind, werden nicht als bedeutsam betrachtet. Diese Parameter sind z. B. gültig:

FORMAT(' ABC ')	3 signifikante Zeichen
FORMAT(ABC)	3 signifikante Zeichen
FORMAT(' A ')	1 signifikantes Zeichen
FORMAT(A)	1 signifikantes Zeichen
FORMAT(' ')	1 signifikantes Zeichen

Diese Parameter sind ungültig, da sie keine signifikanten Zeichen enthalten:

```
FORMAT(' ' )
FORMAT( )
FORMAT()
FORMAT
```

- Platzhalterzeichen werden unterstützt. Sie können das Fragezeichen (?) anstelle jedes einzelnen Zeichens verwenden, mit Ausnahme einer abschließenden Leerzeichen. Sie können den Stern (*) an Stelle von null oder mehr benachbarten Zeichen verwenden. Der Stern (*) und das Fragezeichen (?) werden als **Immer** in Parameterwerten als Platzhalterzeichen interpretiert.
- Sie dürfen keine Platzhalterzeichen in die Parameter der Schlüsselwörter ACTION, HEADER, RETRY, FWDQ, FWDQM und PUTAUT aufnehmen.
- Das Abbilden von Leerzeichen in Parameterwerten und in den entsprechenden Feldern in der Nachricht in der Warteschlange für DLQ ist bei der Ausführung von Platzhalterzeichen nicht von Bedeu-

tung. Führende und eingebettete Leerzeichen innerhalb von Zeichenfolgen in Anführungszeichen sind jedoch von Bedeutung für Platzhalterzeichen.

- Numerische Parameter dürfen das Fragezeichen (?) nicht als Platzhalterzeichen enthalten. Sie können den Stern (*) an Stelle eines gesamten numerischen Parameters einfügen, aber der Stern kann nicht als Teil eines numerischen Parameters angegeben werden. Dies sind z. B. gültige numerische Parameter:

MSGTYPE (2) Es können nur Antwortnachrichten ausgewählt werden.

MSGTYPE (*) Beliebiger Nachrichtentyp ist auswählbar

MSGTYPE ('*') Beliebiger Nachrichtentyp ist auswählbar

MSGTYPE ('2*') ist jedoch ungültig, da es einen Stern (*) als Teil eines numerischen Parameters enthält.

- Numerische Parameter müssen im Bereich von 0 bis 999 999 999 liegen. Wenn der Parameterwert in diesem Bereich liegt, wird er akzeptiert, auch wenn er derzeit in dem Feld, auf das sich das Schlüsselwort bezieht, nicht gültig ist. Sie können symbolische Namen für numerische Parameter verwenden.
- Ist ein Zeichenfolgewart kürzer als das Feld in dem MQDLH oder MQMD, auf das sich das Schlüsselwort bezieht, wird der Wert mit Leerzeichen auf die Länge des Felds aufgefüllt. Wenn der Wert, mit Ausnahme von Sternen, länger als das Feld ist, wird ein Fehler diagnostiziert. Dies sind z. B. alle gültigen Zeichenfolgewartwerte für ein 8-Zeichen-Feld:

'ABCDEFGH' 8 Zeichen

'A*C*E*G*I' 5 Zeichen mit Ausnahme von Sternen

'*A*C*E*G*I*K*M*O*' 8 Zeichen mit Ausnahme von Sternen

- Zeichenfolgen, die Leerzeichen, Kleinbuchstaben oder andere Sonderzeichen als Punkt (.), Schrägstrich (?), Unterstreichungszeichen (_) und Prozentzeichen (%) enthalten, müssen in einfache Anführungszeichen eingeschlossen werden. Kleinbuchstaben, die nicht in Anführungszeichen eingeschlossen sind, werden in Großbuchstaben gefaltet. Wenn die Zeichenfolge ein Anführungszeichen enthält, müssen zwei einfache Anführungszeichen verwendet werden, um sowohl den Anfang als auch das Ende des Angebots zu kennzeichnen. Wenn die Länge der Zeichenfolge berechnet wird, wird jedes Vorkommen von doppelten Anführungszeichen als ein einzelnes Zeichen gezählt.

IBM i *Verarbeitung der DLQ-Regeltabelle unter IBM i*

Der DLQ-Handler (Steuerroutine der Warteschlange für nicht zustellbare Nachrichten) durchsucht die Regeltabelle nach einer Regel mit einem Muster, das mit einer Nachricht in der IBM i-Warteschlange für nicht zustellbare Nachrichten übereinstimmt.

Die Suche beginnt mit der ersten Regel in der Tabelle und wird sequenziell durch die Tabelle fortgesetzt. Wenn eine Regel mit einem übereinstimmenden Muster gefunden wird, versucht die Regeltabelle, die Aktion von dieser Regel zu versuchen. Der DLQ-Handler erhöht den Wiederholungszähler für eine Regel immer um 1, wenn er versucht, diese Regel anzuwenden. Schlägt der erste Versuch fehl, wird der Versuch wiederholt, bis die Anzahl der Versuche mit der im Schlüsselwort RETRY angegebenen Anzahl übereinstimmt. Wenn alle Versuche fehlschlagen, sucht der DLQ-Handler nach der nächsten übereinstimmenden Regel in der Tabelle.

Dieser Prozess wird für nachfolgende übereinstimmende Regeln wiederholt, bis eine Aktion erfolgreich ausgeführt wird. Wenn für jede übereinstimmende Regel versucht wurde, die Anzahl der im Schlüsselwort RETRY angegebenen Zeiten zu verwenden, und alle Versuche fehlgeschlagen sind, wird ACTION (IGNORE) angenommen. ACTION (IGNORE) wird auch angenommen, wenn keine übereinstimmende Regel gefunden wird.

Anmerkung:

1. Übereinstimmende Regelmuster werden nur für Nachrichten in der DLQ gesucht, die mit einem MQDLH beginnen. Nachrichten, die nicht mit einem MQDLH beginnen, werden in regelmäßigen Abständen als fehlerhaft gemeldet und verbleiben auf unbegrenzte Zeit auf der DLQ.
2. Alle Musterschlüsselwörter können standardmäßig verwendet werden, so dass eine Regel nur aus einer Aktion bestehen kann. Beachten Sie jedoch, dass nur Regeln für Aktionen auf alle Nachrichten in der Warteschlange angewendet werden, die über MQDLHs verfügen und die noch nicht gemäß anderen Regeln in der Tabelle verarbeitet worden sind.
3. Die Regeltabelle wird geprüft, wenn der DLQ-Handler gestartet wird, und Fehler, die zu diesem Zeitpunkt markiert wurden. Eine Beschreibung der vom DLQ-Handler ausgegebenen Fehlnachrichten finden Sie im Handbuch *Nachrichten und Ursachencodes*. Sie können jederzeit Änderungen an der Regeltabelle vornehmen, diese Änderungen treten jedoch erst in Kraft, wenn der DLQ-Handler erneut gestartet wird.
4. Der DLQ-Handler ändert den Inhalt von Nachrichten, des MQDLH oder des Nachrichtendeskriptors nicht. Der DLQ-Handler reiht Nachrichten immer mit der Nachrichtenoption MQPMO_PASS_ALL_CONTEXT in andere Warteschlangen ein.
5. Aufeinanderfolgende Syntaxfehler in der Regeltabelle werden möglicherweise nicht erkannt, da die Prüfung der Regeltabelle die Generierung von sich wiederholenden Fehlern verhindert.
6. Der DLQ-Handler öffnet die DLQ mit der Option MQOO_INPUT_AS_Q_DEF.
7. Es können mehrere Instanzen des DLQ-Handlers gleichzeitig mit derselben Warteschlange ausgeführt werden, wobei dieselbe Regeltabelle verwendet wird. Es ist jedoch eher üblich, dass es eine Eins-zu-eins-Beziehung zwischen einer DLQ und einem DLQ-Handler gibt.

IBM i

Sicherstellen, dass alle DLQ-Nachrichten unter IBM i verarbeitet werden

Der DLQ-Handler führt ein Protokoll aller Nachrichten in der IBM i-DLQ, die zwar angezeigt, aber nicht entfernt wurden.

Wenn Sie den DLQ-Handler als Filter verwenden, um eine kleine Untergruppe der Nachrichten aus der DLQ zu extrahieren, speichert der DLQ-Handler weiterhin einen Datensatz dieser Nachrichten in der DLQ, die er nicht verarbeitet hat. Außerdem kann der DLQ-Handler nicht garantieren, dass neue Nachrichten, die in der DLQ ankommen, angezeigt werden, selbst wenn die DLQ als First-in-First-Out (FIFO) definiert ist. Wenn die Warteschlange nicht leer ist, wird die DLQ in regelmäßigen Abständen erneut durchsucht, um alle Nachrichten zu überprüfen.

Versuchen Sie aus diesen Gründen, sicherzustellen, dass die DLQ so wenige Nachrichten wie möglich enthält. Wenn Nachrichten, die nicht gelöscht oder an andere Warteschlangen weitergeleitet werden können (aus welchem Grund auch immer), in der Warteschlange angesammelt werden können, erhöht sich die Auslastung der DLQ-Steueroutine und die DLQ selbst ist in Gefahr, die Warteschlange zu füllen.

Sie können bestimmte Maßnahmen ergreifen, um den DLQ-Handler zu aktivieren, um die DLQ zu leeren. Versuchen Sie z. B. nicht, ACTION (IGNORE) zu verwenden, wodurch Nachrichten in der Warteschlange für nicht zulassene Nachrichten hinterlassen werden. (Denken Sie daran, dass ACTION (IGNORE) für Nachrichten angenommen wird, die nicht explizit durch andere Regeln in der Tabelle adressiert werden.) Verwenden Sie stattdessen für die Nachrichten, die Sie ansonsten ignorieren würden, eine Aktion, mit der die Nachrichten in eine andere Warteschlange verschoben werden. Beispiel:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Außerdem müssen Sie die letzte Regel in der Tabelle zu einem Catchall-Prozess machen, um Nachrichten zu verarbeiten, die nicht von früheren Regeln in der Tabelle adressiert wurden. Die letzte Regel in der Tabelle könnte z. B. wie folgt aussehen:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Dies bewirkt, dass Nachrichten, die bis zur letzten Regel in der Tabelle durchlaufen werden, an die Warteschlange REALLY . DEAD . QUEUE weitergeleitet werden, wo sie manuell verarbeitet werden können.

Wenn Sie nicht über eine solche Regel verfügen, bleiben die Nachrichten wahrscheinlich unbegrenzt auf der DLQ.

IBM i *Beispiel einer Regeltabelle des DLQ-Handlers unter IBM i*

Beispielcode für eine Regeltabelle der Steuerroutine der Warteschlange für nicht zustellbare Nachrichten (DLQ-Handler) unter IBM i. Diese Beispielregeltabelle enthält einen einzigen Steuerdateneintrag und mehrere Regeln.

```
*****
*   An example rules table for the STRMQMDLQ command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* STRMQMDLQ, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to STRMQMDLQ,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation is always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never does things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it must be able
* to cope with the message being lost, so we can afford to
* discard the message.

PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)

* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
```



```
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We do not have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

DLQ-Handler aufrufen

Rufen Sie den Handler für dead-Mail-Warteschlangen mit dem Befehl `runmqdlq` auf. Sie können die DLQ, die Sie verarbeiten möchten, und den Warteschlangenmanager, den Sie auf zwei Arten verwenden möchten, benennen.

Es gibt zwei Möglichkeiten, wie folgt:

- Geben Sie als Parameter in der Eingabeaufforderung `runmqdlq` an. Zum Beispiel:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrule.rul
```

- In der Regeltabelle. Zum Beispiel:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Die Beispiele gelten für die DLQ mit dem Namen `ABC1.DEAD.LETTER.QUEUE`, deren Eigner der WS-Manager `ABC1.QUEUE.MANAGER` ist.

Wenn Sie die DLQ oder den Warteschlangenmanager nicht wie gezeigt angeben, wird der Standardwarteschlangenmanager für die Installation zusammen mit der DLQ, die zu diesem Warteschlangenmanager gehört, verwendet.

Der Befehl `runmqdlq` übernimmt seine Eingabe aus `stdin`. Sie ordnen die Regeltabelle `runmqdlq` zu, indem Sie `stdin` aus der Regeltabelle umleiten.

Um den DLQ-Handler ausführen zu können, müssen Sie berechtigt sein, sowohl auf die DLQ selbst als auch auf alle Nachrichtwarteschlangen zuzugreifen, an die Nachrichten in der DLQ weitergeleitet werden. Damit der DLQ-Handler Nachrichten in Warteschlangen mit der Berechtigung der Benutzer-ID im Nachrichtenkontext einreicht, müssen Sie auch berechtigt sein, die Identität anderer Benutzer zu übernehmen.

Weitere Informationen zum Befehl `runmqdlq` finden Sie in [runmqdlq](#).

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

*Der Beispiel-DLQ-Handler **amqsdlq***

Zusätzlich zu dem mit dem Befehl `runmqdlq` aufgerufenen DLQ-Handler stellt IBM MQ die Quelle eines Beispiel-DLQ-Handlers namens `amqsdlq` mit einer Funktion zur Verfügung, die der von `runmqdlq` bereitgestellten Funktion ähnelt.

Sie können `amqsdlq` anpassen, um einen DLQ-Handler bereitzustellen, der Ihre Anforderungen erfüllt. Sie können beispielsweise entscheiden, dass ein DLQ-Handler Nachrichten ohne Header von nicht zustellbaren Nachrichten verarbeiten kann. (Sowohl der Standard-DLQ-Handler als auch der Mustercode 'amqsdlq' verarbeiten nur die Nachrichten in der Warteschlange für nicht zustellbare Nachrichten, die mit einem

Header von nicht zustellbaren Nachrichten (MQDLH) beginnen. Nachrichten, die nicht mit einem MQDLH beginnen, werden als fehlerhaft erkannt und verbleiben unbegrenzt auf der DLQ.)

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

In IBM MQ for Windows wird die Quelle von `amqsdlq` in folgendem Verzeichnis bereitgestellt:

```
MQ_INSTALLATION_PATH\tools\c\samples\d1q
```

und die kompilierte Version wird im folgenden Verzeichnis bereitgestellt:

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

Auf IBM MQ for UNIX- und Linux-Systemen wird die Quelle von `amqsdlq` in folgendem Verzeichnis bereitgestellt:

```
MQ_INSTALLATION_PATH/samp/d1q
```

und die kompilierte Version wird im folgenden Verzeichnis bereitgestellt:

```
MQ_INSTALLATION_PATH/samp/bin
```

Sie können **amqsdlq** auch im Clientmodus kompilieren. Weitere Informationen finden Sie in den Abschnitten zum Schreiben von Clientprozeduranwendungen, Erstellen von Anwendungen für IBM MQ MQI clients und [Anwendungen in der IBM MQ MQI client-Umgebung ausführen](#).

Die Regeltabelle für den DLQ-Handler

Die Tabelle mit den Regeln für den Handler für dead-Mail-Warteschlangen definiert, wie der DLQ-Handler Nachrichten verarbeitet, die in der DLQ ankommen.

Es gibt zwei Typen von Einträgen in einer Regeltabelle:

- Der erste Eintrag in der Tabelle, der optional ist, enthält *Steuerdaten*.
- Alle anderen Einträge in der Tabelle sind *Regeln*, damit der DLQ-Handler folgen kann. Jede Regel besteht aus einem *Muster* (eine Gruppe von Nachrichtenmerkmalen), mit denen eine Nachricht abgeglichen wird, und eine *Aktion*, die ausgeführt werden soll, wenn eine Nachricht in der DLQ mit dem angegebenen Muster übereinstimmt. Es muss mindestens eine Regel in einer Regeltabelle vorhanden sein.

Jeder Eintrag in der Regeltabelle enthält ein oder mehrere Schlüsselwörter.

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

DLQ-Steuerdaten

Sie können Schlüsselwörter in einen Steuerdateneintrag (control-data) in eine Regeltabelle für einen DLQ-Handler (Steuerroutine der Warteschlange für nicht zustellbare Nachrichten) aufnehmen.

Anmerkung:

- Die vertikale Linie (|) trennt Alternativen, von denen nur eine angegeben werden kann.
- Alle Schlüsselwörter sind optional.

INPUTQ (*QueueName* | ' ')

Der Name der DLQ, die verarbeitet werden soll:

1. Jeder INPUTQ-Wert, den Sie als Parameter für den Befehl `runmqdlq` angeben, überschreibt alle INPUTQ-Werte in der Regeltabelle.
2. Wenn Sie keinen INPUTQ-Wert als Parameter für den Befehl `runmqdlq` angeben, aber Sie **do** einen Wert in der Regeltabelle angeben, wird der INPUTQ-Wert in der Regeltabelle verwendet.

3. Wenn keine DLQ angegeben ist oder Sie INPUTQ (') in der Regeltabelle angeben, wird der Name des DLQ, der zum Warteschlangenmanager gehört, mit dem Namen, der als Parameter für den Befehl `runmqdlq` angegeben wird, verwendet.
4. Wenn Sie keinen INPUTQ-Wert als Parameter für den Befehl `runmqdlq` oder als Wert in der Regeltabelle angeben, wird die DLQ verwendet, die zum Warteschlangenmanager gehört, der in der Regeltabelle INPUTQM in der Regeltabelle angegeben ist.

INPUTQM (*QueueManagerName* | ')

Der Name des Warteschlangenmanagers, der Eigner der im Schlüsselwort INPUTQ genannten DLQ ist:

1. Jeder INPUTQM-Wert, den Sie als Parameter für den Befehl `runmqdlq` angeben, überschreibt alle INPUTQM-Werte in der Regeltabelle.
2. Wenn Sie keinen INPUTQM-Wert als Parameter für den Befehl `runmqdlq` angeben, wird der INPUTQM-Wert in der Regeltabelle verwendet.
3. Wenn kein Warteschlangenmanager angegeben ist oder Sie INPUTQM (') in der Regeltabelle angeben, wird der Standardwarteschlangenmanager für die Installation verwendet.

RETRYINT (*Interval* | 60)

Das Intervall (in Sekunden), in dem der DLQ-Handler Nachrichten in der DLQ erneut verarbeiten soll, die beim ersten Versuch nicht verarbeitet werden konnten und für die wiederholte Versuche angefordert wurden. Das Wiederholungsintervall beträgt standardmäßig 60 Sekunden.

WAIT (YES | NO | *nnn*)

Gibt an, ob der DLQ-Handler warten soll, bis weitere Nachrichten in der DLQ ankommen, wenn er feststellt, dass keine weiteren Nachrichten mehr verarbeitet werden können.

YES

Der DLQ-Handler wartet unbegrenzt.

Nein

Der DLQ-Handler wird beendet, wenn er feststellt, dass der DLQ entweder leer ist oder keine Nachrichten enthält, die er verarbeiten kann.

nnn

Der DLQ-Handler wartet auf *nnn* Sekunden für neue Arbeit, bevor er beendet wird, nachdem er festgestellt hat, dass die Warteschlange entweder leer ist oder keine Nachrichten enthält, die er verarbeiten kann.

Geben Sie WAIT (YES) für ausgelastet-DLQs und WAIT (NO) oder WAIT (*nnn*) für DLQs an, die über eine niedrige Aktivität verfügen. Wenn der DLQ-Handler beendet werden darf, rufen Sie ihn erneut auf, und verwenden Sie die Triggerung. Weitere Informationen zum Auslösen finden Sie im Abschnitt [IBM MQ-Anwendungen mit Triggern starten](#).

Eine Alternative zum Einschluss von Steuerdaten in die Regeltabelle besteht darin, die Namen der DLQ und des zugehörigen Warteschlangenmanagers als Eingabeparameter an den Befehl `runmqdlq` zu liefern. Wenn Sie einen Wert in der Regeltabelle angeben und als Eingabe für den Befehl `runmqdlq` angegeben wird, hat der im Befehl `runmqdlq` angegebene Wert Vorrang.

Wenn Sie einen Steuerdateneintrag in die Regeltabelle aufnehmen, muss er der **erste** Eintrag in der Tabelle sein.

DLQ-Regeln (Muster und Aktionen)

Eine Beschreibung der Schlüsselwörter für die Mustererkennung (die Schlüsselwörter, mit denen Nachrichten in der Warteschlange für dead-letter übereinstimmen), und die Aktionsschlüsselwörter (die Schlüsselwörter, die bestimmen, wie der DLQ-Handler eine übereinstimmende Nachricht verarbeiten soll). Es wird auch eine Beispielregel bereitgestellt.

Die Schlüsselwörter für die Mustererkennung

Die Schlüsselwörter für die Mustererkennung, die Sie verwenden, um Werte anzugeben, mit denen Nachrichten in der DLQ übereinstimmen, lauten wie folgt. (Alle Schlüsselwörter für die Mustererkennung sind optional):

APPLIDAT (*ApplIdentityData* | *)

Der *ApplIdentityData* -Wert, der im Nachrichtendeskriptor MQMD der Nachricht in der Warteschlange für nicht zuorden_Nachrichten angegeben ist.

APPLNAME (*PutApplName* | *)

Der Name der Anwendung, die den Aufruf MQPUT oder MQPUT1 ausgegeben hat, wie im Feld *PutApplName* des Nachrichtendeskriptors, MQMD, der Nachricht in der DLQ angegeben ist.

APPLTYPE (*PutApplType* | *)

Der *PutApplType* -Wert, der im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

DESTQ (*QueueName* | *)

Der Name der Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

DESTQM (*QueueManagerName* | *)

Der Name des Warteschlangenmanagers für die Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

FEEDBACK (*Feedback* | *)

Wenn der Wert *MsgType* den Wert MQFB_REPORT hat, beschreibt *Feedback* die Art des Berichts.

Sie können symbolische Namen verwenden. Sie können beispielsweise den symbolischen Namen MQFB_COA verwenden, um die Nachrichten in der DLQ zu identifizieren, die eine Bestätigung über ihre Ankunft in ihren Zielwarteschlangen benötigen.

FORMAT (*Format* | *)

Der Name, den der Absender der Nachricht verwendet, um das Format der Nachrichtendaten zu beschreiben.

MSGTYPE (*MsgType* | *)

Der Nachrichtentyp der Nachricht in der DLQ.

Sie können symbolische Namen verwenden. Sie können beispielsweise den symbolischen Namen MQMT_REQUEST verwenden, um die Nachrichten in der DLQ zu identifizieren, die Antworten benötigen.

PERSIST (*Persistence* | *)

Der Persistenzwert der Nachricht. (Die Persistenz einer Nachricht bestimmt, ob sie die Neustarts des Warteschlangenmanagers überlebt.)

Sie können symbolische Namen verwenden. Sie können z. B. den symbolischen Namen MQPER_PERSISTENT verwenden, um Nachrichten in der DLQ zu identifizieren, die persistent sind.

REASON (*ReasonCode* | *)

Der Ursachencode, der beschreibt, warum die Nachricht in die DLQ gestellt wurde.

Sie können symbolische Namen verwenden. Sie können beispielsweise den symbolischen Namen MQRC_Q_FULL verwenden, um die Nachrichten zu identifizieren, die in die Warteschlange für nicht zugehenden_Nachrichten gestellt wurden, weil ihre Zielwarteschlangen voll waren.

REPLYQ (*QueueName* | *)

Der Name der Warteschlange für Antwortnachrichten, die im Nachrichtendeskriptor MQMD der Nachricht in der Warteschlange für DLQ angegeben ist.

REPLYQM (*QueueManagerName* | *)

Der Name des Warteschlangenmanagers der Warteschlange für Antwortnachrichten, wie er im Nachrichtendeskriptor MQMD der Nachricht in der Warteschlange für die DLQ angegeben ist.

USERID (*UserIdentifier* | *)

Die Benutzer-ID des Benutzers, der die Nachricht in der Warteschlange für nicht zuorden_Nachrichten erstellt hat, wie im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

Die Aktionsschlüsselwörter

Die Aktionsschlüsselwörter, die verwendet werden, um zu beschreiben, wie eine übereinstimmende Nachricht verarbeitet werden soll, lauten wie folgt:

ACTION (DISCARD | IGNORE | RETRY | FWD)

Die Aktion, die für alle Nachrichten in der DLQ ausgeführt werden soll, die mit dem in dieser Regel definierten Muster übereinstimmen.

KARTE

Löschen Sie die Nachricht aus der DLQ.

IGNORE

Hinterlassen Sie die Nachricht in der DLQ.

WIEDERHO

Wenn der erste Versuch, die Nachricht in die Zielwarteschlange zu stellen, fehlschlägt, versuchen Sie es erneut. Mit dem Schlüsselwort RETRY wird die Anzahl der Versuche festgelegt, eine Aktion zu implementieren. Das Schlüsselwort RETRYINT in den Steuerdaten steuert das Intervall zwischen den Versuchen.

FWD

Die Nachricht an die Warteschlange weiterleiten, die im Schlüsselwort FWDQ angegeben ist.

Sie müssen das Schlüsselwort ACTION angeben.

FWDQ (*Warteschlangename* | & DESTQ | & REPLYQ)

Der Name der Nachrichtenwarteschlange, an die die Nachricht weitergeleitet werden soll, wenn ACTION (FWD) angefordert wird.

Warteschlangename

Der Name einer Nachrichtenwarteschlange. FWDQ (") ist nicht gültig.

& DESTQ

Geben Sie den Warteschlangennamen aus dem Feld *DestQName* in der MQDLH-Struktur ein.

& REPLYQ

Geben Sie den Warteschlangennamen aus dem Feld *ReplyToQ* im Nachrichtendeskriptor MQMD ein.

Um Fehlernachrichten zu vermeiden, wenn eine Regel, die FWDQ (&REPLYQ) angibt, mit einer Nachricht mit einem leeren *ReplyToQ* -Feld übereinstimmt, geben Sie REPLYQ (?*) in dem Nachrichtenmuster an.

FWDQM (*QueueManagerName* | & DESTQM | & REPLYQM | ' ')

Der WS-Manager der Warteschlange, in die eine Nachricht weitergeleitet werden soll.

QueueManagerName

Der Name des Warteschlangenmanagers für die Warteschlange, an die eine Nachricht weitergeleitet werden soll, wenn ACTION (FWD) angefordert wird.

& DESTQM

Geben Sie den Namen des WS-Managers aus dem Feld *DestQMgrName* in der MQDLH-Struktur ein.

& REPLYQM

Geben Sie den Namen des WS-Managers aus dem Feld *ReplyToQMgr* im Nachrichtendeskriptor MQMD ein.

..

FWDQM ("), der der Standardwert ist, gibt den lokalen Warteschlangenmanager an.

HEADER (YES | NO)

Gibt an, ob der MQDLH in einer Nachricht verbleiben soll, für die ACTION (FWD) angefordert wird. Standardmäßig bleibt der MQDLH in der Nachricht. Das Schlüsselwort HEADER ist für andere Aktionen als FWD nicht gültig.

PUTAUT (DEF | CTX)

Die Berechtigung, mit der Nachrichten vom DLQ-Handler gestellt werden sollen:

DEF

Stellen Sie Nachrichten mit der Berechtigung des DLQ-Handlers selbst ein.

CTX

Stellen Sie die Nachrichten mit der Berechtigung der Benutzer-ID in den Nachrichtenkontext. Wenn Sie PUTAUT (CTX) angeben, müssen Sie berechtigt sein, die Identität der anderen Benutzer zu übernehmen.

WIEDERHOLUNG (*RetryCount* | 1)

Die Anzahl der Male im Bereich von 1 bis 999.999.999, um eine Aktion zu versuchen (in dem im Schlüsselwort RETRYINT der Steuerdaten angegebenen Intervall). Die Anzahl der Versuche des DLQ-Handlers, eine bestimmte Regel zu implementieren, ist spezifisch für die aktuelle Instanz des DLQ-Steuerroutrinen; die Anzahl bleibt nicht über Neustarts hinweg bestehen. Wenn der DLQ-Handler erneut gestartet wird, wird die Anzahl der Versuche, eine Regel anwenden zu lassen, auf Null zurückgesetzt.

Beispielregel

Nachfolgend ist eine Beispielregel aus einer Regeltabelle für DLQ-Handler-Regeln:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Diese Regel weist den DLQ-Handler an, drei Versuche zu unternehmen, alle persistenten Nachrichten, die in die Warteschlange für den DLQ gestellt wurden, an die Zielwarteschlange zu übergeben, da MQPUT und MQPUT1 unterdrückt wurden.

Alle Schlüsselwörter, die Sie in einer Regel verwenden können, werden im restlichen Teil dieses Abschnitts beschrieben. Beachten Sie Folgendes:

- Der Standardwert für ein Schlüsselwort (falls vorhanden) ist unterstrichen. Bei den meisten Schlüsselwörtern ist der Standardwert * (Stern), der mit einem beliebigen Wert übereinstimmt.
- Die vertikale Linie (|) trennt Alternativen, von denen nur eine angegeben werden kann.
- Alle Schlüsselwörter mit Ausnahme von ACTION sind optional.

DLQ-Regeltabellenkonventionen

Die Syntax, die Struktur und der Inhalt der Regeltabelle für den DLQ-Handler müssen diesen Konventionen entsprechen.

Die Regeltabelle muss die folgenden Konventionen einhalten:

- Eine Regeltabelle muss mindestens eine Regel enthalten.
- Schlüsselwörter können in beliebiger Reihenfolge vorkommen.
- Ein Schlüsselwort kann in jeder Regel nur einmal enthalten sein.
- Bei Schlüsselwörtern wird die Groß-/Kleinschreibung nicht beachtet.
- Ein Schlüsselwort und sein Parameterwert müssen durch mindestens ein Leerzeichen oder Komma von anderen Schlüsselwörtern getrennt werden.
- Es kann eine beliebige Anzahl von Leerzeichen am Anfang oder Ende einer Regel sowie zwischen Schlüsselwörtern, Interpunktionszeichen und Werten geben.
- Jede Regel muss in einer neuen Zeile beginnen.
- Auf Windows-Systemen muss die letzte Regel in der Tabelle mit einem Wagenrücklauf-/Zeilenvorschubzeichen enden. Sie können dies erreichen, indem Sie sicherstellen, dass Sie die Eingabetaste am Ende der Regel drücken, damit die letzte Zeile der Tabelle eine Leerzeile ist.
- Aus Gründen der Portierbarkeit darf die signifikante Länge einer Zeile nicht größer als 72 Zeichen sein.
- Verwenden Sie das Pluszeichen (+) als letztes nicht leeres Zeichen in einer Zeile, um anzugeben, dass die Regel aus dem ersten nicht leeren Zeichen in der nächsten Zeile fortgesetzt wird. Verwenden Sie das Minuszeichen (-) als letztes nicht leeres Zeichen in einer Zeile, um anzuzeigen, dass die Regel ab dem Anfang der nächsten Zeile fortgesetzt wird. Fortsetzungszeichen können innerhalb von Schlüsselwörtern und Parametern vorkommen.

Zum Beispiel:

```
APPLNAME('ABC+
D')
```

führt zu 'ABCD' und

```
APPLNAME('ABC-
D')
```

ergibt 'ABC D'.

- Kommentarzeilen, die mit einem Stern (*) beginnen, können an einer beliebigen Position in der Regeltabelle vorkommen.
- Leerzeilen werden ignoriert.
- Jeder Eintrag in der Tabelle der DLQ-Handler-Regeln enthält ein oder mehrere Schlüsselwörter und die zugehörigen Parameter. Die Parameter müssen den folgenden Syntaxregeln entsprechen:
 - Jeder Parameterwert muss mindestens ein signifikantes Zeichen enthalten. Die Begrenzungszeichen in Werten, die in Anführungszeichen eingeschlossen sind, werden nicht als bedeutsam betrachtet. Diese Parameter sind z. B. gültig:

FORMAT('ABC')	3 signifikante Zeichen
FORMAT(ABC)	3 signifikante Zeichen
FORMAT('A')	1 signifikantes Zeichen
FORMAT(A)	1 signifikantes Zeichen
FORMAT('')	1 signifikantes Zeichen

Diese Parameter sind ungültig, da sie keine signifikanten Zeichen enthalten:

```
FORMAT('')
FORMAT( )
FORMAT()
FORMAT
```

- Platzhalterzeichen werden unterstützt. Sie können das Fragezeichen (?) anstelle eines einzelnen Zeichens verwenden, mit Ausnahme einer abschließenden Leerzeichen; Sie können den Stern (*) anstelle von null oder mehr benachbarte Zeichen verwenden. Der Stern (*) und das Fragezeichen (?) werden als **Immer** in Parameterwerten als Platzhalterzeichen interpretiert.
- Platzhalterzeichen können nicht in die Parameter für die Schlüsselwörter ACTION, HEADER, RETRY, FWDQ, FWDQM und PUTAUT aufgenommen werden.
- Das Abbilden von Leerzeichen in Parameterwerten und in den entsprechenden Feldern in der Nachricht in der Warteschlange für DLQ ist bei der Ausführung von Platzhalterzeichen nicht von Bedeutung. Führende und eingebettete Leerzeichen in Zeichenfolgen, die in einfache Anführungszeichen eingeschlossen sind, sind jedoch von Bedeutung für Platzhalterzeichen.
- Numerische Parameter dürfen das Fragezeichen (?) nicht als Platzhalterzeichen enthalten. Sie können den Stern (*) anstelle eines gesamten numerischen Parameters verwenden, jedoch nicht als Teil eines numerischen Parameters. Dies sind z. B. gültige numerische Parameter:

MSGTYPE(2)	Es können nur Antwortnachrichten ausgewählt werden.
MSGTYPE(*)	Beliebiger Nachrichtentyp ist auswählbar
MSGTYPE('*')	Beliebiger Nachrichtentyp ist auswählbar

MSGTYPE (' 2*') ist jedoch ungültig, da es einen Stern (*) als Teil eines numerischen Parameters enthält.

- Numerische Parameter müssen im Bereich von 0 bis 999 999 999 liegen. Wenn der Parameterwert in diesem Bereich liegt, wird er akzeptiert, auch wenn er derzeit in dem Feld, auf das sich das Schlüsselwort bezieht, nicht gültig ist. Sie können symbolische Namen für numerische Parameter verwenden.
- Ist ein Zeichenfolgewart kürzer als das Feld in dem MQDLH oder MQMD, auf das sich das Schlüsselwort bezieht, wird der Wert mit Leerzeichen auf die Länge des Felds aufgefüllt. Wenn der Wert, mit Ausnahme von Sternen, länger als das Feld ist, wird ein Fehler diagnostiziert. Dies sind z. B. alle gültigen Zeichenfolgewart für ein 8 Zeichen-Feld:

' ABCDEFGH '	8 Zeichen
' A*C*E*G*I '	5 Zeichen mit Ausnahme von Sternen
' *A*C*E*G*I*K*M*O * '	8 Zeichen mit Ausnahme von Sternen

- Schließen Sie Zeichenfolgen ein, die Leerzeichen, Kleinbuchstaben oder andere Sonderzeichen als Punkt (.), Schrägstrich (/), Unterstreichungszeichen (_) und Prozentzeichen (%) in einfache Anführungszeichen enthalten. Kleinbuchstaben, die nicht in Hochkommas eingeschlossen sind, werden in Großbuchstaben umgelegt. Wenn die Zeichenfolge ein Angebot enthält, verwenden Sie zwei einfache Anführungszeichen, um sowohl den Anfang als auch das Ende des Angebots zu kennzeichnen. Wenn die Länge der Zeichenfolge berechnet wird, wird jedes Vorkommen von doppelten Anführungszeichen als ein einzelnes Zeichen gezählt.

Verarbeitung der DLQ-Regeltabelle

Der Warteschlangen-Handler für dead-Mail durchsucht die Regeltabelle nach einer Regel, bei der das Muster mit einer Nachricht in der Warteschlange für nicht zustellbare Nachrichten übereinstimmt.

Die Suche beginnt mit der ersten Regel in der Tabelle und wird sequenziell durch die Tabelle fortgesetzt. Wenn der DLQ-Handler eine Regel mit einem übereinstimmenden Muster findet, führt er die Aktion von dieser Regel aus. Der DLQ-Handler erhöht den Wiederholungszähler für eine Regel immer dann um 1, wenn er diese Regel anwendet. Wenn der erste Versuch fehlschlägt, versucht der DLQ-Handler erneut, bis die Anzahl der Versuche mit der im Schlüsselwort RETRY angegebenen Anzahl übereinstimmt. Wenn alle Versuche fehlschlagen, sucht der DLQ-Handler nach der nächsten übereinstimmenden Regel in der Tabelle.

Dieser Prozess wird für nachfolgende übereinstimmende Regeln wiederholt, bis eine Aktion erfolgreich ausgeführt wird. Wenn für jede übereinstimmende Regel versucht wurde, die Anzahl der im Schlüsselwort RETRY angegebenen Zeiten zu verwenden, und alle Versuche fehlgeschlagen sind, wird ACTION (IGNORE) angenommen. ACTION (IGNORE) wird auch angenommen, wenn keine übereinstimmende Regel gefunden wird.

Anmerkung:

1. Übereinstimmende Regelmuster werden nur für Nachrichten in der DLQ gesucht, die mit einem MQDLH beginnen. Nachrichten, die nicht mit einem MQDLH beginnen, werden in regelmäßigen Abständen als fehlerhaft gemeldet und verbleiben auf unbegrenzte Zeit auf der DLQ.
2. Alle Musterschlüsselwörter können in der Standardeinstellung verwendet werden, so dass eine Regel nur aus einer Aktion bestehen kann. Beachten Sie jedoch, dass nur Regeln für Aktionen auf alle Nachrichten in der Warteschlange angewendet werden, die über MQDLHs verfügen und die noch nicht gemäß anderen Regeln in der Tabelle verarbeitet worden sind.
3. Die Regeltabelle wird geprüft, wenn der DLQ-Handler gestartet wird, und die Fehler werden zu diesem Zeitpunkt markiert. Sie können jederzeit Änderungen an der Regeltabelle vornehmen, diese Änderungen treten jedoch erst in Kraft, wenn der DLQ-Handler erneut gestartet wird.
4. Der DLQ-Handler ändert den Inhalt von Nachrichten, den MQDLH oder den Nachrichtendeskriptor nicht. Der DLQ-Handler reiht Nachrichten immer mit der Nachrichtenoption MQPMO_PASS_ALL_CON-TEXT in andere Warteschlangen ein.

5. Aufeinanderfolgende Syntaxfehler in der Regeltabelle werden möglicherweise nicht erkannt, da die Regeltabelle so konzipiert ist, dass die Generierung von Wiederholungsfehlern während der Prüfung nicht mehr generiert wird.
6. Der DLQ-Handler öffnet die DLQ mit der Option MQOO_INPUT_AS_Q_DEF.
7. Es können mehrere Instanzen des DLQ-Handlers gleichzeitig mit derselben Warteschlange ausgeführt werden, wobei dieselbe Regeltabelle verwendet wird. Es ist jedoch eher üblich, dass es eine Eins-zu-eins-Beziehung zwischen einer DLQ und einem DLQ-Handler gibt.

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

Sicherstellen, dass alle DLQ-Nachrichten verarbeitet werden

Der Handler für dead-Mail-Warteschlangen protokolliert alle Nachrichten in der DLQ, die zwar angezeigt, aber nicht entfernt wurden.

Wenn Sie den DLQ-Handler als Filter verwenden, um eine kleine Untergruppe der Nachrichten aus der DLQ zu extrahieren, muss der DLQ-Handler weiterhin einen Satz dieser Nachrichten in der DLQ aufbewahren, die er nicht verarbeitet hat. Außerdem kann der DLQ-Handler nicht garantieren, dass neue Nachrichten, die in der DLQ ankommen, angezeigt werden, selbst wenn die DLQ als First-in-First-out (FIFO) definiert ist. Wenn die Warteschlange nicht leer ist, wird die DLQ in regelmäßigen Abständen erneut durchsucht, um alle Nachrichten zu überprüfen.

Aus diesen Gründen sollte sichergestellt werden, dass die DLQ so wenige Nachrichten wie möglich enthält. Wenn Nachrichten, die nicht gelöscht oder an andere Warteschlangen weitergeleitet werden können (aus welchem Grund auch immer), in der Warteschlange angesammelt werden können, erhöht sich die Auslastung des DLQ-Handlers und die DLQ selbst kann sich füllen.

Sie können bestimmte Maßnahmen ergreifen, um den DLQ-Handler zu aktivieren, um die DLQ zu leeren. Versuchen Sie z. B. nicht, ACTION (IGNORE) zu verwenden, wodurch Nachrichten in der Warteschlange für nicht zulassene Nachrichten hinterlassen werden. (Denken Sie daran, dass ACTION (IGNORE) für Nachrichten angenommen wird, die nicht explizit durch andere Regeln in der Tabelle adressiert werden.) Verwenden Sie stattdessen für die Nachrichten, die Sie ansonsten ignorieren würden, eine Aktion, mit der die Nachrichten in eine andere Warteschlange verschoben werden. Zum Beispiel:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Außerdem müssen Sie die letzte Regel in der Tabelle zu einem Catchall-Prozess machen, um Nachrichten zu verarbeiten, die nicht von früheren Regeln in der Tabelle adressiert wurden. Die letzte Regel in der Tabelle könnte z. B. wie folgt aussehen:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Dadurch werden Nachrichten, die bis zur letzten Regel in der Tabelle durchgehen, an die Warteschlange REALLY . DEAD . QUEUE weitergeleitet, wo sie manuell verarbeitet werden können. Wenn Sie nicht über eine solche Regel verfügen, bleiben die Nachrichten wahrscheinlich unbegrenzt auf der DLQ.

Beispieltabelle für DLQ-Handler-Regeln

Eine Beispieltabelle für eine Warteschlange für nicht zustellbare Nachrichten für den Befehl `runmqdlq` mit einem einzigen Steuerdateneintrag und mehreren Regeln.

```
*****
*   An example rules table for the runmqdlq command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
```

```

*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

```

```

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.

REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)

```

Zugehörige Informationen

[Warteschlangen für nicht zustellbare Nachrichten](#)

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

[runmqdlq \(Steueroutine der Warteschlange für nicht zustellbare Nachrichten ausführen\)](#)

Mit Modellwarteschlangen arbeiten

Ein Warteschlangenmanager erstellt eine *dynamische Warteschlange*, wenn er einen MQI-Aufruf von einer Anwendung empfängt, die einen Warteschlangennamen angibt, der als Modellwarteschlange definiert wurde. Der Name der neuen dynamischen Warteschlange wird vom WS-Manager beim Erstellen der Warteschlange generiert. Eine *Modellwarteschlange* ist eine Schablone, die die Attribute aller dynamischen Warteschlangen angibt, die aus dieser Warteschlange erstellt wurden. Modellwarteschlangen stellen eine komfortable Methode für Anwendungen zur Verfügung, um Warteschlangen nach Bedarf zu erstellen.

Modellwarteschlange definieren

Sie verwenden den Befehl **DEFINE QMODEL**, um eine Modellwarteschlange mit einer Gruppe von Attributen in derselben Weise zu definieren, wie Sie eine lokale Warteschlange definieren. Modellwarteschlangen und lokale Warteschlangen verfügen über dieselbe Gruppe von Attributen, mit der Ausnahme, dass Sie in Modellwarteschlangen angeben können, ob die erstellten dynamischen Warteschlangen temporär oder permanent sind. (Permanente Warteschlangen werden über WS-Manager-Neustarts gepflegt, temporäre werden nicht ausgeführt.) Beispiel:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
DESCR('Queue for messages from application X') +
PUT (DISABLED) +
GET (ENABLED) +
NOTRIGGER +
MSGDLVSQ (FIFO) +
MAXDEPTH (1000) +
MAXMSGL (2000) +
USAGE (NORMAL) +
DEFTYPE (PERMDYN)
```

Mit diesem Befehl wird eine Modellwarteschlangendefinition erstellt. Über das Attribut **DEFTYPE** können Sie sehen, dass die tatsächlichen Warteschlangen, die aus dieser Vorlage erstellt wurden, permanente dynamische Warteschlangen sind. Alle Attribute, die nicht angegeben sind, werden automatisch aus der Standardwarteschlange `SYSYSTEM.DEFAULT.MODEL.QUEUE` kopiert.

Sie können die Attribute **LIKE** und **REPLACE** bei der Definition von Modellwarteschlangen auf dieselbe Weise verwenden, wie Sie sie mit lokalen Warteschlangen verwenden.

Andere Befehle mit Modellwarteschlangen verwenden

Sie können die entsprechenden MQSC-Befehle verwenden, um die Attribute einer Modellwarteschlange anzuzeigen oder zu ändern oder um das Modellwarteschlangenobjekt zu löschen. Beispiel:

Verwenden Sie den Befehl **DISPLAY QUEUE**, um die Attribute der Modellwarteschlange anzuzeigen:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Verwenden Sie den Befehl **ALTER QMODEL**, um das Modell so zu ändern, dass es in jede dynamische Warteschlange, die aus diesem Modell erstellt wird, versetzt wird:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Verwenden Sie den Befehl **DELETE QMODEL**, um diese Modellwarteschlange zu löschen:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

Zugehörige Informationen

[ALTER QMODEL](#)

[QMODEL DEFINIER](#)

[DELETE QMODEL](#)

Mit Verwaltungsthemen arbeiten

Verwenden Sie MQSC-Befehle, um Verwaltungsthemen zu verwalten.

Detaillierte Informationen zu diesen Befehlen finden Sie in den [MQSC-Befehlen](#) .

Zugehörige Konzepte

„[Verwaltungsthema definieren](#)“ auf Seite 180

Verwenden Sie den MQSC-Befehl **DEFINE TOPIC** , um ein Verwaltungsthema zu erstellen. Wenn Sie ein Verwaltungsthema definieren, können Sie optional jedes Themenattribut festlegen.

„[Attribute des Verwaltungsthemenobjekts anzeigen](#)“ auf Seite 181

Verwenden Sie den MQSC-Befehl **DISPLAY TOPIC** , um ein Verwaltungsthemenobjekt anzuzeigen.

„[Verwaltungsthemenattribute ändern](#)“ auf Seite 181

Sie können die Themenattribute auf zwei Arten ändern, indem Sie entweder den Befehl **ALTER TOPIC** oder den Befehl **DEFINE TOPIC** mit dem Attribut **REPLACE** verwenden.

„[Eine Verwaltungsthemendefinition kopieren](#)“ auf Seite 182

Sie können eine Themendefinition mit dem Attribut **LIKE** im Befehl **DEFINE** kopieren.

„[Löschen einer Verwaltungsthemendefinition](#)“ auf Seite 182

Sie können den MQSC-Befehl **DELETE TOPIC** verwenden, um ein Verwaltungsthema zu löschen.

Zugehörige Informationen

[Verwaltungsthemenobjekte](#)

Verwaltungsthema definieren

Verwenden Sie den MQSC-Befehl **DEFINE TOPIC** , um ein Verwaltungsthema zu erstellen. Wenn Sie ein Verwaltungsthema definieren, können Sie optional jedes Themenattribut festlegen.

Jedes Attribut des Themas, das nicht explizit festgelegt wurde, wird aus dem Standardverwaltungsthema **SYSTEM.DEFAULT.TOPIC** übernommen, das bei der Installation der Systeminstallation erstellt wurde.

Der folgende Befehl **DEFINE TOPIC** definiert beispielsweise ein Thema mit dem Namen **ORANGE . TOPIC** mit den folgenden Merkmalen:

- Löst den Wert für die Themenzeichenfolge **ORANGE** auf. Informationen dazu, wie Themenzeichenfolgen verwendet werden können, finden Sie unter [Themenzeichenfolgen kombinieren](#).
- Jedes Attribut, das auf "ASPARENT" gesetzt ist, verwendet das Attribut, wie es durch das übergeordnete Thema dieses Themas definiert ist. Diese Aktion wird die Themenstruktur so weit wiederholt, bis das Stammthema **SYSTEM.BASE.TOPIC** gefunden wird. Weitere Informationen finden Sie unter [Themenbäume](#).

```
DEFINE TOPIC (ORANGE.TOPIC) +  
TOPICSTR (ORANGE) +  
DEFPRTY (ASPARENT) +  
NPMSGDLV (ASPARENT)
```

Anmerkung:

- Mit Ausnahme des Werts für die Themenzeichenfolge sind alle angezeigten Attributwerte die Standardwerte. Sie sind hier nur als Illustration dargestellt. Sie können sie übergehen, wenn Sie sicher sind, dass die Standardwerte die gewünschten Werte sind oder nicht geändert wurden. Siehe auch „[Attribute des Verwaltungsthemenobjekts anzeigen](#)“ auf Seite 181.
- Wenn Sie bereits ein Verwaltungsthema auf demselben WS-Manager mit dem Namen **ORANGE.TOPIC** haben, schlägt dieser Befehl fehl. Verwenden Sie das Attribut **REPLACE**, wenn die vorhandene Definition eines Themas überschrieben werden soll. Lesen Sie dazu auch „[Verwaltungsthemenattribute ändern](#)“ auf Seite 181.

Zugehörige Informationen

[TOPIC DEFINI](#)

Attribute des Verwaltungsthemenobjekts anzeigen

Verwenden Sie den MQSC-Befehl **DISPLAY TOPIC** , um ein Verwaltungsthemenobjekt anzuzeigen.

Um alle Themen anzuzeigen, verwenden Sie:

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

Sie können Attribute selektiv anzeigen, indem Sie sie einzeln mit dem Befehl **DISPLAY TOPIC** angeben. Beispiel:

```
DISPLAY TOPIC(ORANGE.TOPIC) +  
TOPICSTR +  
DEFPRTY +  
NPMSGDLV
```

Mit diesem Befehl werden die drei angegebenen Attribute wie folgt angezeigt:

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC)                                TYPE(LOCAL)  
TOPICSTR(ORANGE)                                    DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

Verwenden Sie den Befehl **DISPLAY TPSTATUS** , um die ASPARENT-Topic-Werte anzuzeigen, während sie in der Laufzeit verwendet werden. Verwenden Sie zum Beispiel:

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

Der Befehl zeigt die folgenden Details an:

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE)                                DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

Wenn Sie ein Verwaltungsthema definieren, werden alle Attribute, die Sie nicht explizit angeben, aus dem Standardverwaltungsthema, das als SYSTEM.DEFAULT.TOPIC. bezeichnet wird, verwendet. Verwenden Sie den folgenden Befehl, um zu sehen, welche Standardattribute verwendet werden:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

Zugehörige Informationen

[ANZEIGEN TOPIC](#)

[ANZEIGEN TPSTATUS](#)

Verwaltungsthemenattribute ändern

Sie können die Themenattribute auf zwei Arten ändern, indem Sie entweder den Befehl **ALTER TOPIC** oder den Befehl **DEFINE TOPIC** mit dem Attribut **REPLACE** verwenden.

Wenn Sie beispielsweise die Standardpriorität von Nachrichten ändern möchten, die an ein Thema mit dem Namen ORANGE.TOPIC übergeben werden, müssen Sie einen der folgenden Befehle verwenden.

- Mit dem Befehl **ALTER** :

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Dieser Befehl ändert ein einzelnes Attribut, das der Standardpriorität der Nachricht, die diesem Thema zugestellt wird, auf 5; alle anderen Attribute bleiben unverändert.

- Mit dem Befehl **DEFINE** :

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Mit diesem Befehl wird die Standardpriorität von Nachrichten geändert, die zu diesem Thema bereitgestellt werden. Alle anderen Attribute werden mit ihren Standardwerten angegeben.

Wenn Sie die Priorität von Nachrichten ändern, die an dieses Thema gesendet werden, sind die vorhandenen Nachrichten nicht betroffen. Jede neue Nachricht verwendet jedoch die angegebene Priorität, wenn sie nicht von der Veröffentlichungsanwendung bereitgestellt wird.

Zugehörige Informationen

[ALTER TOPIC](#)

[ANZEIGEN TOPIC](#)

Eine Verwaltungsthemendefinition kopieren

Sie können eine Themendefinition mit dem Attribut **LIKE** im Befehl **DEFINE** kopieren.

Beispiel:

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Dieser Befehl erstellt ein Thema, MAGENTA.TOPIC, mit denselben Attributen wie das ursprüngliche Thema ORANGE.TOPIC und nicht die Attribute des Standardverwaltungsthemas für das System. Geben Sie den Namen des Themas ein, das genau so kopiert werden soll, wie es bei der Erstellung des Themas eingegeben wurde. Wenn der Name Kleinbuchstaben enthält, schließen Sie den Namen in einfache Anführungszeichen ein.

Sie können diese Form des Befehls **DEFINE** auch verwenden, um eine Themendefinition zu kopieren, aber Änderungen an den Attributen des Originals vorzunehmen. Beispiel:

```
DEFINE TOPIC(BLUE.TOPIC) +  
TOPICSTR(BLUE) +  
LIKE(ORANGE.TOPIC)
```

Sie können auch die Attribute des Themas BLUE.TOPIC in das Thema "GREEN.TOPIC" kopieren und angeben, dass, wenn Veröffentlichungen nicht an die richtige Subskribentenwarteschlange zugestellt werden können, sie nicht in die Warteschlange für dead-letter gestellt werden. Beispiel:

```
DEFINE TOPIC(GREEN.TOPIC) +  
TOPICSTR(GREEN) +  
LIKE(BLUE.TOPIC) +  
USEDLQ(NO)
```

Zugehörige Informationen

[TOPIC DEFINI](#)

Löschen einer Verwaltungsthemendefinition

Sie können den MQSC-Befehl **DELETE TOPIC** verwenden, um ein Verwaltungsthema zu löschen.

Beispiel:

```
DELETE TOPIC(ORANGE.TOPIC)
```

Anwendungen können das Thema nicht mehr für die Veröffentlichung öffnen oder neue Subskriptionen unter Verwendung des Objektnamens ORANGE.TOPIC. öffnen. Die Veröffentlichung von Anwendungen, für die das Thema geöffnet ist, kann die Veröffentlichung der aufgelösten Themenzeichenfolge fortsetzen. Alle Subskriptionen, die bereits zu diesem Thema ausgeführt wurden, erhalten weiterhin Veröffentlichungen, nachdem das Thema gelöscht wurde.

Anwendungen, die dieses Themenobjekt nicht referenzieren, sondern die aufgelöste Themenzeichenfolge verwenden, die dieses Themenobjekt (in diesem Beispiel 'ORANGE') darstellt, arbeiten weiter. In diesem Fall übernehmen sie die Eigenschaften aus einem Themenobjekt in der Themenstruktur. Weitere Informationen finden Sie unter [Themenbäume](#).

Zugehörige Informationen

[DELETE TOPIC](#)

Mit Subskriptionen arbeiten

Verwenden Sie MQSC-Befehle zum Verwalten von Subskriptionen.

Subskriptionen können einen von drei Typen haben, die im Attribut **SUBTYPE** definiert sind:

ADMIN

Administrativ definiert durch einen Benutzer.

PROXY

Eine intern erstellte Subskription für Routing-Veröffentlichungen zwischen WS-Managern.

API

Das Programm wurde über das Programm erstellt, z. B. mit dem MQI-Aufruf MQSUB.

Detaillierte Informationen zu diesen Befehlen finden Sie in den [MQSC-Befehlen](#).

Zugehörige Konzepte

[„Verwaltungssubskription definieren“](#) auf Seite 183

Verwenden Sie den MQSC-Befehl **DEFINE SUB**, um eine Verwaltungssubskription zu erstellen. Sie können auch den Standardwert verwenden, der in der standardmäßigen lokalen Subskriptionsdefinition definiert ist. Sie können auch die Subskriptionsmerkmale von denen der lokalen Standardsubskription SYSTEM.DEFAULT.SUB ändern, die bei der Installation des Systems erstellt wurde.

[„Attribute von Subskriptionen anzeigen“](#) auf Seite 184

Mit dem Befehl **DISPLAY SUB** können Sie konfigurierte Attribute einer beliebigen Subskription anzeigen, die dem Warteschlangenmanager bekannt ist.

[„Attribute für lokale Subskription ändern“](#) auf Seite 185

Sie können die Subskriptionsattribute auf zwei Arten ändern, indem Sie entweder den Befehl **ALTER SUB** oder den Befehl **DEFINE SUB** mit dem Attribut **REPLACE** verwenden.

[„Lokale Subskriptionsdefinition kopieren“](#) auf Seite 186

Sie können eine Subskriptionsdefinition mit dem Attribut **LIKE** im Befehl **DEFINE** kopieren.

[„Lokale Subskription löschen“](#) auf Seite 186

Sie können den MQSC-Befehl **DELETE SUB** verwenden, um eine lokale Subskription zu löschen.

Verwaltungssubskription definieren

Verwenden Sie den MQSC-Befehl **DEFINE SUB**, um eine Verwaltungssubskription zu erstellen. Sie können auch den Standardwert verwenden, der in der standardmäßigen lokalen Subskriptionsdefinition definiert ist. Sie können auch die Subskriptionsmerkmale von denen der lokalen Standardsubskription SYSTEM.DEFAULT.SUB ändern, die bei der Installation des Systems erstellt wurde.

Der folgende Befehl **DEFINE SUB** definiert beispielsweise eine Subskription mit dem Namen ORANGE mit den folgenden Merkmalen:

- Dauerhafte Subskription, d. d. sie bleibt beim Neustart des Warteschlangenmanagers bestehen, mit unbegrenztem Ablaufdatum.


```
DISPLAY SUB +
SUBID(414D51204141412020202020202020EE921E4E20002A03) +
TOPICSTR +
DURABLE
```

Mit diesem Befehl wird die gleiche Ausgabe wie zuvor ausgegeben:

```
AMQ8096: IBM MQ subscription inquired.
SUBID(414D51204141412020202020202020EE921E4E20002A03)
SUB(ORANGE)                                TOPICSTR(ORANGE)
DURABLE(YES)
```

Proxy-Subskriptionen in einem Warteschlangenmanager werden standardmäßig nicht angezeigt. Um sie anzuzeigen, geben Sie eine **SUBTYPE** von PROXY oder ALL an.

Sie können den Befehl [DISPLAY SBSTATUS](#) verwenden, um die Laufzeitattribute anzuzeigen. Verwenden Sie z. B. den folgenden Befehl:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGS
```

Die folgende Ausgabe wird angezeigt:

```
AMQ8099: IBM MQ subscription status inquired.
SUB(ORANGE)
SUBID(414D51204141412020202020202020EE921E4E20002A03)
NUMMSGS(0)
```

Wenn Sie eine Verwaltungssubskription definieren, werden alle Attribute, die Sie nicht explizit angeben, aus der Standardsubskription, die als SYSTEM.DEFAULT.SUB bezeichnet wird, verwendet. Verwenden Sie den folgenden Befehl, um zu sehen, welche Standardattribute verwendet werden:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

Zugehörige Informationen

[ANZEIGEN SUB](#)

Attribute für lokale Subskription ändern

Sie können die Subskriptionsattribute auf zwei Arten ändern, indem Sie entweder den Befehl **ALTER SUB** oder den Befehl **DEFINE SUB** mit dem Attribut **REPLACE** verwenden.

Wenn Sie beispielsweise die Priorität von Nachrichten, die an eine Subskription mit dem Namen ORANGE übergeben werden, ändern möchten, verwenden Sie einen der folgenden Befehle:

- Mit dem Befehl **ALTER** :

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Dieser Befehl ändert ein einzelnes Attribut, das der Priorität von Nachrichten, die an diese Subskription zugestellt werden, auf 5; alle anderen Attribute bleiben unverändert.

- Mit dem Befehl **DEFINE** :

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

Dieser Befehl ändert nicht nur die Priorität von Nachrichten, die an diese Subskription übergeben werden, sondern auch alle anderen Attribute, die ihre Standardwerte erhalten.

Wenn Sie die Priorität von Nachrichten ändern, die an diese Subskription gesendet werden, sind die vorhandenen Nachrichten nicht betroffen. Alle neuen Nachrichten haben jedoch die angegebene Priorität.

Zugehörige Informationen

[ALTER SUB](#)
[SUB DEFINI](#)

Lokale Subskriptionsdefinition kopieren

Sie können eine Subskriptionsdefinition mit dem Attribut **LIKE** im Befehl **DEFINE** kopieren.

Beispiel:

```
DEFINE SUB(BLUE) +  
      LIKE(ORANGE)
```

Sie können auch die Attribute der untergeordneten REAL in das untergeordnete Element THIRD.SUB kopieren und angeben, dass die Korrelations-ID der bereitgestellten Veröffentlichungen DIRD ist und nicht die Publisher-CorrelID. Beispiel:

```
DEFINE SUB(THIRD.SUB) +  
      LIKE(BLUE) +  
      DESTCORL(ORANGE)
```

Zugehörige Informationen

[SUB DEFINI](#)

Lokale Subskription löschen

Sie können den MQSC-Befehl **DELETE SUB** verwenden, um eine lokale Subskription zu löschen.

```
DELETE SUB(ORANGE)
```

Sie können eine Subskription auch mit der SUBID löschen:

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

Zugehörige Informationen

[DELETE SUB](#)

Nachrichten in einer Subskription überprüfen

Wenn eine Subskription definiert ist, wird sie einer Warteschlange zugeordnet. Veröffentlichte Nachrichten, die mit dieser Subskription übereinstimmen, werden in diese Warteschlange gestellt.

Informationen zu diesem Vorgang

Beachten Sie, dass die folgenden **runmqsc** -Befehle nur die Subskriptionen anzeigen, die Nachrichten empfangen haben.

Führen Sie die folgenden Schritte aus, um nach Nachrichten zu suchen, die sich derzeit in der Warteschlange für eine Subskription befinden

Vorgehensweise

1. Informationen zum Überprüfen von Nachrichten, die sich in der Warteschlange für einen Subskriptionstyp `DISPLAY SBSTATUS(sub_name) NUMMSGs` befinden, finden Sie im Abschnitt [„Attribute von Subskriptionen anzeigen“](#) auf Seite 184.
2. Wenn der Wert für **NUMMSGs** größer als null ist, identifizieren Sie die Warteschlange, die der Subskription zugeordnet ist, indem Sie `DISPLAY SUB(sub_name) DEST` eingeben.

3. Wenn Sie den Namen der zurückgegebenen Warteschlange verwenden, können Sie die Nachrichten nach dem in „Warteschlangen durchsuchen“ auf Seite 154 beschriebenen Verfahren anzeigen.

Zugehörige Informationen

[ANZEIGEN SBSTATUS](#)

Mit Services arbeiten

Serviceobjekte sind ein Mittel, mit dem zusätzliche Prozesse als Teil eines Warteschlangenmanagers verwaltet werden können. Mit Services können Sie Programme definieren, die gestartet und gestoppt werden, wenn der WS-Manager gestartet und beendet wird. IBM MQ-Services werden immer unter der Benutzer-ID des Benutzers gestartet, der den Warteschlangenmanager gestartet hat.

Verwenden Sie den MQSC-Befehl **DEFINE SERVICE**, um eine neue IBM MQ -Servicedefinition zu definieren.

Serviceobjekte können einen der folgenden Typen haben:

Server

Ein Server ist ein Serviceobjekt, das den Parameter **SERVTYPE** hat, der als **SERVER** angegeben ist. Ein Serverserviceobjekt ist die Definition eines Programms, das ausgeführt wird, wenn ein bestimmter WS-Manager gestartet wird. Serverserviceobjekte definieren Programme, die in der Regel lange ausgeführt werden. Beispielsweise kann ein Server-Service-Objekt verwendet werden, um einen Auslösemonitorprozess wie **runmqtrm** auszuführen.

Es kann nur eine Instanz eines Server-Service-Objekts gleichzeitig ausgeführt werden. Der Status aktiver Serverserviceobjekte kann mit dem MQSC-Befehl **DISPLAY SVSTATUS** überwacht werden.

Befehl

Ein Befehl ist ein Serviceobjekt, für das der Parameter **SERVTYPE** als **COMMAND** angegeben wurde. Befehlsserviceobjekte ähneln Serverserviceobjekten, jedoch können mehrere Instanzen eines Befehlsserviceobjekts gleichzeitig ausgeführt werden und ihr Status kann nicht mit dem MQSC-Befehl **DISPLAY SVSTATUS** überwacht werden.

Wenn der MQSC-Befehl **STOP SERVICE** ausgeführt wird, wird nicht geprüft, ob das vom MQSC-Befehl **START SERVICE** gestartete Programm noch aktiv ist, bevor das Programm gestoppt wird.

Zugehörige Informationen

[ANZEIGEN SVSTATUS](#)

[START SERVICE](#)

[STOP-SERVICE](#)

Serviceobjekt definieren

Sie definieren ein Serviceobjekt mit dem MQSC-Befehl **DEFINE SERVICE** .

Zu den Attributen, die Sie definieren müssen, gehören die folgenden:

SERVTYPE

Definiert den Typ des Serviceobjekts. Folgende Werte sind möglich:

SERVER

Ein Serverserviceobjekt.

Es kann immer nur eine Instanz eines Serverserviceobjekts ausgeführt werden. Der Status der Server-Service-Objekte kann mit dem MQSC-Befehl **DISPLAY SVSTATUS** überwacht werden.

BEFEHL

Ein Befehlsserviceobjekt.

Es können mehrere Instanzen eines Befehlsserviceobjekts gleichzeitig ausgeführt werden. Der Status einer Befehlsserviceobjekte kann nicht überwacht werden.

STARTCMD

Das Programm, das zum Starten des Service ausgeführt wird. Es muss ein vollständig qualifizierter Pfad zum Programm angegeben werden.

STARTARG

Argumente, die an das Startprogramm übergeben werden.

STDERR

Gibt den Pfad zu einer Datei an, in die der Standardfehler (stderr) des Serviceprogramms umgeleitet werden soll.

STDOUT

Gibt den Pfad zu einer Datei an, in die die Standardausgabe (stdout) des Serviceprogramms umgeleitet werden soll.

STOPCMD

Das Programm, das zum Stoppen des Service ausgeführt wird. Es muss ein vollständig qualifizierter Pfad zum Programm angegeben werden.

STOPARG

Argumente, die an das Stoppprogramm übergeben werden.

CONTROL

Gibt an, wie der Service gestartet und gestoppt werden soll:

MANUAL

Der Service wird nicht automatisch gestartet oder automatisch gestoppt. Es wird durch die Verwendung der Befehle **START SERVICE** und **STOP SERVICE** gesteuert. Dies ist der Standardwert.

QMGR

Der Service, der definiert wird, soll gleichzeitig gestartet und gestoppt werden, wenn der Warteschlangenmanager gestartet und gestoppt wird.

STARTONLY

Der Service soll zur gleichen Zeit wie der Warteschlangenmanager gestartet werden, aber er wird nicht zum Stoppen aufgefordert, wenn der Warteschlangenmanager gestoppt wird.

Zugehörige Konzepte

„Services verwalten“ auf Seite 188

Wenn Sie den Parameter **CONTROL** verwenden, kann eine Instanz eines Serviceobjekts entweder automatisch vom Warteschlangenmanager gestartet oder gestoppt werden oder mit den MQSC-Befehlen **START SERVICE** und **STOP SERVICE** gestartet und gestoppt werden.

Zugehörige Informationen

[SERVICE DEFINI](#)

[ANZEIGEN SVSTATUS](#)

[START SERVICE](#)

[STOP-SERVICE](#)

Services verwalten

Wenn Sie den Parameter **CONTROL** verwenden, kann eine Instanz eines Serviceobjekts entweder automatisch vom Warteschlangenmanager gestartet oder gestoppt werden oder mit den MQSC-Befehlen **START SERVICE** und **STOP SERVICE** gestartet und gestoppt werden.

Wenn eine Instanz eines Serviceobjekts gestartet wird, wird eine Nachricht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben, in der der Name des Serviceobjekts und die Prozess-ID des gestarteten Prozesses enthalten sind. Es folgt ein Beispielprotokolleintrag für ein Serverserviceobjekt:

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).
```

EXPLANATION:

```
The Server process has started.  
ACTION:  
None.
```

Es folgt ein Beispielprotokolleintrag für ein Befehlsserviceobjekt:

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)  
Host(HOST_1) Installation(Installation1)  
VRMF(7.1.0.0) QMgr(A.B.C)  
AMQ5030: The Command 'C1' has started. ProcessId(13030).
```

```
EXPLANATION:  
The Command has started.  
ACTION:  
None.
```

Wenn ein Instanzserver-Service gestoppt wird, wird eine Nachricht in die Fehlerprotokolle des Warteschlangenmanagers geschrieben, die den Namen des Service und die Prozess-ID des Endprozesses enthalten. Im Folgenden wird ein Beispielprotokolleintrag für ein Serverserviceobjekt gestoppt:

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)  
Host(HOST_1) Installation(Installation1)  
VRMF(7.1.0.0) QMgr(A.B.C)  
AMQ5029: The Server 'S1' has ended. ProcessId(13031).
```

```
EXPLANATION:  
The Server process has ended.  
ACTION:  
None.
```

Zugehörige Verweise

„Zusätzliche Umgebungsvariablen“ auf Seite 189

Wenn ein Service gestartet wird, wird die Umgebung, in der der Serviceprozess gestartet wird, aus der Umgebung des Warteschlangenmanagers übernommen. Es ist möglich, zusätzliche Umgebungsvariablen zu definieren, die in der Umgebung des Serviceprozesses festgelegt werden sollen, indem die Variablen hinzugefügt werden, die Sie in einer der `service.env`-Umgebungsüberschreibungsdateien definieren wollen.

Zugehörige Informationen

[STOP-SERVICE](#)

[START SERVICE](#)

Zusätzliche Umgebungsvariablen




Wenn ein Service gestartet wird, wird die Umgebung, in der der Serviceprozess gestartet wird, aus der Umgebung des Warteschlangenmanagers übernommen. Es ist möglich, zusätzliche Umgebungsvariablen zu definieren, die in der Umgebung des Serviceprozesses festgelegt werden sollen, indem die Variablen hinzugefügt werden, die Sie in einer der `service.env`-Umgebungsüberschreibungsdateien definieren wollen.

Dateien, denen Sie Umgebungsvariablen hinzufügen können

Es gibt zwei mögliche Dateien, denen Sie Umgebungsvariablen hinzufügen können:




Die Datei mit dem Maschinenbereich `service.env`

Diese Datei befindet sich in:

-   `/var/mqm` auf UNIX and Linux-Systemen.
-  Datenverzeichnis, das während der Installation auf Windows-Systemen ausgewählt wurde.

Die Datei mit dem WS-Manager-Bereich `service.env`

Diese Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers. Beispiel: Die Position der Umgebungsüberschreibungsdatei für einen WS-Manager mit dem Namen QMNAME lautet:

-   Auf Systemen unter UNIX and Linux, `/var/mqm/qmgrs/QMNAME/service.env`
-  Auf Systemen unter Windows, `C:\ProgramData\IBM\MQ\qmgrs\QMNAME\service.env`

Beide Dateien werden, falls verfügbar, verarbeitet, wobei Definitionen in der Bereichsdatei des Warteschlangenmanagers Vorrang vor diesen Definitionen in der Datei des Maschinengeltungsbereichs haben.

Umgebungsvariablen, die in `service.env` angegeben werden können.

Eine Umgebungsvariable kann in `service.env` angegeben werden. Wenn der IBM MQ-Service beispielsweise eine Reihe von Befehlen ausführt, kann es sinnvoll sein, die `PATH`-Benutzervariable in der Datei `service.env` zu setzen. Die Werte, auf die Sie die Variable setzen, können keine Umgebungsvariablen sein. Beispiel: `CLASSPATH= %CLASSPATH%` ist falsch. In ähnlicher Weise würde auf Linux `PATH=$PATH : /opt/mqm/bin` unerwartete Ergebnisse erzielt werden.

`CLASSPATH` muss in Großbuchstaben geschrieben werden, und die Klassenpfadanweisung darf nur Literale enthalten. Einige Services (z. B. Telemetrie) legen ihren eigenen Klassenpfad fest. Der in `service.env` definierte `CLASSPATH` wird hinzugefügt.

Das Format der in der Datei definierten Variablen, `service.env`, ist eine Liste der Paare aus Name und Wert. Jede Variable muss in einer neuen Zeile definiert werden, und jede Variable wird so verwendet, wie sie explizit definiert ist, einschließlich des Leerraums.

Beispiel für `service.env`

```
#####  
##  
## <N_OCO_COPYRIGHT> ##  
## Licensed Materials - Property of IBM ##  
## ##  
## 63H9336 ##  
## (C) Copyright IBM Corporation 2005, 2023. ##  
## ##  
## <NOC_COPYRIGHT> ##  
## ##  
#####  
##* Module Name: service.env ##*  
##* Type : IBM MQ service environment file ##*  
##* Function : Define additional environment variables to be set ##*  
##* for SERVICE programs. ##*  
##* Usage : <VARIABLE>=<VALUE> ##*  
##* ##*  
#####  
MYLOC=/opt/myloc/bin  
MYTMP=/tmp  
TRACEDIR=/tmp/trace  
MYINITQ=ACCOUNTS.INITIATION.QUEUE
```

Zugehörige Verweise

[„Ersetzbare Einfügungen in Servicedefinitionen“ auf Seite 191](#)

In der Definition eines Serviceobjekts ist es möglich, Token zu ersetzen. Ersetzte Token werden automatisch durch ihren erweiterten Text ersetzt, wenn das Serviceprogramm ausgeführt wird. Ersatztoken können aus der folgenden Liste der allgemeinen Token oder aus allen Variablen, die in der Datei `service.env` definiert sind, entnommen werden.

Ersetzbare Einfügungen in Servicedefinitionen

In der Definition eines Serviceobjekts ist es möglich, Token zu ersetzen. Ersetzte Token werden automatisch durch ihren erweiterten Text ersetzt, wenn das Serviceprogramm ausgeführt wird. Ersatztoken können aus der folgenden Liste der allgemeinen Token oder aus allen Variablen, die in der Datei `service.env` definiert sind, entnommen werden.




Im Folgenden finden Sie allgemeine Tokens, die verwendet werden können, um Token in der Definition eines Serviceobjekts zu ersetzen:

MQ_INSTALLATIONSPFAD

Die Position, an der IBM MQ installiert ist.

MQ_DATA_PATH

Die Position des IBM MQ-Datenverzeichnisses:

-   Auf Systemen unter UNIX and Linux ist die IBM MQ-Datenverzeichnisposition `/var/mqm/`.
-  Auf Windows-Systemen ist die Position des Datenverzeichnisses von IBM MQ das Datenverzeichnis, das während der Installation von IBM MQ ausgewählt wurde.

QMNAME

Der Name des aktuellen Warteschlangenmanagers.

MQ_SERVICE_NAME

Der Name des Service.

MQ_SERVER_PID

Dieses Token kann nur von den Argumenten **STOPARG** und **STOPCMD** verwendet werden.

Für Serverserviceobjekte wird dieses Token durch die Prozess-ID des Prozesses ersetzt, der von den Argumenten **STARTCMD** und **STARTARG** gestartet wird. Andernfalls wird dieses Token durch 0 ersetzt.

MQ_Q_MGR_DATA_PATH

Die Position des Datenverzeichnisses des Warteschlangenmanagers.

MQ_Q_MGR_DATA_NAME

Der umgewandelte Name des Warteschlangenmanagers. Weitere Informationen zur Namensumsetzung finden Sie im Abschnitt [Informationen zu IBM MQ-Dateinamen](#).

Wenn Sie austauschbare Einfügungen verwenden möchten, fügen Sie das Token in + -Zeichen in die Zeichenfolgen **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** oder **STDERR** ein. Beispiele hierzu finden Sie im Abschnitt [„Beispiele für die Verwendung von Serviceobjekten“](#) auf Seite 191.

Beispiele für die Verwendung von Serviceobjekten

Die Services in diesem Abschnitt werden, sofern nicht anders angegeben, mit UNIX-Pfadtrennzeichen geschrieben.

Serverserviceobjekt verwenden

In diesem Beispiel wird gezeigt, wie ein Serverserviceobjekt definiert, verwendet und geändert wird, um einen Auslösemonitor zu starten.

1. Ein Server-Serviceobjekt wird mit dem MQSC-Befehl **DEFINE SERVICE** definiert:

```
DEFINE SERVICE(S1) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+bin/runmqtrm') +
STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +
STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +
STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

Dabei gilt Folgendes:

+MQ_INSTALL_PATH+ ist ein Token, das das Installationsverzeichnis darstellt.

+QMNAME+ ist ein Token, das den Namen des Warteschlangenmanagers darstellt.

ACCOUNTS . INITIATION . QUEUE ist die Initialisierungswarteschlange.

amqsstop ist ein mit IBM MQ bereitgestelltes Beispielprogramm, das den Warteschlangenmanager auffordert, alle Verbindungen für die Prozess-ID zu unterbrechen. amqsstop generiert PCF-Befehle, daher muss der Befehlsserver aktiv sein.

+MQ_SERVER_PID+ ist ein Token, das die Prozess-ID darstellt, die an das Stoppprogramm übergeben wurde.

Eine Liste der allgemeinen Tokens finden Sie im Abschnitt [„Ersetzbare Einfügungen in Servicedefinitionen“](#) auf Seite 191.

2. Eine Instanz des Serverserviceobjekts wird ausgeführt, wenn der WS-Manager als Nächstes gestartet wird. Es wird jedoch sofort eine Instanz des Serverserviceobjekts mit dem MQSC-Befehl **START SERVICE** gestartet:

```
START SERVICE(S1)
```

3. Der Status des Serverserviceprozesses wird mit dem MQSC-Befehl **DISPLAY SVSTATUS** angezeigt:

```
DISPLAY SVSTATUS(S1)
```

4. In diesem Beispiel wird jetzt gezeigt, wie das Serverserviceobjekt geändert wird und dass die Aktualisierungen durch einen manuellen Neustart des Serverserviceprozesses übernommen werden. Das Serverserviceobjekt wird so geändert, dass die Initialisierungswarteschlange als JUPITER . INITIATION . QUEUE angegeben wird. Der MQSC-Befehl **ALTER SERVICE** wird verwendet:

```
ALTER SERVICE(S1) +  
STARTARG(' -m +QMNAME+ -q JUPITER . INITIATION . QUEUE ')
```

Anmerkung: Ein laufender Service nimmt keine Aktualisierungen an seiner Servicedefinition an, bis er erneut gestartet wird.

5. Der Serverserviceprozess wird erneut gestartet, damit die Änderung mit den MQSC-Befehlen **STOP SERVICE** und **START SERVICE** übernommen wird:

```
STOP SERVICE(S1)
```

Gefolgt von:

```
START SERVICE(S1)
```

Der Serverserviceprozess wird erneut gestartet und nimmt die in [„4“](#) auf Seite 192 vorgenommenen Änderungen wieder auf.

Anmerkung: Der MQSC-Befehl **STOP SERVICE** kann nur verwendet werden, wenn ein Argument **STOPCMD** in der Servicedefinition angegeben ist.

Zugehörige Informationen

[ALTER SERVICE](#)

[SERVICE DEFINI](#)

[ANZEIGEN SVSTATUS](#)

[START SERVICE](#)

[STOP-SERVICE](#)

Befehlsserviceobjekt verwenden

In diesem Beispiel wird gezeigt, wie ein Befehlsserviceobjekt definiert wird, um ein Programm zu starten, das Einträge in das Systemprotokoll des Betriebssystems schreibt, wenn ein Warteschlangenmanager gestartet oder gestoppt wird.

1. Das Befehlsserviceobjekt wird mit dem MQSC-Befehl **DEFINE SERVICE** definiert:

```
DEFINE SERVICE(S2) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STARTCMD('/usr/bin/logger') +
STARTARG('Queue manager +QMNAME+ starting') +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

Dabei gilt Folgendes:

logger ist der von UNIX and Linux bereitgestellte Befehl zum Schreiben in das Systemprotokoll.
+QMNAME+ ist ein Token, das den Namen des Warteschlangenmanagers darstellt.

Zugehörige Informationen

SERVICE DEFINI

Verwenden eines Befehls-Service-Objekts, wenn ein Warteschlangenmanager nur beendet wird

Dieses Beispiel zeigt, wie ein Befehlsserviceobjekt definiert wird, um ein Programm zu starten, das Einträge in das Systemprotokoll des Betriebssystems schreibt, wenn ein WS-Manager nur gestoppt wird.

1. Das Befehlsserviceobjekt wird mit dem MQSC-Befehl **DEFINE SERVICE** definiert:

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

Dabei gilt Folgendes:

logger ist ein mit IBM MQ bereitgestelltes Beispielprogramm, das Einträge in das Systemprotokoll des Betriebssystems schreiben kann.
+QMNAME+ ist ein Token, das den Namen des Warteschlangenmanagers darstellt.

Zugehörige Informationen

SERVICE DEFINI

Weitere Informationen zum Übergeben von Argum

In diesem Beispiel wird gezeigt, wie ein Serverserviceobjekt definiert wird, um ein Programm mit dem Namen runserv zu starten, wenn ein Warteschlangenmanager gestartet wird.

Dieses Beispiel wird mit Windows-Pfadtrennzeichen geschrieben.

Eines der Argumente, die an das Startprogramm übergeben werden sollen, ist eine Zeichenfolge, die einen Speicherbereich enthält. Dieses Argument muss als einzelne Zeichenfolge übergeben werden. Um dies zu erreichen, werden doppelte Anführungszeichen verwendet, wie im folgenden Befehl gezeigt, um das Befehlsserviceobjekt zu definieren:

1. Das Serverserviceobjekt wird mit dem MQSC-Befehl **DEFINE SERVICE** definiert:

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\ "') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')

DEFINE SERVICE(S4) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\ "') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

Dabei gilt:

+QMNAME+ ist ein Token, das den Namen des Warteschlangenmanagers darstellt.

"C:\Program Files\Tools\" ist eine Zeichenfolge, die einen Speicherbereich enthält, der als einzelne Zeichenfolge übergeben wird.

Zugehörige Informationen

SERVICE DEFINI

Automatisches Starten eines Service

Dieses Beispiel zeigt, wie ein Serverserviceobjekt definiert wird, das zum automatischen Starten des Auslösemonitors beim Starten des Warteschlangenmanagers verwendet werden kann.

1. Das Serverserviceobjekt wird mit dem MQSC-Befehl **DEFINE SERVICE** definiert:

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtrm') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

Dabei gilt:

+QMNAME+ ist ein Token, das den Namen des Warteschlangenmanagers darstellt.

+IQNAME+ ist eine vom Benutzer in einer der service.env-Dateien definierte Umgebungsvariable, die den Namen der Initialisierungswarteschlange darstellt.

Zugehörige Informationen

SERVICE DEFINI

Objekte zum Auslösen verwalten

IBM MQ ermöglicht es, eine Anwendung automatisch zu starten, wenn bestimmte Bedingungen in einer Warteschlange erfüllt sind. Beispiel: Sie möchten eine Anwendung starten, wenn die Anzahl der Nachrichten in einer Warteschlange eine angegebene Zahl erreicht. Diese Funktion wird als *Auslösefunktion* bezeichnet. Sie müssen die Objekte definieren, die das Auslösen unterstützen.

Das Auslösen wird im Abschnitt [IBM MQ-Anwendungen mit Triggern starten](#) ausführlich beschrieben.

Anwendungswarteschlange zum Auslösen definieren

Eine Anwendungswarteschlange ist eine lokale Warteschlange, die von Anwendungen für die Nachrichtenübertragung über die MQI verwendet wird. Für die Triggerung ist es erforderlich, dass eine Reihe von Warteschlangenattributen in der Anwendungswarteschlange definiert wird.

Die Triggerung selbst wird durch das Attribut **Trigger** (TRIGGER in MQSC-Befehlen) aktiviert. In diesem Beispiel soll ein Auslöserereignis generiert werden, wenn 100 Nachrichten mit Priorität 5 oder höher in der lokalen Warteschlange MOTOR.INSURANCE.QUEUE vorhanden sind, wie folgt:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
MAXMSGL (2000) +
DEFPSIST (YES) +
INITQ (MOTOR.INS.INIT.QUEUE) +
TRIGGER +
TRIGTYPE (DEPTH) +
TRIGDPH (100)+
TRIGMPRI (5)
```

Dabei gilt:

QLOCAL (MOTOR.INSURANCE.QUEUE)

Ist der Name der Anwendungswarteschlange, die definiert wird.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

Ist der Name der Prozessdefinition, die die Anwendung definiert, die von einem Auslösemonitorprogramm gestartet werden soll.

MAXMSGL (2000)

Gibt die maximale Länge von Nachrichten in der Warteschlange an.

DEFPSIST (YES)

Gibt an, dass Nachrichten in dieser Warteschlange standardmäßig persistent sind.

INITQ (MOTOR.INS.INIT.QUEUE)

Gibt den Namen der Initialisierungswarteschlange an, in die der Warteschlangenmanager die Auslösenachricht stellen soll.

TRIGGER

Ist der Auslöserattributwert.

TRIGTYPE (DEPTH)

Gibt an, dass ein Auslöserereignis generiert wird, wenn die Anzahl der Nachrichten mit der erforderlichen Priorität (TRIGMPRI) die in TRIGDPTH angegebene Zahl erreicht.

TRIGDPTH (100)

Gibt die Anzahl der Nachrichten an, die zum Generieren eines Auslöserereignisses erforderlich sind.

TRIGMPRI (5)

Ist die Priorität von Nachrichten, die vom Warteschlangenmanager gezählt werden sollen, wenn Sie entscheiden, ob ein Auslöserereignis generiert werden soll. Es werden nur Nachrichten mit Priorität 5 oder höher gezählt.

Initialisierungswarteschlange definieren

Wenn ein Auslöserereignis eintritt, reißt der Warteschlangenmanager eine Auslösenachricht in die Initialisierungswarteschlange ein, die in der Definition der Anwendungswarteschlange angegeben ist. Initialisierungswarteschlangen haben keine speziellen Einstellungen. Sie können jedoch die folgende Definition der lokalen Warteschlange MOTOR.INS.INIT.QUEUE als Anleitung verwenden:

```
DEFINE QLOCAL(MOTOR.INS.INIT.QUEUE) +
GET (ENABLED) +
NOSHARE +
NOTRIGGER +
MAXMSGL (2000) +
MAXDEPTH (1000)
```

Prozess definieren

Verwenden Sie den Befehl DEFINE PROCESS, um eine Prozessdefinition zu erstellen. Eine Prozessdefinition definiert die Anwendung, die für die Verarbeitung von Nachrichten aus der Anwendungswarteschlange verwendet werden soll. Die Anwendungswarteschlangendefinition benennt den zu verwendenden Prozess und ordnet dadurch die Anwendungswarteschlange der Anwendung zu, die für die Verarbeitung ihrer Nachrichten verwendet werden soll. Dies wird über das Attribut PROCESS in der Anwendungswarteschlange MOTOR.INSURANCE.QUOTE.PROCESS ausgeführt. Der folgende MQSC-Befehl definiert den erforderlichen Prozess (MOTOR.INSURANCE.QUOTE.PROCESS), der in diesem Beispiel angegeben ist:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

Dabei gilt Folgendes:

MOTOR.INSURANCE.QUOTE.PROCESS

Ist der Name der Prozessdefinition.

DESCR ('Insurance request message processing')

Beschreibt das Anwendungsprogramm, auf das sich diese Definition bezieht. Dieser Text wird angezeigt, wenn Sie den Befehl DISPLAY PROCESS verwenden. Dadurch können Sie feststellen, was der Prozess tut. Wenn Sie Leerzeichen in der Zeichenfolge verwenden, müssen Sie die Zeichenfolge in einfache Anführungszeichen setzen.

APPLTYPE (UNIX)

Ist der Typ der Anwendung, die gestartet werden soll.

APPLICID ('/u/admin/test/IRMP01')

Ist der Name der ausführbaren Datei der Anwendung, die als vollständig qualifizierter Dateiname angegeben ist. In Windows-Systemen wäre c:\app1\test\irmp01.exe ein typischer APPLICID-Wert.

USERDATA ('open, close, 235')

Benutzerdefinierte Daten, die von der Anwendung verwendet werden können.

Attribute einer Prozessdefinition anzeigen

Verwenden Sie den Befehl DISPLAY PROCESS, um die Ergebnisse Ihrer Definition zu untersuchen. Beispiel:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

Sie können den MQSC-Befehl ALTER PROCESS auch verwenden, um eine vorhandene Prozessdefinition zu ändern, und den Befehl DELETE PROCESS, um eine Prozessdefinition zu löschen.

Verwenden des Dienstprogramms dmpmqmsg zwischen zwei Systemen

Das Dienstprogramm **dmpmqmsg** (früher **qload**) ist ab IBM MQ 8.0 in das Produkt integriert. Früher war das Dienstprogramm **qload** als SupportPac MO03 verfügbar.

Übersicht

Mit dem Dienstprogramm **dmpmqmsg** können Sie den Inhalt einer Warteschlange oder ihrer Nachrichten in eine Datei kopieren oder versetzen. Diese Datei kann nach Bedarf gespeichert werden und zu einem späteren Zeitpunkt verwendet werden, um die Nachrichten wieder in die Warteschlange zu laden.

Wichtig: Die Datei hat ein bestimmtes Format, das vom Dienstprogramm verstanden wird. Die Datei ist jedoch vom Menschen lesbar, sodass Sie sie in einem Editor aktualisieren können, bevor Sie sie erneut laden. Wenn Sie die Datei bearbeiten, dürfen Sie ihr Format nicht ändern.

Folgende Verwendungsmöglichkeiten sind möglich:

- Speichern der Nachrichten, die sich in einer Warteschlange befinden, in eine Datei. Möglicherweise zu Archivierungszwecken, und später erneut in eine Warteschlange zurückgeladen.
- Erneutes Laden einer Warteschlange mit Nachrichten, die Sie zuvor in einer Datei gespeichert haben.
- Alte Nachrichten werden aus einer Warteschlange entfernt.
- ' Testnachrichten von einer gespeicherten Position wiederverwenden, wenn erforderlich, sogar die richtige Zeit zwischen den Nachrichten beibehalten.



Achtung: SupportPac MO03 verwendet den Parameter **-l** für die Angabe der lokalen Bindung oder der Clientbindung. **-l** wurde durch den Parameter **-c** ersetzt.

-P wird jetzt für die Codepage-Informationen anstelle von **-c** verwendet.

Weitere Informationen zum Befehl und zu den verfügbaren Parametern finden Sie in [dmpmqmsg](#).

Beispiel für die Verwendung des Dienstprogramms **dmpmqmsg** unter Linux auf einer Windows-Maschine

Sie verfügen über einen Warteschlangenmanager auf einer Linux-Maschine, der Nachrichten in einer Warteschlange (*Q1*) hält, die Sie in eine andere Warteschlange (*Q2*) im selben Warteschlangenmanager verschieben möchten. Sie möchten das Dienstprogramm **dmpmqmsg** von einer Windows-Maschine aus starten.

Die Warteschlange (*Q1*) enthält vier Nachrichten, die mithilfe des Beispiels **amqspu**t (lokaler Warteschlangenmanager) oder **amqspu**tc (ferne WS-Manager-Anwendung) hinzugefügt wurden.

Auf der Linux-Maschine wird Folgendes angezeigt:

```
display ql(Q1) CURDEPTH
      2 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Setzen Sie die Umgebungsvariable MQSERVER so, dass sie auf den Warteschlangenmanager in Linux verweist. Beispiel:

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

Dabei steht *veracruz* für den Namen der Maschine.

Führen Sie das Dienstprogramm **dmpmqmsg** aus, um aus der Warteschlange *Q1* zu lesen, und speichern Sie die Ausgabe in `c:\temp\mqqload.txt`.

Stellen Sie als ferner Client eine Verbindung zum Warteschlangenmanager *QM_VER* her, der in dem Linux-Host und an dem Port aktiv ist, die von MQSERVER eingerichtet wurden. Sie erreichen die Verbindung als fernen Client, indem Sie das Attribut `-c` verwenden.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqload.txt -c
Read      - Files:    0  Messages:    4  Bytes:    22
Written   - Files:    1  Messages:    4  Bytes:    22
```

Die Ausgabedatei `c:\temp\mqqload.txt` enthält Text, wobei ein Format verwendet wird, das vom Dienstprogramm **dmpmqmsg** verstanden wird.

Geben Sie auf der Windows-Maschine den Befehl **dmpmqmsg** aus (mit der Option `-o` statt der Option `-i`), um die Warteschlange (*Q2*) aus einer Datei auf der Windows-Maschine auf die Linux-Maschine zu laden:

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqload.txt -c
Read      - Files:    1  Messages:    4  Bytes:    22
Written   - Files:    0  Messages:    4  Bytes:    22
```

Beachten Sie, dass auf der Linux-Maschine jetzt vier Nachrichten in der Warteschlange stehen, die aus der Datei wiederhergestellt wurden.

```
display ql(Q2) CURDEPTH
      6 : display ql(Q2) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q2)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Auf der Linux-Maschine:

Löschen Sie die Nachrichten aus der ursprünglichen Warteschlange.

```
clear qlocal(Q1)
    4 : clear qlocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

Bestätigen Sie, dass keine Nachrichten mehr in der ursprünglichen Warteschlange vorhanden sind:

```
display ql(Q1) CURDEPTH
    5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
    QUEUE(Q1)
    TYPE(QLLOCAL)
    CURDEPTH(0)
```

Im Abschnitt [dmpmqmsg](#) finden Sie eine Beschreibung des Befehls und seiner Parameter.

Zugehörige Konzepte

„[Beispiele für die Verwendung des Dienstprogramms dmpmqmsg](#)“ auf Seite 198

Einfache Möglichkeiten zur Verwendung des Dienstprogramms **dmpmqmsg** (früher **qload**). Dieses Dienstprogramm ist ab IBM MQ 8.0 in das Produkt integriert.

Beispiele für die Verwendung des Dienstprogramms dmpmqmsg

Einfache Möglichkeiten zur Verwendung des Dienstprogramms **dmpmqmsg** (früher **qload**). Dieses Dienstprogramm ist ab IBM MQ 8.0 in das Produkt integriert.

Früher war das Dienstprogramm **qload** als SupportPac MO03 verfügbar.

Entladen einer Warteschlange in eine Datei

Verwenden Sie die folgenden Optionen in der Befehlszeile, um die Nachrichten, die sich in einer Warteschlange befinden, in eine Datei zu speichern:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

Dieser Befehl verwendet eine Kopie der Nachrichten aus der Warteschlange und speichert sie in der angegebenen Datei.

Entladen einer Warteschlange in eine Reihe von Dateien

Sie können eine Warteschlange in eine Reihe von Dateien entladen, indem Sie ein `insert` -Zeichen im Dateinamen verwenden. In diesem Modus wird jede Nachricht in eine neue Datei geschrieben:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

Mit diesem Befehl wird die Warteschlange auf Dateien, `myfile1`, `myfile2`, `myfile3` usw., entladen.

Eine Warteschlange aus einer Datei laden

Wenn Sie eine Warteschlange mit den Nachrichten, die Sie in „[Entladen einer Warteschlange in eine Datei](#)“ auf Seite 198 gespeichert haben, erneut laden möchten, verwenden Sie die folgenden Optionen in der Befehlszeile:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Mit diesem Befehl wird die Warteschlange auf Dateien, `myfile1`, `myfile2`, `myfile3` usw., entladen.

Eine Warteschlange aus einer Reihe von Dateien laden

Sie können eine Warteschlange aus einer Reihe von Dateien laden, indem Sie ein `insert` -Zeichen in dem Dateinamen verwenden. In diesem Modus wird jede Nachricht in eine neue Datei geschrieben:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Mit diesem Befehl wird die Warteschlange in Dateien, `myfile1`, `myfile2`, `myfile3` usw. geladen.

Nachrichten aus einer Warteschlange in eine andere Warteschlange kopieren

Ersetzen Sie den Dateiparameter in „[Entladen einer Warteschlange in eine Datei](#)“ auf Seite 198 durch einen anderen Warteschlangennamen und verwenden Sie die folgenden Optionen:

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

Mit diesem Befehl können die Nachrichten aus einer Warteschlange in eine andere Warteschlange kopiert werden.

Kopieren Sie die ersten 100 Nachrichten aus einer Warteschlange in eine andere Warteschlange.

Verwenden Sie den Befehl im vorherigen Beispiel und fügen Sie die Option `-r#100` hinzu:

```
dmpmqmsg -m QM1 -i Q1 -o Q2 -r#100
```

Nachrichten von einer Warteschlange in eine andere Warteschlange verschieben

Eine Variation von „[Eine Warteschlange aus einer Datei laden](#)“ auf Seite 198. Beachten Sie die Unterscheidung zwischen der Verwendung von `-i` (Kleinbuchstabe), bei der nur eine Warteschlange durchsucht wird, und `-I` (Großschreibung), in der ein destruktiver Abruf aus einer Warteschlange vorgenommen wird:

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

Nachrichten, die älter als ein Tag sind, aus einer Warteschlange in eine andere Warteschlange verschieben

Dieses Beispiel zeigt die Verwendung der Altersauswahl. Es können Nachrichten ausgewählt werden, die älter als, jünger als oder in einem Bereich von Alter sind.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

Zeigt das Alter der Nachrichten an, die sich derzeit in einer Warteschlange stellen

Verwenden Sie die folgenden Optionen in der Befehlszeile:

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

Mit der Nachrichtendatei arbeiten

Wenn Sie die Nachricht aus Ihrer Warteschlange entladen haben (wie in „[Entladen einer Warteschlange in eine Datei](#)“ auf Seite 198), können Sie die Datei bearbeiten.

Möglicherweise möchten Sie auch das Format der Datei ändern, um eine der Anzeigoptionen zu verwenden, die Sie zum Zeitpunkt der Entladen der Warteschlange nicht angegeben haben.

Sie können das Dienstprogramm **dmpmqmsg** verwenden, um die Datei auch nach dem Entladen der Warteschlange erneut in das erforderliche Format zu verarbeiten. Verwenden Sie die folgenden Optionen in der Befehlszeile.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

Im Abschnitt [dmpmqmsg](#) finden Sie eine Beschreibung des Befehls und seiner Parameter.

Ferne IBM MQ-Objekte verwalten

In diesem Abschnitt erfahren Sie, wie Sie IBM MQ-Objekte mithilfe von MQSC-Befehlen auf einem fernen Warteschlangenmanager verwalten und wie ferne Warteschlangenobjekte zum Steuern der Zieladresse von Nachrichten und Antwortnachrichten verwendet werden.

Prozedur

- Informationen zum Verwalten von fernen IBM MQ-Objekten finden Sie in den folgenden Unterabschnitten:
 - [„Kanäle und ferne Warteschlangen“](#) auf Seite 200
 - [„Ferne Verwaltung von einem lokalen WS-Manager“](#) auf Seite 202
 - [„Lokale Definition einer fernen Warteschlange erstellen“](#) auf Seite 208
 - [„Prüfen, ob asynchrone Befehle für verteilte Netze abgeschlossen sind“](#) auf Seite 211
 - [„Ferne Warteschlangendefinitionen als Aliasnamen verwenden“](#) auf Seite 213
 - [„Datenkonvertierung“](#) auf Seite 214

Kanäle und ferne Warteschlangen

Sie können einen Kanal einrichten, um eine Einweg-Kommunikationsverbindung zwischen einem lokalen WS-Manager und einem fernen Warteschlangenmanager bereitzustellen. Der Kanal kann Nachrichten übertragen, die für eine beliebige Anzahl von Warteschlangen auf dem fernen Warteschlangenmanager bestimmt sind. Wenn der ferne WS-Manager auf Nachrichten antworten soll, die vom lokalen Warteschlangenmanager gesendet werden, können Sie einen zweiten Kanal einrichten, um Antworten vom fernen Warteschlangenmanager zurück an den lokalen Warteschlangenmanager zu senden.

Ein WS-Manager kommuniziert mit einem anderen Warteschlangenmanager, indem er eine Nachricht sendet und, falls erforderlich, eine Antwort zurückerhält. Der empfangende WS-Manager könnte wie folgt sein:

- Auf derselben Maschine
- Auf einer anderen Maschine in der gleichen Lage (oder sogar auf der anderen Seite der Welt)
- Ausführung auf derselben Plattform wie der lokale WS-Manager
- Ausführung auf einer anderen Plattform, die von IBM MQ unterstützt wird

Diese Nachrichten können aus folgenden Nachrichten stammen:

- Benutzergeschriebene Anwendungsprogramme, die Daten von einem Knoten in einen anderen übertragen
- Benutzergeschriebene Verwaltungsanwendungen, die PCF-Befehle oder die MQAI verwenden
- IBM MQ Explorer
- Warteschlangenmanager senden:
 - Instrumentierungsereignisnachrichten an einen anderen WS-Manager

- MQSC-Befehle, die von einem **runmqsc** -Befehl im indirekten Modus ausgegeben werden (die Befehle werden auf einem anderen Warteschlangenmanager ausgeführt)

Bevor eine Nachricht an einen fernen Warteschlangenmanager gesendet werden kann, benötigt der lokale WS-Manager einen Mechanismus zum Erkennen der Ankunft von Nachrichten und Transport, die aus folgenden Bereichen bestehen:

- Mindestens ein Kanal
- eine Übertragungswarteschlange
- Kanalinitiator

Damit ein ferner Warteschlangenmanager eine Nachricht empfangen kann, ist ein Empfangsprogramm erforderlich.

Ein Kanal ist eine Einweg-Kommunikationsverbindung zwischen zwei Warteschlangenmanagern und kann Nachrichten übertragen, die für eine beliebige Anzahl von Warteschlangen auf dem fernen Warteschlangenmanager bestimmt sind.

Jedes Ende des Kanals verfügt über eine separate Definition. Wenn ein Ende z. B. ein Sender oder ein Server ist, muss das andere Ende ein Empfänger oder ein Requester sein. Ein einfacher Kanal besteht aus einer *Senderkanaldefinition* am lokalen WS-Manager-Ende und einer *Empfängerkanaldefinition* auf dem fernen WS-Manager-Ende. Die beiden Definitionen müssen denselben Namen haben und zusammen einen einzigen Nachrichtenkanal bilden.

Wenn der ferne WS-Manager auf Nachrichten antworten soll, die vom lokalen Warteschlangenmanager gesendet werden, konfigurieren Sie einen zweiten Kanal, um Antworten an den lokalen Warteschlangenmanager zu senden.

Mit dem MQSC-Befehl **DEFINE CHANNEL** können Sie Kanäle definieren. In diesem Abschnitt verwenden die Beispiele in Bezug auf Kanäle die Standardkanalattribute, sofern nicht anders angegeben.

An jedem Ende eines Kanals befindet sich ein Nachrichtenkanalagent (MCA), der das Senden und Empfangen von Nachrichten steuert. Der MCA nimmt Nachrichten aus der Übertragungswarteschlange und stellt sie in die Kommunikationsverbindung zwischen den Warteschlangenmanagern.

Eine Übertragungswarteschlange ist eine spezielle lokale Warteschlange, in der Nachrichten temporär gespeichert werden, bevor der MCA sie abholt und sie an den fernen Warteschlangenmanager sendet. Sie geben den Namen der Übertragungswarteschlange in einer *Definition der fernen Warteschlange* an.

Sie können einem Nachrichtenkanalagenten ermöglichen, Nachrichten unter Verwendung mehrerer Threads zu übertragen. Dieser Prozess ist als *Pipelining* bekannt. Mithilfe von Pipelining kann der Nachrichtenkanalagent Nachrichten effizienter übertragen, was die Kanalleistung verbessert. Ausführliche Informationen zum Konfigurieren eines Kanals für die Verwendung von Pipelining finden Sie unter [Attribute von Kanälen](#).

In „[Kanäle und Übertragungswarteschlangen für die Fernverwaltung vorbereiten](#)“ auf Seite 203 erfahren Sie, wie Sie diese Definitionen verwenden, um die Fernverwaltung einzurichten.

Weitere Informationen zum Einrichten der verteilten Warteschlangensteuerung im Allgemeinen finden Sie im Abschnitt [Verteilte Warteschlangenkomponenten](#).

Zugehörige Informationen

[runmqsc \(MQSC-Befehle ausführen\)](#)

Fernverwaltung mit Clustern

In einem IBM MQ-Netz, das die verteilte Steuerung von Warteschlangen verwendet, ist jeder Warteschlangenmanager unabhängig. Wenn ein Warteschlangenmanager Nachrichten an einen anderen WS-Manager senden muss, muss er eine Übertragungswarteschlange, einen Kanal zum fernen Warteschlangenmanager und eine Definition einer fernen Warteschlange für jede Warteschlange definieren, an die Nachrichten gesendet werden sollen.

Ein *Cluster* ist eine Gruppe von Warteschlangenmanagern, die so konfiguriert sind, dass die Warteschlangenmanager über ein einziges Netz ohne komplexe Übertragungswarteschlange, Kanal und Warteschlangendefinitionen direkt miteinander kommunizieren können. Cluster können ohne großen Aufwand konfiguriert werden und enthalten in der Regel Warteschlangenmanager, die logisch miteinander verknüpft sind und Daten oder Anwendungen gemeinsam nutzen müssen. Selbst der kleinste Cluster reduziert die Systemverwaltungskosten.

Die Einrichtung eines Netzes von Warteschlangenmanagern in einem Cluster umfasst weniger Definitionen als die Erstellung einer traditionellen verteilten Warteschlangenumgebung. Wenn weniger Definitionen vorhanden sind, können Sie Ihr Netz schneller und einfacher einrichten oder ändern und das Risiko verringern, einen Fehler in Ihren Definitionen zu erstellen.




Um einen Cluster zu konfigurieren, benötigen Sie einen Clustersendersender (CLUSDR) und eine Clusterempfängerdefinition (CLUSRCVR) für jeden Warteschlangenmanager. Sie benötigen keine Übertragungswarteschlangendefinitionen oder Definitionen ferner Warteschlangen. Die Prinzipien der Fernverwaltung sind dieselben, wenn sie in einem Cluster verwendet werden, aber die Definitionen selbst sind stark vereinfacht.

Ferne Verwaltung von einem lokalen WS-Manager

Sie können einen fernen Warteschlangenmanager unter Verwendung von MQSC- und PCF-Befehlen aus einem lokalen Warteschlangenmanager verwalten.

Das Vorbereiten der Warteschlangen und Kanäle ist für die beiden MQSC- und PCF-Befehle im Wesentlichen identisch. In diesem Abschnitt werden in den Beispielen die MQSC-Befehle angezeigt, da sie einfacher zu verstehen sind. Weitere Informationen zum Schreiben von Verwaltungsprogrammen mit Hilfe von PCF-Befehlen finden Sie in „[IBM MQ Programmable Command Formats verwenden](#)“ auf Seite 23.

Sie senden MQSC-Befehle entweder interaktiv oder aus einer Textdatei, die die Befehle enthält, an einen fernen Warteschlangenmanager. Der ferne WS-Manager kann sich auf derselben Maschine befinden oder, in der Regel, auf einer anderen Maschine. Sie können Warteschlangenmanager in anderen IBM MQ-Umgebungen über Fernzugriff verwalten, einschließlich:

-  UNIX
-  Linux
-  Windows
-  IBM i
-  z/OS

Um die Fernverwaltung zu implementieren, müssen Sie bestimmte Objekte erstellen. Wenn Sie keine speziellen Anforderungen angegeben haben, sind die Standardwerte, z. B. für die maximale Nachrichtenlänge, ausreichend.

Vorbereiten von Warteschlangenmanagern für die Fernverwaltung

Verwendung von MQSC-Befehlen zum Vorbereiten von Warteschlangenmanagern für die Fernverwaltung.

Abbildung 18 auf Seite 203 zeigt die Konfiguration von Warteschlangenmanagern und Kanälen, die Sie für die Fernverwaltung mit dem Befehl **runmqsc** benötigen. Das Objekt `source.queue.manager` ist der Quellenwarteschlangenmanager, von dem aus Sie MQSC-Befehle ausgeben können und zu dem die Ergebnisse dieser Befehle (Bedienernachrichten) zurückgegeben werden. Das Objekt `target.queue.manager` ist der Name des Zielwarteschlangenmanagers, der die Befehle verarbeitet und alle Bedienernachrichten generiert.

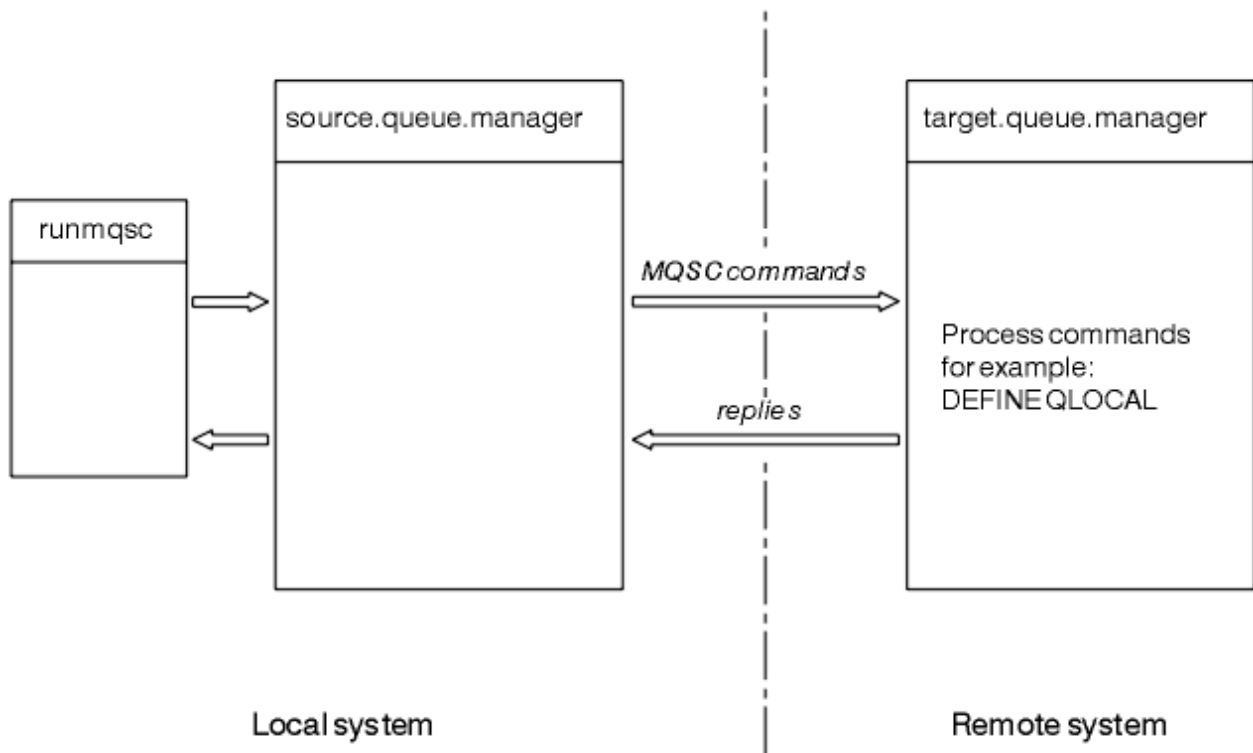


Abbildung 18. Ferne Verwaltung mit MQSC-Befehlen

Auf beiden Systemen, wenn Sie dies noch nicht getan haben:

- Erstellen Sie den Warteschlangenmanager und die Standardobjekte mit dem Befehl `crtmqm`. Weitere Informationen finden Sie unter [crtmqm](#).
- Starten Sie den WS-Manager mit dem Befehl `strmqm`. Weitere Informationen finden Sie im Abschnitt [strmqm](#).

Auf dem Zielwarteschlangenmanager:

- Die Befehlswarteschlange `SYSTEM.ADMIN.COMMAND.QUEUE` muss vorhanden sein. Diese Warteschlange wird standardmäßig bei der Erstellung eines Warteschlangenmanagers erstellt.

Sie müssen diese Befehle lokal oder über eine Netzfunktion wie Telnet ausführen.

Kanäle und Übertragungswarteschlangen für die Fernverwaltung vorbereiten

Verwendung von MQSC-Befehlen zum Bereiten von Kanälen und Übertragungswarteschlangen für die Fernverwaltung.

Um MQSC-Befehle über Remotezugriff auszuführen, konfigurieren Sie zwei Kanäle, eine für jede Richtung, und die zugehörigen Übertragungswarteschlangen. In diesem Beispiel wird vorausgesetzt, dass Sie TCP/IP als Transporttyp verwenden und dass Sie die TCP/IP-Adresse kennen.

Der Kanal `source.to.target` dient zum Senden von MQSC-Befehlen vom Quellenwarteschlangenmanager an den Ziel-WS-Manager. Der Sender befindet sich unter `source.queue.manager` und sein Empfänger befindet sich unter `target.queue.manager`. Der Kanal `target.to.source` dient zum Zurückgeben der Ausgabe von Befehlen und aller Bedienernachrichten, die für den Quellenwarteschlangenmanager generiert werden. Außerdem müssen Sie für jeden Kanal eine Übertragungswarteschlange definieren. Bei dieser Warteschlange handelt es sich um eine lokale Warteschlange, die den Namen des empfangenden Warteschlangenmanagers erhält. Der XMITQ-Name muss mit dem Namen des fernen Warteschlangenmanagers übereinstimmen, damit die ferne Verwaltung funktioniert, es sei denn, Sie verwenden einen WS-Manager-Aliasnamen. In [Abbildung 19 auf Seite 204](#) ist diese Konfiguration zusammengefasst.

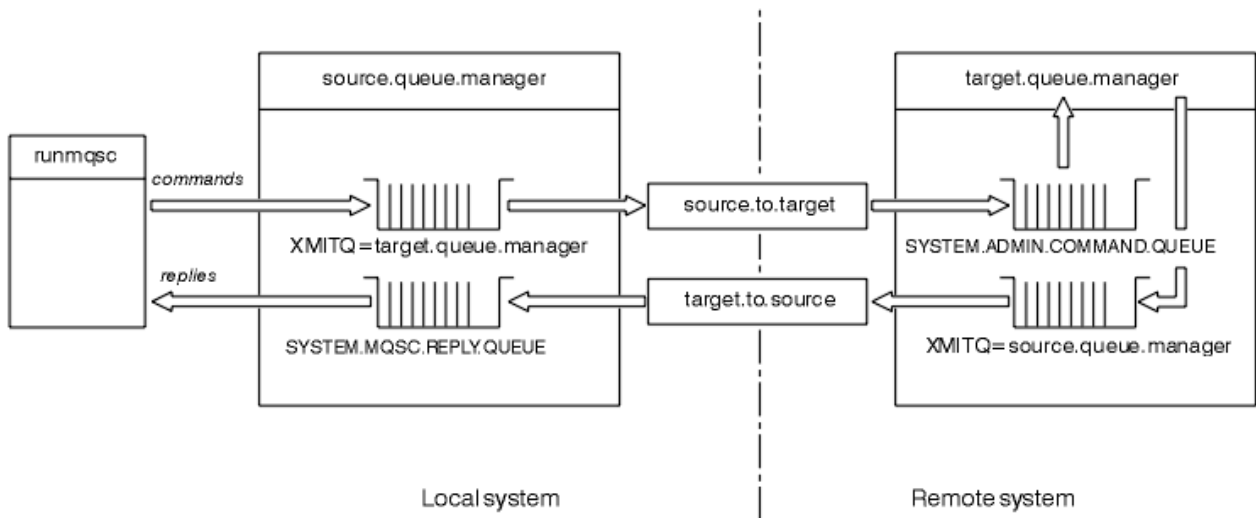


Abbildung 19. Kanäle und Warteschlangen für die Fernverwaltung konfigurieren

Weitere Informationen zum Konfigurieren von Kanälen finden Sie unter [Configuring distributed queuing](#).

Kanäle, Empfangsprogramme und Übertragungswarteschlangen definieren

Geben Sie auf dem Quellenwarteschlangenmanager (`source.queue.manager`) die folgenden MQSC-Befehle aus, um die Kanäle, den Listener und die Übertragungswarteschlange zu definieren:

1. Definieren Sie den Senderkanal auf dem Quellen-WS-Manager:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(SDR) +
CONNNAME (RHX5498) +
XMITQ ('target.queue.manager') +
TRPTYPE(TCP)
```

2. Definieren Sie den Empfängerkanal auf dem Quellenwarteschlangenmanager:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

3. Definieren Sie den Listener auf dem Quellen-WS-Manager:

```
DEFINE LISTENER ('source.queue.manager') +
TRPTYPE (TCP)
```

4. Definieren Sie die Übertragungswarteschlange auf dem Quellen-WS-Manager:

```
DEFINE QLOCAL ('target.queue.manager') +
USAGE (XMITQ)
```

Setzen Sie die folgenden Befehle auf dem Zielwarteschlangenmanager (`target.queue.manager`) ab, um die Kanäle, den Listener und die Übertragungswarteschlange zu erstellen:

1. Definieren Sie den Senderkanal auf dem Ziel-WS-Manager:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(SDR) +
CONNNAME (RHX7721) +
XMITQ ('source.queue.manager') +
TRPTYPE(TCP)
```

2. Definieren Sie den Empfängerkanal auf dem Ziel-WS-Manager:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

3. Definieren Sie den Listener auf dem Ziel-WS-Manager:

```
DEFINE LISTENER ('target.queue.manager') +
TRPTYPE (TCP)
```

4. Definieren Sie die Übertragungswarteschlange auf dem Ziel-WS-Manager:

```
DEFINE QLOCAL ('source.queue.manager') +
USAGE (XMITQ)
```

Anmerkung: Die für das Attribut CONNAME in den Senderkanaldefinitionen angegebenen TCP/IP-Verbindungsnamen sind nur zur Veranschaulichung bestimmt. Dies ist der Netzname der Maschine am *anderen* Ende der Verbindung. Verwenden Sie die Werte, die für Ihr Netz geeignet sind.

Empfangsprogramme und Kanäle starten

Verwendung von MQSC-Befehlen zum Starten von Empfangsprogrammen und Kanälen.

Starten Sie beide Empfangsprogramme mit den folgenden MQSC-Befehlen:

1. Starten Sie den Listener auf dem Quellen-WS-Manager, `source.queue.manager`, indem Sie den folgenden MQSC-Befehl ausgeben:

```
START LISTENER ('source.queue.manager')
```

2. Starten Sie den Listener auf dem Zielwarteschlangenmanager, `target.queue.manager`, indem Sie den folgenden MQSC-Befehl ausgeben:

```
START LISTENER ('target.queue.manager')
```

Starten Sie beide Senderkanäle, indem Sie die folgenden MQSC-Befehle verwenden:

1. Starten Sie den Senderkanal auf dem Quellenwarteschlangenmanager, `source.queue.manager`, indem Sie den folgenden MQSC-Befehl ausgeben:

```
START CHANNEL ('source.to.target')
```

2. Starten Sie den Senderkanal auf dem Zielwarteschlangenmanager, `target.queue.manager`, indem Sie den folgenden MQSC-Befehl ausgeben:

```
START CHANNEL ('target.to.source')
```

Automatische Definition von Kanälen

Sie aktivieren die automatische Definition von Empfänger- und Serververbindungsdefinitionen, indem Sie das WS-Manager-Objekt mit dem WebSphere MQ-Scriptbefehl ALTER QMGR (oder mit dem PCF-Befehl Change Queue Manager) aktualisieren.

Wenn IBM MQ eine eingehende Verbindungsanforderung empfängt und keinen geeigneten Empfänger- oder Serververbindungskanal findet, wird automatisch ein Kanal erstellt. Automatische Definitionen basieren auf zwei Standarddefinitionen, die mit IBM MQ bereitgestellt werden: SYSTEM.AUTO.RECEIVER und SYSTEM.AUTO.SVRCONN

Weitere Informationen zum automatischen Erstellen von Kanaldefinitionen finden Sie unter [Kanäle vorbereiten](#). Informationen zum automatischen Definieren von Kanälen für Cluster finden Sie im Abschnitt [Mit automatisch definierten Kanälen arbeiten](#).

Den Befehlsserver für die Fernverwaltung verwalten

Vorgehensweise zum Starten, Stoppen und Anzeigen des Status des Befehlsservers. Ein Befehlsserver ist für die gesamte Verwaltung mit PCF-Befehlen, der MQAI und auch für die Fernverwaltung obligatorisch.

Jedem WS-Manager kann ein Befehlsserver zugeordnet sein. Ein Befehlsserver verarbeitet alle ankommenden Befehle von fernen Warteschlangenmanagern oder PCF-Befehlen aus Anwendungen. Er stellt die Befehle für den Warteschlangenmanager zur Verarbeitung dar und gibt abhängig vom Ursprung des Befehls einen Beendigungscode oder eine Bedienernachricht zurück.

Anmerkung: Stellen Sie für die Fernverwaltung sicher, dass der Zielwarteschlangenmanager aktiv ist. Andernfalls können die Nachrichten, die Befehle enthalten, den WS-Manager nicht verlassen, von dem sie ausgegeben werden. Stattdessen werden diese Nachrichten in die lokale Übertragungswarteschlange eingereiht, die dem fernen Warteschlangenmanager dient. Vermeiden Sie diese Situation.

Es gibt separate Steuerbefehle zum Starten und Stoppen des Befehlsservers. Wenn der Befehlsserver aktiv ist, können Benutzer von IBM MQ for Windows oder IBM MQ for Linux (x86 -und x86-64 -Plattformen) die in den folgenden Abschnitten beschriebenen Operationen mit IBM MQ Explorer ausführen. Weitere Informationen finden Sie unter [„Verwaltung über die IBM MQ Explorer“](#) auf Seite 135.

Starten des Befehlsservers

Abhängig vom Wert des WS-Managerattributs *SCMDSERV* wird der Befehlsserver entweder automatisch gestartet, wenn der Warteschlangenmanager gestartet wird oder manuell gestartet werden muss. Der Wert des Warteschlangenmanagerattributs kann mit dem MQSC-Befehl *ALTER QMGR* unter Angabe des Parameters *SCMDSERV* geändert werden. Standardmäßig wird der Befehlsserver automatisch gestartet.

Wenn *SCMDSERV* auf *MANUAL* gesetzt ist, starten Sie den Befehlsserver mit dem folgenden Befehl:

```
stmqcsv saturn.queue.manager
```

Hierbei steht *saturn.queue.manager* für den Warteschlangenmanager, für den der Befehlsserver gestartet wird.

Status des Befehlsservers anzeigen

Stellen Sie für die Fernverwaltung sicher, dass der Befehlsserver auf dem Zielwarteschlangenmanager ausgeführt wird. Wenn der Befehl nicht aktiv ist, können ferne Befehle nicht verarbeitet werden. Alle Nachrichten, die Befehle enthalten, werden in der Befehlswarteschlange des Zielwarteschlangenmanagers in die Warteschlange eingereiht.

Geben Sie den folgenden MQSC-Befehl aus, um den Status des Befehlsservers für einen WS-Manager anzuzeigen:

```
DISPLAY QMSTATUS CMDSERV
```

Befehlsserver stoppen

Verwenden Sie den folgenden Befehl, um den mit dem vorherigen Beispiel gestarteten Befehlsserver zu beenden:

```
endmqcsv saturn.queue.manager
```

Sie können den Befehlsserver auf zwei Arten stoppen:

- Verwenden Sie für einen kontrollierten Stopp den Befehl *endmqcsv* mit dem Flag *-c*, das die Standardeinstellung ist.
- Verwenden Sie für einen sofortigen Stopp den Befehl *endmqcsv* mit dem Flag *-i*.

Anmerkung: Wenn Sie einen WS-Manager stoppen, wird auch der ihm zugeordnete Befehlsserver beendet.

MQSC-Befehle auf einem fernen WS-Manager absetzen

Sie können eine bestimmte Form des Befehls `runmqsc` verwenden, um MQSC-Befehle auf einem fernen WS-Manager auszuführen.

Der Befehlsserver muss im Zielwarteschlangenmanager aktiv sein, wenn er über Fernzugriff MQSC-Befehle ausführen soll. (Dies ist auf dem Quellenwarteschlangenmanager nicht erforderlich.) Informationen zum Starten des Befehlsservers auf einem Warteschlangenmanager finden Sie im Abschnitt [„Den Befehlsserver für die Fernverwaltung verwalten“](#) auf Seite 206.

Auf dem Quellenwarteschlangenmanager können Sie dann MQSC-Befehle interaktiv im indirekten Modus ausführen, indem Sie Folgendes eingeben:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

Dieses Format des Befehls `runmqsc` mit dem Flag `-w` führt die MQSC-Befehle im indirekten Modus aus, wobei Befehle (in geänderter Form) in die Eingabewarteschlange des Befehlsservers gestellt und in der angegebenen Reihenfolge ausgeführt werden.

Wenn Sie einen MQSC-Befehl eingeben, wird er in diesem Fall an den fernen WS-Manager umgeleitet, in diesem Fall `target.queue.manager`. Das Zeitlimit wird auf 30 Sekunden gesetzt. Wenn eine Antwort nicht innerhalb von 30 Sekunden empfangen wird, wird die folgende Nachricht auf dem lokalen (Quellenwarteschlangenmanager) generiert:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

Wenn Sie die Ausgabe von MQSC-Befehlen stoppen, zeigt der lokale WS-Manager alle zeitlich getakten Antworten an, die eingetroffen sind, und löscht alle weiteren Antworten.

Der Quellenwarteschlangenmanager nimmt standardmäßig den lokalen Standardwarteschlangenmanager an. Wenn Sie die Option `-m LocalQmgrName` im Befehl `runmqsc` angeben, können Sie die Befehle direkt über jeden lokalen WS-Manager ausführen.

Im indirekten Modus können Sie auch eine MQSC-Befehlsdatei auf einem fernen WS-Manager ausführen. Beispiel:

```
runmqsc -w 60 target.queue.manager < mycomds.in > report.out
```

Hierbei steht `mycomds.in` für eine Datei, die MQSC-Befehle enthält, und `report.out` ist die Berichtsdatei.

Empfohlene Methode zum Fernabsetzen von Befehlen

Wenn Sie Befehle für einen fernen Warteschlangenmanager absetzen, sollten Sie die folgenden Schritte ausführen:

1. Stellen Sie die MQSC-Befehle auf dem fernen System in einer Befehlsdatei zur Ausführung.
2. Überprüfen Sie Ihre MQSC-Befehle lokal, indem Sie das Flag `-v` im Befehl `runmqsc` angeben.
Sie können `runmqsc` nicht verwenden, um MQSC-Befehle in einem anderen WS-Manager zu überprüfen.
3. Überprüfen Sie, ob die Befehlsdatei lokal ohne Fehler ausgeführt wird.
4. Führen Sie die Befehlsdatei auf dem fernen System aus.

Bei Problemen mit MQSC-Befehlen über Remotezugriff

Wenn Sie bei der Ausführung von MQSC-Befehlen über Remotezugriff Schwierigkeiten haben, stellen Sie sicher, dass die folgenden Schritte ausgeführt werden

- Der Befehlsserver wurde auf dem Ziel-WS-Manager gestartet.
- Definiert eine gültige Übertragungswarteschlange.
- Definiert die beiden Enden der Nachrichtenkanäle für beide:
 - Der Kanal, über den die Befehle gesendet werden.
 - Der Kanal, entlang dem die Antworten zurückgegeben werden sollen.
- Geben Sie den korrekten Verbindungsnamen (CONNNAME) in der Kanaldefinition an.
- Sie haben die Empfangsprogramme gestartet, bevor Sie die Nachrichtenkanäle gestartet haben.
- Markiert, dass das Unterbrechungsintervall nicht abgelaufen ist, z. B. wenn ein Kanal gestartet wurde, aber dann nach einiger Zeit heruntergefahren wurde. Dies ist besonders wichtig, wenn Sie die Kanäle manuell starten.
- Gesendete Anforderungen von einem Quellenwarteschlangenmanager, die für den Zielwarteschlangenmanager nicht sinnvoll sind (z. B. Anforderungen, die Parameter enthalten, die auf dem fernen Warteschlangenmanager nicht unterstützt werden).

Weitere Informationen hierzu finden Sie im Abschnitt [„Probleme mit MQSC-Befehlen beheben“](#) auf Seite 20.

Mit Warteschlangenmanagern unter z/OS arbeiten

Sie können MQSC-Befehle an einen z/OS-Warteschlangenmanager von einem Warteschlangenmanager auf den in diesem Handbuch beschriebenen Plattformen absetzen. Um dies zu tun, müssen Sie jedoch den Befehl `runmqsc` und die Kanaldefinitionen am Absender ändern.

Fügen Sie insbesondere das Flag `-x` dem Befehl `runmqsc` auf dem Quellenknoten hinzu, um anzugeben, dass der Zielwarteschlangenmanager unter z/OS ausgeführt wird:

```
runmqsc -w 30 -x target.queue.manager
```

Lokale Definition einer fernen Warteschlange erstellen

Eine lokale Definition einer fernen Warteschlange ist eine Definition in einem lokalen Warteschlangenmanager, die sich auf eine Warteschlange in einem fernen Warteschlangenmanager bezieht.

Sie müssen keine ferne Warteschlange aus einer lokalen Position definieren, aber der Vorteil ist, dass Anwendungen auf die ferne Warteschlange mit ihrem lokal definierten Namen verweisen können, statt einen Namen anzugeben, der durch die ID des Warteschlangenmanagers qualifiziert wird, auf dem sich die ferne Warteschlange befindet.

Informationen zur Funktionsweise lokaler Definitionen von fernen Warteschlangen

Eine Anwendung stellt eine Verbindung zu einem lokalen WS-Manager her und gibt dann einen MQOPEN -Aufruf aus. Im geöffneten Aufruf ist der angegebene Warteschlangename der Name einer fernen Warteschlangendefinition im lokalen Warteschlangenmanager. Die Definition der fernen Warteschlange stellt die Namen der Zielwarteschlange, des Zielwarteschlangenmanagers und optional einer Übertragungswarteschlange bereit. Um eine Nachricht in die ferne Warteschlange zu stellen, gibt die Anwendung einen MQPUT -Aufruf aus und gibt dabei die Kennung an, die vom Aufruf MQOPEN zurückgegeben wurde. Der WS-Manager verwendet den Namen der fernen Warteschlange und den Namen des fernen Warteschlangenmanagers in einem Übertragungsheader am Anfang der Nachricht. Diese Informationen werden verwendet, um die Nachricht an ihr korrektes Ziel im Netz weiterzuleiten.

Als Administrator können Sie die Zieladresse der Nachricht steuern, indem Sie die Definition der fernen Warteschlange ändern.

Das folgende Beispiel zeigt, wie eine Anwendung eine Nachricht in eine Warteschlange einreicht, deren Eigner ein ferner WS-Manager ist. Die Anwendung stellt eine Verbindung zu einem WS-Manager her, z. B. saturn.queue.manager. Die Zielwarteschlange ist Eigentum eines anderen Warteschlangenmanagers.

Im Aufruf MQOPEN gibt die Anwendung die folgenden Felder an:

Feldwert	Beschreibung
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Gibt den lokalen Namen des fernen Warteschlangenobjekts an. Dadurch werden die Zielwarteschlange und der Zielwarteschlangenmanager definiert.
<i>ObjectType</i> (Warteschlange)	Bezeichnet dieses Objekt als Warteschlange.
<i>ObjectQmgrName</i> Leer oder saturn.queue.manager	Dieses Feld ist optional. Wenn dieses Feld leer ist, wird der Name des lokalen Warteschlangenmanagers angenommen. (Hierbei handelt es sich um den Warteschlangenmanager, auf dem die Definition der fernen Warteschlange vorhanden ist.)

Danach gibt die Anwendung einen MQPUT -Aufruf aus, um eine Nachricht in diese Warteschlange zu stellen.

Auf dem lokalen WS-Manager können Sie mit den folgenden MQSC-Befehlen eine lokale Definition einer fernen Warteschlange erstellen:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
RQMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

Dabei gilt:

QREMOTE (CYAN.REMOTE.QUEUE)

Gibt den lokalen Namen des fernen Warteschlangenobjekts an. Dies ist der Name, den Anwendungen, die mit diesem WS-Manager verbunden sind, im Aufruf MQOPEN angeben müssen, um die Warteschlange AUTOMOBILE.INSURANCE.QUOTE.QUEUE auf dem fernen Warteschlangenmanager jupiter.queue.manager zu öffnen.

DESCR ('Queue for auto insurance requests from the branches')

Stellt zusätzlichen Text zur Verfügung, der die Verwendung der Warteschlange beschreibt.

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

Gibt den Namen der Zielwarteschlange auf dem fernen WS-Manager an. Dies ist die echte Zielwarteschlange für Nachrichten, die von Anwendungen gesendet werden, die den Warteschlangennamen CYAN.REMOTE.QUEUE angeben. Die Warteschlange AUTOMOBILE.INSURANCE.QUOTE.QUEUE muss als lokale Warteschlange auf dem fernen WS-Manager definiert sein.

RQMNAME (jupiter.queue.manager)

Gibt den Namen des fernen Warteschlangenmanagers an, der Eigner der Zielwarteschlange ist AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

XMITQ (INQUOTE.XMIT.QUEUE)

Gibt den Namen der Übertragungswarteschlange an. Dies ist optional. Wenn der Name einer Übertragungswarteschlange nicht angegeben wird, wird eine Warteschlange mit demselben Namen wie der ferne Warteschlangenmanager verwendet.

In beiden Fällen muss die entsprechende Übertragungswarteschlange als lokale Warteschlange mit einem Attribut **Usage** definiert sein, in dem angegeben ist, dass es sich um eine Übertragungswarteschlange (USAGE (XMITQ) in MQSC-Befehlen) handelt.

Eine alternative Möglichkeit, Nachrichten in eine ferne Warteschlange zu stellen

Die Verwendung einer lokalen Definition einer fernen Warteschlange ist nicht die einzige Möglichkeit, Nachrichten in eine ferne Warteschlange zu stellen. Anwendungen können den vollständigen Warteschlangennamen, einschließlich des Namens des fernen Warteschlangenmanagers, als Teil des Aufrufs MQOPEN angeben. In diesem Fall benötigen Sie keine lokale Definition einer fernen Warteschlange. Dies bedeutet jedoch, dass Anwendungen zur Ausführungszeit entweder den Namen des fernen Warteschlangenmanagers kennen oder Zugriff darauf haben müssen.

Andere Befehle mit fernen Warteschlangen verwenden

Sie können MQSC-Befehle verwenden, um die Attribute eines fernen Warteschlangenobjekts anzuzeigen oder zu ändern, oder Sie können das ferne Warteschlangenobjekt löschen. Beispiel:

- Gehen Sie wie folgt vor, um die Attribute der fernen Warteschlange anzuzeigen

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Sie können die ferne Warteschlange ändern, um sie zu aktivieren. Dies hat keine Auswirkungen auf die Zielwarteschlange, sondern nur Anwendungen, die diese ferne Warteschlange angeben:

```
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)
```

- Zum Löschen dieser fernen Warteschlange. Dies hat keine Auswirkungen auf die Zielwarteschlange, sondern nur ihre lokale Definition:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Anmerkung: Wenn Sie eine ferne Warteschlange löschen, löschen Sie nur die lokale Darstellung der fernen Warteschlange. Sie löschen nicht die ferne Warteschlange selbst oder alle Nachrichten darauf.

Übertragungswarteschlange definieren

Eine Übertragungswarteschlange ist eine lokale Warteschlange, die verwendet wird, wenn ein WS-Manager Nachrichten über einen Nachrichtenkanal an einen fernen Warteschlangenmanager weiterleitet.

Der Kanal stellt einen Einweg-Link zum fernen Warteschlangenmanager bereit. Nachrichten werden in der Übertragungswarteschlange in die Warteschlange gestellt, bis der Kanal sie akzeptieren kann. Wenn Sie einen Kanal definieren, müssen Sie einen Namen für die Übertragungswarteschlange an der sendenden Seite des Nachrichtenkanals angeben.

Das MQSC-Befehlsattribut USAGE definiert, ob es sich bei einer Warteschlange um eine Übertragungswarteschlange oder eine normale Warteschlange handelt.

Standardübertragungswarteschlangen

Wenn ein Warteschlangenmanager Nachrichten an einen fernen Warteschlangenmanager sendet, ermittelt er die Übertragungswarteschlange in der folgenden Reihenfolge:

1. Die Übertragungswarteschlange, die im Attribut XMITQ der lokalen Definition einer fernen Warteschlange angegeben ist.
2. Eine Übertragungswarteschlange mit demselben Namen wie der Zielwarteschlangenmanager. (Dieser Wert ist der Standardwert in XMITQ für die lokale Definition einer fernen Warteschlange.)
3. Die Übertragungswarteschlange, die im Attribut DEFXMITQ des lokalen Warteschlangenmanagers angegeben ist.

Der folgende MQSC-Befehl erstellt beispielsweise eine Standardübertragungswarteschlange auf `source.queue.manager` für Nachrichten, die an `target.queue.manager` gesendet werden:

```
DEFINE QLOCAL ('target.queue.manager') +
DESCR ('Default transmission queue for target qm') +
USAGE (XMITQ)
```

Anwendungen können Nachrichten direkt in eine Übertragungswarteschlange einreihen oder indirekt mithilfe der Definition einer fernen Warteschlange. Weitere Informationen hierzu finden Sie im Abschnitt „[Lokale Definition einer fernen Warteschlange erstellen](#)“ auf Seite 208.

Prüfen, ob asynchrone Befehle für verteilte Netze abgeschlossen sind

Viele Befehle sind asynchron, wenn sie in einem verteilten Netz verwendet werden. Abhängig von dem Befehl und dem Netzstatus, wenn er ausgegeben wird, kann es sehr viel Zeit in Anspruch nehmen, die Verarbeitung abzuschließen. Der WS-Manager gibt keine Nachricht nach Abschluss aus, so dass Sie andere Möglichkeiten zur Überprüfung der Ausführung des Befehls benötigen.

Informationen zu diesem Vorgang

Fast jede Konfigurationsänderung, die Sie an einem Cluster vornehmen, ist wahrscheinlich asynchron abgeschlossen. Dies liegt an der internen Verwaltung und Aktualisierung von Zyklen, die in Clustern betrieben werden. Bei Publish/Subscribe-Hierarchien ist jede Konfigurationsänderung, die sich auf Subskriptionen auswirkt, wahrscheinlich asynchron abgeschlossen. Dies ist bei dem Namen des Befehls nicht immer klar.

Die folgenden MQSC-Befehle werden möglicherweise asynchron ausgeführt. Jeder dieser Befehle verfügt über ein PCF-Äquivalent, und die meisten sind auch über den IBM MQ Explorer verfügbar. Wenn diese Befehle in einem kleinen Netz ohne Workload ausgeführt werden, werden diese Befehle in der Regel innerhalb weniger Sekunden abgeschlossen. Dies ist jedoch bei größeren und stärkeren Netzen nicht der Fall. Darüber hinaus kann der Befehl **REFRESH CLUSTER** sehr viel länger dauern, insbesondere wenn er gleichzeitig auf mehreren Warteschlangenmanagern abgesetzt wird.

Überprüfen Sie, ob die erwarteten Objekte auf den fernen Warteschlangenmanagern vorhanden sind, um das Vertrauen zu haben, dass diese Befehle beendet sind.

Prozedur

- [ALTER QMGR](#)

Verwenden Sie für den Befehl `ALTER QMGR PARENT DISPLAY PUBSUB TYPE(PARENT) ALL`, um den Status der angeforderten übergeordneten Beziehung zu verfolgen.

Verwenden Sie für die Befehle `ALTER QMGR REPOS` und `ALTER QMGR REPOSNL` die Option `DISPLAY CLUSQMGR QMTYPE`, um die Fertigstellung zu bestätigen.

- [DEFINE CHANNEL](#), [ALTER CHANNEL](#) und [DELETE CHANNEL](#)

Verwenden Sie für alle Parameter, die in der Tabelle [ALTER CHANNEL parameters](#) aufgeführt sind, den Befehl `DISPLAY CLUSQMGR`, um zu überwachen, wann Änderungen an den Cluster weitergegeben wurden.

- [DEFINE NAMELIST](#), [ALTER NAMELIST](#) und [DELETE NAMELIST](#).

Wenn Sie ein **NAMELIST** für das Attribut **CLUSNL** eines **QMGR**-Objekts verwenden, kann sich eine Warteschlange oder ein Clusterkanal auf dieses Objekt auswirken. Überwachen Sie das betroffene Objekt nach Bedarf.

Änderungen an `SYSTEM.QPUBSUB.QUEUE.NAMELIST` können die Erstellung oder den Abbruch von Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie beeinflussen. Verwenden Sie den Befehl `DISPLAY SUB SUBTYPE(PROXY)`, um dies zu überwachen.

- [DEFINE Queues](#), [ALTER queues](#) und [DELETE queues](#).

Verwenden Sie für alle Parameter, die in der Tabelle Parameter, die vom Befehl DISPLAY QUEUE zurückgegeben werden können, den Befehl DISPLAY QCLUSTER, um zu überwachen, wann Änderungen an den Cluster weitergegeben wurden.

- DEFINE SUB und DELETE SUB

Wenn Sie die erste Subskription für eine Themenzeichenfolge definieren, können Sie Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster erstellen. Wenn Sie die letzte Subskription für eine Themenzeichenfolge löschen, können Sie außerdem Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster abbrechen.

Um zu überprüfen, ob ein Befehl zum Definieren oder Löschen einer Subskription abgeschlossen ist, überprüfen Sie, ob die erwartete Proxy-Subskription auf anderen Warteschlangenmanagern im verteilten Netz vorhanden ist oder nicht. Wenn Sie *direktes Routing* in einem Cluster verwenden, überprüfen Sie, ob die erwartete Proxy-Subskription in den anderen Teilrepositorys im Cluster vorhanden ist. Wenn Sie *Topic-Host-Routing* in einem Cluster verwenden, überprüfen Sie, ob die erwartete Proxy-Subskription auf den übereinstimmenden Themenhosts vorhanden ist. Verwenden Sie den folgenden MQSC-Befehl:

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

Verwenden Sie dieselbe Prüfung für die folgenden äquivalenten Subskriptions- und Abmelde- MQI-Aufrufe, wenn sie in einem Cluster oder einer Hierarchie ausgegeben werden:

- Subskribieren Sie über MQSUB.
- Sie können die Subskription mithilfe von MQCLOSE mit MQCO_REMOVE_SUB aufheben.

- DEFINE TOPIC, ALTER TOPIC und DELETE TOPIC

Wenn Sie überprüfen möchten, ob ein Befehl zum Definieren, Ändern oder Löschen eines Clusterthemas abgeschlossen ist, zeigen Sie das Thema in den anderen Teilrepositorys im Cluster an (wenn Sie *direktes Routing* verwenden) oder auf den anderen Themenhosts (wenn Sie *Topic-Host-Routing* verwenden).

Verwenden Sie für alle Parameter, die in der Tabelle Parameter, die vom Befehl DISPLAY TOPIC zurückgegeben werden können, den Befehl DISPLAY TCLUSTER, um zu überwachen, wann Änderungen an den Cluster weitergegeben wurden.

Anmerkung:

- Der Parameter **CLUSTER** kann die Erstellung oder den Abbruch von Proxy-Subskriptionen in einem Publish/Subscribe-Cluster beeinflussen.
- Die Parameter **PROXYSUB** und **SUBSCOPE** können die Erstellung oder den Abbruch von Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster beeinflussen.
- Verwenden Sie den Befehl DISPLAY SUB SUBTYPE(PROXYSUB), um dies zu überwachen.

- REFRESH CLUSTER

Wenn Sie den Befehl **REFRESH CLUSTER** ausführen, wählen Sie die Länge der Clusterbefehlwarteschlange aus. Warten Sie, bis der Wert null erreicht ist, und bleiben Sie bei null, bevor Sie nach den Objekten suchen.

1. Verwenden Sie den folgenden MQSC-Befehl, um zu überprüfen, ob die Länge der Clusterbefehlwarteschlange null ist.

```
DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
```

2. Wiederholen Sie die Überprüfung, bis die Warteschlangenlänge null erreicht, und bleibt bei der nachfolgenden Überprüfung auf Null.

Mit dem Befehl **REFRESH CLUSTER** werden Objekte entfernt und erneut erstellt, und in großen Konfigurationen kann eine erhebliche Zeit in Anspruch genommen werden. Weitere Informationen zu Publish/Subscribe-Clustern finden Sie im Abschnitt [Hinweise zu REFRESH CLUSTER](#).

- [REFRESH QMGR TYPE \(PROXYSUB\)](#)

Um zu überprüfen, ob der Befehl **REFRESH QMGR TYPE (PROXYSUB)** ausgeführt wurde, überprüfen Sie, ob die Proxy-Subskriptionen in anderen Warteschlangenmanagern im verteilten Netz korrigiert wurden. Wenn Sie *direktes Routing* in einem Cluster verwenden, überprüfen Sie, ob die Proxy-Subskriptionen in den anderen Teilrepositorys im Cluster korrigiert wurden. Wenn Sie *Topic-Host-Routing* in einem Cluster verwenden, überprüfen Sie, ob die erwarteten Proxy-Subskriptionen auf den übereinstimmenden Themenhosts korrigiert wurden. Verwenden Sie den folgenden MQSC-Befehl:

```
DISPLAY SUB(*) SUBTYPE(PROXYSUB)
```

- [RESET CLUSTER](#)

Verwenden Sie `DISPLAY CLUSQMGR`, um zu überprüfen, ob der Befehl **RESET CLUSTER** ausgeführt wurde.

- [RESET QMGR TYPE \(PUBSUB\)](#)

Verwenden Sie `DISPLAY PUBSUB TYPE(PARENT|CHILD)`, um zu prüfen, ob der Befehl **RESET QMGR** ausgeführt wurde.

Anmerkung: Der Befehl **RESET QMGR** kann dazu führen, dass Proxy-Subskriptionen in einer Publish/Subscribe-Hierarchie oder einem Publish/Subscribe-Cluster storniert werden. Verwenden Sie den Befehl `DISPLAY SUB SUBTYPE(PROXYSUB)`, um dies zu überwachen.

- Sie können auch andere Systemwarteschlangen überwachen, die, wenn Befehle abgeschlossen sind, dazu neigen, eine Warteschlangenlänge von null zu verwenden.

Sie können z. B. die `SYSTEM.INTER.QMGR.CONTROL`-Warteschlange und die `SYSTEM.INTER.QMGR.FANREQ`-Warteschlange überwachen. Weitere Informationen finden Sie unter [Proxy-Subskriptionsdatenverkehr in Clustern überwachen und Erzeuger und Konsumenten in Publish/Subscribe-Netzen abgleichen](#).

Nächste Schritte

Wenn diese Prüfungen nicht bestätigen, dass ein asynchroner Befehl beendet wurde, ist möglicherweise ein Fehler aufgetreten. Um zu untersuchen, überprüfen Sie zuerst das Protokoll für den Warteschlangenmanager, auf dem der Befehl abgesetzt wurde, und überprüfen Sie dann (für einen Cluster) die vollständigen Repository-Protokolle des Clusters.

Zugehörige Informationen

 [Asynchrones Verhalten von Clusterbefehlen unter z/OS](#)

Ferne Warteschlangendefinitionen als Aliasnamen verwenden

Zusätzlich zum Lokalisieren einer Warteschlange in einem anderen Warteschlangenmanager können Sie auch eine lokale Definition einer fernen Warteschlange für WS-Manager-Aliasnamen und Aliasnamen für Antwortwarteschlangen verwenden. Beide Typen von Aliasnamen werden durch die lokale Definition einer fernen Warteschlange aufgelöst. Sie müssen die entsprechenden Kanäle für die Nachricht einrichten, die an ihrem Ziel ankommen soll.

WS-Manager-Aliasnamen

Ein Aliasname ist der Prozess, durch den der Name des Zielwarteschlangenmanagers, wie in einer Nachricht angegeben, von einem WS-Manager auf dem Nachrichtenweg geändert wird. WS-Manager-Aliasnamen sind wichtig, da Sie sie verwenden können, um das Ziel von Nachrichten in einem Netz von Warteschlangenmanagern zu steuern.

Dies tun Sie, indem Sie die Definition der fernen Warteschlange auf dem Warteschlangenmanager am Steuerungspunkt ändern. Der sendenden Anwendung ist nicht bekannt, dass der angegebene Name des WS-Managers ein Aliasname ist.

Weitere Informationen zu Warteschlangen-Manager-Aliasnamen finden Sie unter [Was sind Aliasnamen?](#).

Antwortwarteschlangenaliasnamen

Optional kann eine Anwendung den Namen einer Warteschlange für Antwortnachrichten angeben, wenn sie eine *Anforderungsnachricht* in eine Warteschlange einreicht.

Wenn die Anwendung, die die Nachricht verarbeitet, den Namen der Warteschlange für Antwortnachrichten extrahiert, weiß sie, wohin die *Antwortnachricht* gesendet werden soll, falls erforderlich.

Ein Aliasname der Empfangswarteschlange für Antworten ist der Prozess, mit dem eine Empfangswarteschlange für Antworten, wie in einer Anforderungsnachricht angegeben, von einem WS-Manager auf dem Nachrichtenweg geändert wird. Der sendenden Anwendung ist nicht bekannt, dass der angegebene Antwortwarteschlangenname ein Aliasname ist.

Mit einem Aliasnamen für die Antwortwarteschlange können Sie den Namen der Empfangswarteschlange für Antworten und optional dessen Warteschlangenmanager ändern. Über diese wiederum können Sie steuern, welche Route für Antwortnachrichten verwendet wird.

Weitere Informationen zu Anforderungsnachrichten, Antwortnachrichten und Antwortwarteschlangen finden Sie unter [Nachrichtenarten und Empfangswarteschlange für Antworten und Warteschlangenmanager](#).

Weitere Informationen zu den Aliasnamen für Antwortwarteschlangen finden Sie im Abschnitt [Aliasnamen und Cluster für Antwortwarteschlangen](#).

Datenkonvertierung

Nachrichtendaten in IBM MQ-definierten Formaten (auch bekannt als integrierte Formate) können vom Warteschlangenmanager von einem codierten Zeichensatz in einen anderen konvertiert werden, sofern sich beide Zeichensätze auf eine einzige Sprache oder eine Gruppe ähnlicher Sprachen beziehen.

Beispielsweise wird die Konvertierung zwischen codierten Zeichensätzen mit Kennungen (CCSIDs) 850 und 500 unterstützt, da beide auf westeuropäisch-sprachige Sprachen anwendbar sind.

Für EBCDIC-Zeilenumsetzungen in ASCII (NL) siehe [Alle WS-Manager](#).

Unterstützte Konvertierungen werden in der [Datenkonvertierungsverarbeitung](#) definiert.

Wenn ein WS-Manager Nachrichten in integrierten Formaten nicht konvertieren kann

Der WS-Manager kann Nachrichten nicht automatisch in integrierte Formate konvertieren, wenn ihre CCSIDs unterschiedliche Landessprachengruppen darstellen. Beispielsweise wird die Konvertierung zwischen CCSID 850 und CCSID 1025 (die ein EBCDIC-codierter Zeichensatz für Sprachen mit kyrillischem Script ist) nicht unterstützt, da viele der Zeichen in einem codierten Zeichensatz nicht in der anderen dargestellt werden können. Wenn Sie über ein Netz von Warteschlangenmanagern in verschiedenen Landessprachen verfügen und die Datenkonvertierung unter einigen der codierten Zeichensätze nicht unterstützt wird, können Sie eine Standardkonvertierung aktivieren.

V 9.0.0 Weitere Informationen zu Plattformen, für die `ccsid_part2.tbl` gilt, finden Sie unter „Standarddatenkonvertierung angeben“ auf Seite 218 [Verwendung ccsid_part2.tbl](#). Die Standarddatenkonvertierung auf anderen Plattformen als denen, auf die die Datei `ccsid_part2.tbl` angewendet wird, wird in „Standarddatenkonvertierung“ auf Seite 216 beschrieben.

Verbesserte Unterstützung für Unicode-Datenkonvertierung in IBM MQ 9.0

V 9.0.0

Vor IBM MQ 9.0 haben frühere Versionen des Produkts keine Konvertierung von Daten unterstützt, die Unicode-Codepunkte über die Basic Multilingual Plane (Codepunkte über U+FFFF) hinaus enthalten. Die Unterstützung für Unicode-Datenkonvertierung war auf Codepunkte beschränkt, die im Unicode 3.0-Standard definiert sind und entweder in UTF-8 oder UCS-2 codiert sind, einem 2 Byte langen Subset mit fester Breite von UTF-16.

Ab IBM MQ 9.0 unterstützt IBM MQ alle Unicode-Zeichen, die in der Norm Unicode 8.0 in der Datenkonvertierung definiert sind. Dies schließt die vollständige Unterstützung für UTF-16 ein, einschließlich Ersatzpaare (ein Paar aus 2-Byte-UTF-16-Zeichen im Bereich von X ' D800 ' bis zu X ' DFFF ' , die einen Unicode-Codepunkt oberhalb von U+FFFF darstellen).

Das Kombinieren von Zeichenfolgen wird auch in Fällen unterstützt, in denen ein vorkomponiertes Zeichen in einer CCSID einer Kombination aus einer Kombination von Zeichen in einer anderen CCSID zugeordnet ist.

Die Datenkonvertierung in und aus Unicode- und CCSIDs 1388, 1390, 1399, 4933, 5488 und 16884 wurde auf einigen Plattformen erweitert, um alle derzeit für diese CCSIDs definierten Codepunkte zu unterstützen, einschließlich der Codepunkte, die Codepunkten in Unicode-Zusatzebenen zugeordnet sind.

Im Falle der CCSIDs 1390, 1399 und 16884 enthält dies Zeichen, die im JIS-Standard X 0213 (JIS2004) definiert sind.

Es wurde auch Unterstützung für die Konvertierung in und aus Unicode und sechs neue CCSIDs (1374 bis 1379) hinzugefügt.

ccsid_part2.tbl (Datei)

V 9.0.0

Ab IBM MQ 9.0 wird eine zusätzliche Datei, `ccsid_part2.tbl`, bereitgestellt.

Die Datei `ccsid_part2.tbl` hat Vorrang vor der Datei `ccsid.tbl` und ermöglicht Ihnen Folgendes:

- Hinzufügung oder Änderung von CCSID-Einträgen
- Angabe einer Standarddatenkonvertierung
- Angabe von Daten für verschiedene Befehlsebenen

Die Datei `ccsid_part2.tbl` ist nur für folgende Plattformen gültig:

- **Linux** Linux - Alle Versionen
- **Solaris** Solaris
- **Windows** Windows

Windows Ab IBM MQ 9.0 befindet sich `ccsid_part2.tbl` unter IBM MQ for Windows standardmäßig im Verzeichnis `MQDataRoot\conv\table`. Darüber hinaus werden in IBM MQ for Windows alle unterstützten codierten Zeichensätze aufgezeichnet.

Solaris **Linux** Ab IBM MQ 9.0 befindet sich `ccsid_part2.tbl` auf IBM MQ for Linux - und Solaris -Plattformen im Verzeichnis `MQDataRoot/conv/table`. Für alle Linux- und Solaris-Plattformen werden die unterstützten codierten Zeichensätze in Konvertierungstabellen aufbewahrt, die von IBM MQ bereitgestellt werden.

Obwohl die Datei `ccsid_part2.tbl` die vorhandene Datei `ccsid.tbl` ersetzt, die in früheren Versionen von IBM MQ verwendet wurde, um zusätzliche CCSID-Informationen bereitzustellen, wird die Datei `ccsid.tbl` weiterhin von IBM MQ geparkt und darf daher nicht gelöscht werden.

Weitere Informationen finden Sie unter „[Die Datei ccsid_part2.tbl](#)“ auf Seite 217.

ccsid.tbl -Datei

V 9.0.0 Auf anderen Plattformen als denen, auf die ccsid_part2.tbl angewendet wird, wird die Datei ccsid.tbl für die folgenden Zwecke verwendet:

- **HP-UX** **AIX** Auf AIX- und HP-UX-Plattformen werden die unterstützten codierten Zeichensätze intern vom Betriebssystem gehalten.
- Sie gibt alle zusätzlichen codierten Zeichensätze an. Wenn Sie zusätzliche Codesensätzen angeben möchten, müssen Sie ccsid.tbl bearbeiten (Anleitung dazu, wie dies in der Datei ausgeführt wird).
- Sie gibt eine Standarddatenkonvertierung an.

Sie können die in ccsid.tbl aufgezeichneten Informationen aktualisieren. Dies kann beispielsweise der Fall sein, wenn ein zukünftiges Release des Betriebssystems zusätzliche codierte Zeichensätze unterstützt.

Standarddatenkonvertierung

V 9.0.0 Ab IBM MQ 9.0 hat sich die Methode der Standarddatenkonvertierung auf folgenden Plattformen geändert:

- Linux - Alle Versionen
- Solaris
- Windows

Weitere Informationen finden Sie unter „[Standarddatenkonvertierung angeben](#)“ auf [Seite 218](#) unter Verwendung von ccsid_part2.tbl .

Wenn Sie Kanäle zwischen zwei Maschinen einrichten, auf denen die Datenkonvertierung normalerweise nicht unterstützt wird, müssen Sie die Standarddatenkonvertierung für die Kanäle aktivieren, damit sie funktionieren.

V 9.0.0 Auf anderen Plattformen als denen, auf die ccsid_part2.tbl angewendet wird, um die Standarddatenkonvertierung zu aktivieren, bearbeiten Sie die Datei ccsid.tbl , um eine EBCDIC-Standard-CCSID und eine Standard-ASCII-CCSID anzugeben. Anweisungen dazu, wie Sie dies tun, sind in der Datei enthalten. Sie müssen dies auf allen Maschinen tun, die über die Kanäle verbunden werden. Starten Sie den WS-Manager erneut, damit die Änderung wirksam wird.

Der Standarddatenkonvertierungsprozess ist wie folgt:

- Wenn die Konvertierung zwischen der Quellen- und der Ziel-CCSIDs nicht unterstützt wird, die CCSIDs der Quellen- und Zielumgebungen jedoch entweder EBCDIC oder beide ASCII sind, werden die Zeichen- und Daten ohne Konvertierung an die Zielanwendung übergeben.
- Wenn eine CCSID einen codierten ASCII-Zeichensatz darstellt und die andere einen EBCDIC-codierten Zeichensatz darstellt, konvertiert IBM MQ die Daten unter Verwendung der in ccsid.tbl definierten Standard-CCSIDs für die Datenkonvertierung.

Anmerkung: Versuchen Sie, die zu konvertierenden Zeichen auf diejenigen zu beschränken, die die gleichen Codewerte in dem codierten Zeichensatz haben, der für die Nachricht und den codierten Standardzeichensatz angegeben ist. Wenn Sie nur die Gruppe von Zeichen verwenden, die für IBM MQ -Objektnamen gültig sind (wie in [IBM MQ -Objekte benennen](#) definiert) Sie werden diese Anforderung im Allgemeinen erfüllen. Es treten Ausnahmeregelungen mit EBCDIC-CCSIDs 290, 930, 1279 und 5026 auf, die in Japan verwendet werden, wobei die Kleinbuchstaben andere Codes als die in anderen EBCDIC-CCSIDs verwendeten Zeichen haben.

Konvertieren von Nachrichten in benutzerdefinierte Formate

Der WS-Manager kann keine Nachrichten in benutzerdefinierten Formaten von einem codierten Zeichensatz in einen anderen konvertieren. Wenn Sie Daten in ein benutzerdefiniertes Format konvertieren müssen, müssen Sie für jedes dieser Formate einen Datenkonvertierungscode angeben. Verwenden Sie keine

Standard-CCSIDs, um Zeichendaten in benutzerdefinierte Formate zu konvertieren. Weitere Informationen zum Konvertieren von Daten in benutzerdefinierte Formate und zum Schreiben von Datenkonvertierungsexits finden Sie im Abschnitt [Datenkonvertierungsexits schreiben](#).

CCSID des Warteschlangenmanagers ändern

Wenn Sie das Attribut CCSID des Befehls ALTER QMGR verwendet haben, um die CCSID des Warteschlangenmanagers zu ändern, stoppen Sie den Warteschlangenmanager und starten Sie ihn erneut, um sicherzustellen, dass alle aktiven Anwendungen, einschließlich des Befehlsservers und des Kanalprogramms, gestoppt und erneut gestartet werden.

Dies ist erforderlich, da alle Anwendungen, die ausgeführt werden, wenn die CCSID des Warteschlangenmanagers geändert wird, weiterhin die vorhandene CCSID verwenden.

Die Datei `ccsid_part2.tbl`

Ab IBM MQ 9.0 ersetzt die Datei `ccsid_part2.tbl` die in vorherigen Versionen des Produkts vorhandene Datei `ccsid.tbl`, damit weitere CCSID-Informationen bereitgestellt werden. Beachten Sie, dass die Datei `ccsid.tbl` weiterhin von IBM MQ syntaktisch analysiert wird und deshalb nicht gelöscht werden sollte und dass die Einträge in `ccsid_part2.tbl` Vorrang vor anderen Einträgen in `ccsid.tbl` haben.

`ccsid_part2.tbl` befindet sich im Verzeichnis `MQDataRoot/conv/table` unter:

- Linux - Alle Versionen
- Solaris

und im Verzeichnis `MQDataRoot\conv\table` unter Windows

Beachten Sie, dass die Datei `ccsid_part2.tbl` auf keiner anderen Plattform verfügbar ist. Die Datei wird verwendet, um die folgenden Aktionen auszuführen:

- Vorhandene CCSID-Einträge für die Verwendung in der IBM MQ-Datenkonvertierung hinzufügen oder ändern
- Standarddatenkonvertierung angeben

Sie sollten `ccsid_part2.tbl` als folgende Datei verwenden:

- Enthält Unterstützung für die neuen Unicode-Codierungswerte.
- Hier können Sie die Version der CCSID-Einträge angeben, so dass die Einträge nur auf ausgewählte Befehlsebenen angewendet werden können.

CCSID-Einträge hinzufügen oder ändern

Ein Eintrag in der Datei `ccsid_part2.tbl` hat das folgende Format:

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

Ein Beispieleintrag für CCSID 1200 (UTF-16) lautet wie folgt:

```
1200 1200 1200 1200 3 8 0 UTF-16
```

Anmerkung: Weitere Informationen zum Wert für ACRI enthält der Kommentar in der Datei `ccsid_part2.tbl`.

Im `ccsid_part2.tbl`-Format:

Typ kann gleich sein:

- 1=SBCS
- 2=DBCS
- 3=MBCS

Codierung kann gleich sein:

- 1=EBCDIC
- 2 = ASCII
- 3 = ISO
- 4 = UCS-2
- 5 = UTF-8
- 6 = Euc
- 7 = GB18030
- 8 = UTF-16
- 9 = UTF-32

Beim Bearbeiten der Datei:

- Sie können einen Kommentar angeben, indem Sie das Symbol # am Anfang einer Zeile verwenden. Dadurch wird verhindert, dass IBM MQ versucht, Zeile zu parsen.
- In-line-Kommentare können nicht angegeben werden.
- Es muss sichergestellt werden, dass keine Leerzeilen erstellt werden.
- Am Ende der Datei dürfen keine neuen Einträge hinzugefügt werden.
Neue CCSID-Einträge sollten vor den ACRI-Tabelleninformationen hinzugefügt werden.

Standarddatenkonvertierung angeben

Sie können CCSIDs für die Standardkonvertierung definieren, die verwendet werden, um zwischen ASCII- oder ähnlichen und EBCDIC-CCSIDs zu konvertieren, wenn keine Konvertierung zwischen zwei CCSIDs unterstützt wird.

Wenn Sie diese Funktion aktivieren, wird die Standardkonvertierung für Übertragungs- und Nachrichtenheader verwendet und kann auch bei der Benutzerdatenkonvertierung verwendet werden.

Standardkonvertierungen werden aktiviert, indem zwei Zeilen ähnlich der folgenden erstellt werden:

```
default      0      500      1      1      0
default      0      850      1      2      0
```

Die erste Zeile setzt die Standardeinstellung für EBCDIC-CCSIDs auf 500 und die zweite Zeile setzt den Standardwert für ASCII und ähnliche CCSIDs auf 850.

Daten für verschiedene Befehlsebenen angeben

Um CCSID-Einträge für verschiedene Befehlsebenen von IBM MQ anzugeben, verwenden Sie einen Doppelpunkt, gefolgt von der Befehlsebene (oder den Befehlsebenen) von IBM MQ, auf die der nächste Abschnitt angewendet werden soll.

Die Zahl stellt die Mindestbefehlsstufe dar, in der der WS-Manager oder Client ausgeführt werden muss. Wenn der aktuelle Warteschlangenmanager beispielsweise die Befehlsebene 900 hat und ein Flag für die Befehlsebene 800 oder 900 erkannt wird, werden die CCSIDs gelesen.

Ein WS-Manager auf Ebene 800 ignoriert jedoch alle CCSIDs im Abschnitt 900.

Die angegebene Befehlsebene gilt für alle CCSID-Einträge, die nach einer Markierung auf Befehlsebene gefunden werden, bis eine neue Markierung auf Befehlsebene gefunden wird.

Wenn Sie die Befehlsebene auf alle Befehlsebenen setzen müssen, geben Sie die Zahl null an.

Beim ersten Parsing von `ccsid_part2.tbl` behandelt IBM MQ alle gefundenen CCSIDs als gültig für alle Befehlsebenen von IBM MQ.

Die Verwendung der Versionssteuerung beginnt erst, wenn IBM MQ das erste Flag auf Befehlsebene findet.

Das folgende Code-Snippet zeigt ein Beispiel für die Verwendung der Versionssteuerung:

```
# Comment Block
# End of Comment Block
# Because no command level flag is specified and we're at the start of the file
# the following CCSIDs will be read on all versions
  819  819    0    819    1    3    0    IS08859-1
  923  923    0    923    1    3    0    IS08859-15
 1051 1051    0   1051    1    3    0    IBM-1051
# The colon :900 below shows that the CCSIDs after will only be for MQ cmd level 900 and above
:900
  8629 437    0    437    1    2    0    IBM-437
 12725 437    0    437    1    2    0    IBM-437
 16821 437    0    437    1    2    0    IBM-437
 20917 437    0    437    1    2    0    IBM-437
# The colon :0 below shows that the CCSIDs after will be for all version of MQ
:0
  4946 850    0    850    1    2    0    IBM-850
 33618 850    0    850    1    2    0    IBM-850
 61697 850    0    850    1    2    0    IBM-850
 61698 850    0    850    1    2    0    IBM-850
```

Windows

Linux

AIX

MQ Telemetry verwalten

MQ Telemetry wird mit IBM MQ Explorer oder über eine Befehlszeile verwaltet. Verwenden Sie den Explorer, um Telemetriekanäle zu konfigurieren, den Telemetrieservice zu steuern und die MQTT-Clients zu überwachen, die mit IBM MQ verbunden sind. Konfigurieren Sie die Sicherheit von MQ Telemetry mithilfe von JAAS, TLS und des Objektberechtigungsmanagers von IBM MQ.

Verwaltung mit IBM MQ Explorer

Verwenden Sie den Explorer, um Telemetriekanäle zu konfigurieren, den Telemetrieservice zu steuern und die MQTT-Clients zu überwachen, die mit IBM MQ verbunden sind. Konfigurieren Sie die Sicherheit von MQ Telemetry mithilfe von JAAS, TLS und des Objektberechtigungsmanagers von IBM MQ.

Verwaltung über die Befehlszeile

MQ Telemetry kann vollständig über die Befehlszeile mithilfe der IBM MQMQSC-Befehle verwaltet werden.

Die MQ Telemetry-Dokumentation enthält auch Beispielskripts, die die grundlegende Verwendung der IBM MQ Telemetry Transport v3-Clientanwendung veranschaulichen.

Lesen und verstehen Sie die Beispiele in [IBM MQ Telemetry Transport-Beispielprogramme](#), bevor Sie sie verwenden.

Zugehörige Informationen

[MQ Telemetry](#)

[MQXR-Eigenschaften](#)

Linux

AIX

Warteschlangenmanager unter Linux und AIX für Telemetrie konfigurieren

Führen Sie diese Schritte aus, um einen Warteschlangenmanager manuell für die Ausführung von MQ Telemetry zu konfigurieren. Sie können eine automatisierte Prozedur ausführen, um mithilfe der MQ Telemetry-Unterstützung für IBM MQ Explorer eine einfachere Konfiguration einzurichten.

Vorbereitende Schritte

1. Siehe [Installieren MQ Telemetry](#) für Informationen zum Installieren IBM MQ, und zum MQ Telemetry Feature.
2. Erstellen und starten Sie einen WS-Manager. Der WS-Manager wird in dieser Task als *qMgr* bezeichnet.
3. Im Rahmen dieser Task konfigurieren Sie den Telemetrieservice (MQXR). Die MQXR-Eigenschaftseinstellungen werden in einer plattformspezifischen Eigenschaftendatei gespeichert: `mqxr_unix.pro`

properties. Normalerweise müssen Sie die MQXR-Eigenschaftendatei nicht direkt bearbeiten, da fast alle Einstellungen über MQSC-Verwaltungsbefehle oder mit IBM MQ Explorer konfiguriert werden können. Wenn Sie die Datei direkt bearbeiten möchten, stoppen Sie den Warteschlangenmanager, bevor Sie Ihre Änderungen vornehmen. Siehe [MQXR-Eigenschaften](#).

Informationen zu diesem Vorgang

Die MQ Telemetry -Unterstützung für IBM MQ Explorer enthält einen Assistenten und eine Beispielbefehlsprozedur `sampleMQM`. Diese richten eine Erstkonfiguration mit der Gastbenutzer-ID ein (siehe [Installation von MQ Telemetry mithilfe von IBM MQ Explorer und IBM MQ Telemetry Transport-Beispielprogrammen überprüfen](#)).

Führen Sie die Schritte in dieser Task aus, um MQ Telemetry manuell mit verschiedenen Berechtigungsschemas zu konfigurieren.

Vorgehensweise

1. Öffnen Sie ein Befehlsfenster auf dem Telemetriemusterverzeichnis.

Das Telemetry-Beispielverzeichnis ist `/opt/mqm/mqxr/samples`.

2. Erstellen Sie die Übertragungswarteschlange für Telemetrie.

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

Wenn der Telemetrieservice (MQXR) zuerst gestartet wird, wird `SYSTEM.MQTT.TRANSMIT.QUEUE` erstellt.

Sie wird in dieser Task manuell erstellt, da `SYSTEM.MQTT.TRANSMIT.QUEUE` vorhanden sein muss, bevor der Telemetrieservice (MQXR) gestartet wird, um den Zugriff auf sie zu autorisieren.

3. Standardübertragungswarteschlange festlegen

Wenn der Telemetrieservice (MQXR) zum ersten gestartet wird, ändert er den Warteschlangenmanager nicht, um `SYSTEM.MQTT.TRANSMIT.QUEUE` zur Standardübertragungswarteschlange zu machen.

Um `SYSTEM.MQTT.TRANSMIT.QUEUE` zu der Standardübertragungswarteschlange zu machen, ändern Sie die Eigenschaft der Standardübertragungswarteschlange. Ändern Sie die Eigenschaft mit IBM MQ Explorer oder mit dem Befehl im folgenden Beispiel:

```
echo "ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE')" | runmqsc qMgr
```

Das Ändern der Standardübertragungswarteschlange kann die vorhandene Konfiguration beeinträchtigen. Der Grund für die Änderung der Standardübertragungswarteschlange in `SYSTEM.MQTT.TRANSMIT.QUEUE` ist, das Senden von Nachrichten direkt an MQTT-Clients zu erleichtern. Ohne Änderung der Standardübertragungswarteschlange müssen Sie für jeden Client, der IBM MQ Explorer-Nachrichten empfängt, eine Definition einer fernen Warteschlange hinzufügen (siehe [„Eine Nachricht direkt an einen Client senden“](#) auf Seite 225).

4. Führen Sie eine Prozedur in [„MQTT-Clients für den Zugriff auf IBM MQ-Objekte berechtigen“](#) auf Seite 227 aus, um eine oder mehrere Benutzer-IDs zu erstellen. Die Benutzer-IDs verfügen über die Berechtigung zum Veröffentlichen, Subskribieren und Senden von Veröffentlichungen an MQTT-Clients.
5. Installieren Sie den Telemetrieservice (MQXR).

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc qMgr
```

Siehe auch den Beispielcode in [Abbildung 20](#) auf Seite 221.

6. Starten Sie den Service.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE)" | runmqsc qMgr
```

Der Telemetrieservice (MQXR) wird automatisch gestartet, wenn der WS-Manager gestartet wird.

Diese Task wird manuell in dieser Task gestartet, da der Warteschlangenmanager bereits ausgeführt wird.

7. Konfigurieren Sie mithilfe von IBM MQ Explorer Telemetriedienste, sodass sie Verbindungen von MQTT-Clients akzeptieren.

Die Telemetriedienste müssen so konfiguriert sein, dass ihre Identitäten eine der in Schritt 4 definierten Benutzer-IDs sind.

Siehe auch [DEFINE CHANNEL \(MQTT\)](#).

8. Überprüfen Sie die Konfiguration, indem Sie den Beispielclient ausführen.

Damit der Beispielclient mit Ihrem Telemetriedienst arbeiten kann, muss der Kanal den Client für die Veröffentlichung, Subskription und Empfang von Veröffentlichungen berechtigen. Der Beispielclient stellt die Verbindung zum Telemetriedienst an Port 1883 standardmäßig her. Siehe auch [IBM MQ Telemetry Transport-Beispielprogramme](#).

Beispiel

Abbildung 20 auf Seite 221 zeigt den Befehl `runmqsc` zum manuellen Erstellen der `SYSTEM.MQXR.SERVICE` unter Linux.

```
DEF      SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

Abbildung 20. `installMQXRService_unix.mqsc`

Windows Warteschlangenmanager unter Windows für Telemetrie konfigurieren

Führen Sie diese Schritte aus, um einen Warteschlangenmanager manuell für die Ausführung von MQ Telemetry zu konfigurieren. Sie können eine automatisierte Prozedur ausführen, um mithilfe der MQ Telemetry-Unterstützung für IBM MQ Explorer eine einfachere Konfiguration einzurichten.

Vorbereitende Schritte

1. Siehe [Installieren MQ Telemetry](#) für Informationen zum Installieren IBM MQ, und zum MQ Telemetry Feature.
2. Erstellen und starten Sie einen WS-Manager. Der WS-Manager wird in dieser Task als `qMgr` bezeichnet.
3. Im Rahmen dieser Task konfigurieren Sie den Telemetrieservice (MQXR). Die MQXR-Eigenschaftseinstellungen werden in einer plattformspezifischen Eigenschaftendatei gespeichert: `mqxr_win.properties`. Normalerweise müssen Sie die MQXR-Eigenschaftendatei nicht direkt bearbeiten, da fast alle Einstellungen über MQSC-Verwaltungsbefehle oder mit IBM MQ Explorer konfiguriert werden können. Wenn Sie die Datei direkt bearbeiten möchten, stoppen Sie den Warteschlangenmanager, bevor Sie Ihre Änderungen vornehmen. Siehe [MQXR-Eigenschaften](#).

Informationen zu diesem Vorgang

Die MQ Telemetry -Unterstützung für IBM MQ Explorer enthält einen Assistenten und eine Beispielbefehlsprozedur `sampleMQM`. Diese richten eine Erstkonfiguration mit der Gastbenutzer-ID ein (siehe [Installation von MQ Telemetry mithilfe von IBM MQ Explorer und IBM MQ Telemetry Transport-Beispielprogrammen überprüfen](#)).

Führen Sie die Schritte in dieser Task aus, um MQ Telemetry manuell mit verschiedenen Berechtigungsschemas zu konfigurieren.

Vorgehensweise

1. Öffnen Sie ein Befehlsfenster auf dem Telemetriemusterverzeichnis.

Das Telemetry-Beispielverzeichnis ist `WMQ program installation directory\mqxr\samples`.

2. Erstellen Sie die Übertragungswarteschlange für Telemetrie.

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

Wenn der Telemetrieservice (MQXR) zuerst gestartet wird, wird `SYSTEM.MQTT.TRANSMIT.QUEUE` erstellt.

Sie wird in dieser Task manuell erstellt, da `SYSTEM.MQTT.TRANSMIT.QUEUE` vorhanden sein muss, bevor der Telemetrieservice (MQXR) gestartet wird, um den Zugriff auf sie zu autorisieren.

3. Standardübertragungswarteschlange für `qMgr` festlegen

```
echo ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

Abbildung 21. Standardübertragungswarteschlange festlegen

Wenn der Telemetrieservice (MQXR) zum ersten gestartet wird, ändert er den Warteschlangenmanager nicht, um `SYSTEM.MQTT.TRANSMIT.QUEUE` zur Standardübertragungswarteschlange zu machen.

Um `SYSTEM.MQTT.TRANSMIT.QUEUE` zu der Standardübertragungswarteschlange zu machen, ändern Sie die Eigenschaft der Standardübertragungswarteschlange. Ändern Sie die Eigenschaft mit dem IBM MQ Explorer oder mit dem Befehl in [Abbildung 21 auf Seite 222](#).

Das Ändern der Standardübertragungswarteschlange kann die vorhandene Konfiguration beeinträchtigen. Der Grund für die Änderung der Standardübertragungswarteschlange in `SYSTEM.MQTT.TRANSMIT.QUEUE` ist, das Senden von Nachrichten direkt an MQTT-Clients zu erleichtern. Ohne Änderung der Standardübertragungswarteschlange müssen Sie für jeden Client, der IBM MQ-Nachrichten empfängt, eine Definition einer fernen Warteschlange hinzufügen (siehe [„Eine Nachricht direkt an einen Client senden“](#) auf Seite 225).

4. Führen Sie eine Prozedur in [„MQTT-Clients für den Zugriff auf IBM MQ-Objekte berechtigen“](#) auf Seite 227 aus, um eine oder mehrere Benutzer-IDs zu erstellen. Die Benutzer-IDs verfügen über die Berechtigung zum Veröffentlichen, Subskribieren und Senden von Veröffentlichungen an MQTT-Clients.
5. Installieren Sie den Telemetrieservice (MQXR).

```
type  
installMQXRService_win.mqsc | runmqsc qMgr
```

6. Starten Sie den Service.

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

Der Telemetrieservice (MQXR) wird automatisch gestartet, wenn der WS-Manager gestartet wird.

Diese Task wird manuell in dieser Task gestartet, da der Warteschlangenmanager bereits ausgeführt wird.

7. Konfigurieren Sie mithilfe von IBM MQ Explorer Telemetriedienste, sodass sie Verbindungen von MQTT-Clients akzeptieren.

Die Telemetriedienste müssen so konfiguriert sein, dass ihre Identitäten eine der in Schritt 4 definierten Benutzer-IDs sind.

Siehe auch [DEFINE CHANNEL \(MQTT\)](#).

8. Überprüfen Sie die Konfiguration, indem Sie den Beispielclient ausführen.

Damit der Beispielclient mit Ihrem Telemetriedienst arbeiten kann, muss der Kanal den Client für die Veröffentlichung, Subskription und Empfang von Veröffentlichungen berechtigen. Der Beispielclient stellt die Verbindung zum Telemetriedienst an Port 1883 standardmäßig her. Siehe auch [IBM MQ Telemetry Transport-Beispielprogramme](#).

SYSTEM.MQXR.SERVICE manuell erstellen

Abbildung 22 auf Seite 223 zeigt den Befehl `runmqsc` zum manuellen Erstellen der `SYSTEM.MQXR.SERVICE` unter Windows.

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+mqxr\bin\runMQXRService.bat') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\"') +
STOPCMD('+MQ_INSTALL_PATH+mqxr\bin\endMQXRService.bat') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+mqxr.stderr')
```

Abbildung 22. `installMQXRService_win.mqsc`

Windows

Linux

AIX

Verteilte Steuerung von Warteschlangen für Senden von Nachrichten an MQTT-Clients konfigurieren

IBM MQ -Anwendungen können Nachrichten von MQTT v3 -Clients senden, indem sie eine von einem Client erstellte Subskription veröffentlichen oder eine Nachricht direkt senden. Whichever-Methode wird verwendet, die Nachricht wird auf `SYSTEM.MQTT.TRANSMIT.QUEUE` gestellt und vom Telemetrieservice (MQXR) an den Client gesendet. Es gibt eine Reihe von Möglichkeiten, eine Nachricht in `SYSTEM.MQTT.TRANSMIT.QUEUE` zu platzieren.

Nachricht als Antwort auf eine MQTT-Clientsubskription senden

Der Telemetrieservice (MQXR) erstellt eine Subskription im Namen des MQTT-Clients. Der Client ist das Ziel für alle Veröffentlichungen, die mit der Subskription übereinstimmen, die vom Client gesendet wurde. Die Telemetrie-Services leiten übereinstimmende Veröffentlichungen wieder an den Client weiter.

Ein MQTT-Client ist als Warteschlangenmanager mit IBM MQ verbunden, wobei der Wert von `ClientIdentifier` als Warteschlangenmanagername angegeben ist. Die Zieladresse für Veröffentlichungen, die an den Client gesendet werden sollen, ist eine Übertragungswarteschlange, `SYSTEM.MQTT.TRANSMIT.QUEUE`. Der Telemetrieservice leitet Nachrichten in `SYSTEM.MQTT.TRANSMIT.QUEUE` an MQTT-Clients weiter, wobei der Name des Zielwarteschlangenmanagers als Schlüssel für einen bestimmten Client verwendet wird.

Der Telemetrieservice (MQXR) öffnet die Übertragungswarteschlange unter Verwendung von `ClientIdentifier` als Name des Warteschlangenmanagers. Der Telemetrieservice (MQXR) übergibt die Objektkennung der Warteschlange an den Aufruf `MQSUB`, um Veröffentlichungen weiterzuleiten, die der Clientsubskription entsprechen. Bei der Auflösung des Objektnamens wird `ClientIdentifier` als Name des fernen Warteschlangenmanagers erstellt und die Übertragungswarteschlange muss in `SYSTEM.MQTT.TRANSMIT.QUEUE` aufgelöst werden. Bei Verwendung der Standardauflösung für IBM MQ -Objektnamen wird `ClientIdentifier` wie folgt aufgelöst (siehe [Tabelle 14 auf Seite 224](#)).

1. `ClientIdentifier` stimmt mit nichts überein.

ClientIdentifier ist ein Name des fernen Warteschlangenmanagers. Er stimmt nicht mit dem Namen des lokalen Warteschlangenmanagers, einem WS-Manager-Aliasnamen oder einem Übertragungswarteschlangennamen überein.

Der Warteschlangenname ist nicht definiert. Derzeit setzt der Telemetrieservice (MQXR) SYSTEM.MQTT.PUBLICATION.QUEUE als Namen der Warteschlange. Da ein MQTT v3-Client keine Warteschlangen unterstützt, wird der aufgelöste Warteschlangenname vom Client ignoriert.

Die Eigenschaft des lokalen Warteschlangenmanagers, Standardübertragungswarteschlange, muss auf SYSTEM.MQTT.TRANSMIT.QUEUE gesetzt werden, damit die Veröffentlichung SYSTEM.MQTT.TRANSMIT.QUEUE an den Client gesendet werden kann.

2. *ClientIdentifier* stimmt mit einem WS-Manager-Aliasnamen mit dem Namen *ClientIdentifier* überein

ClientIdentifier ist ein Name des fernen Warteschlangenmanagers. Er stimmt mit dem Namen eines WS-Manager-Aliasnamens überein.

Der Aliasname des WS-Managers muss mit *ClientIdentifier* als Name des fernen Warteschlangenmanagers definiert werden.

Wenn Sie den Namen der Übertragungswarteschlange in der WS-Manager-Aliasdefinition festlegen, ist es nicht erforderlich, dass die Standardübertragung auf SYSTEM.MQTT.TRANSMIT.QUEUE gesetzt wird.

	Eingabe		Ausgabe		
	Name des Warteschlangenmanagers	Warteschlangenname	Name des Warteschlangenmanagers	Warteschlangenname	Übertragungswarteschlange
Stimmt mit nichts überein	<i>ClientIdentifier</i>	<i>undefined</i>	<i>ClientIdentifier</i>	<i>undefined</i>	Standardübertragungswarteschlange. SYSTEM.MQTT.TRANSMIT.QUEUE
Übereinstimmung mit einem WS-Manager-Aliasnamen mit dem Namen <i>ClientIdentifier</i>	<i>ClientIdentifier</i>	<i>undefined</i>	<i>ClientIdentifier</i>	<i>undefined</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Weitere Informationen zur Namensauflösung finden Sie in [Namensauflösung](#).

Jedes IBM MQ-Programm kann Veröffentlichungen für dasselbe Thema durchführen. Die Veröffentlichung wird an ihre Subskribenten gesendet, einschließlich der MQTT v3-Clients, die über eine Subskription für das Thema verfügen.

Wenn in einem Cluster ein Verwaltungsthema mit dem Attribut CLUSTER(*clusterName*) erstellt wird, kann jede beliebige Anwendung im Cluster auf dem Client veröffentlicht werden, z. B.:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

Abbildung 23. Clusterthema unter Windows definieren

Anmerkung: Geben Sie SYSTEM.MQTT.TRANSMIT.QUEUE kein Clusterattribut an.

MQTT-Clients können als Subskribenten oder Publisher eine Verbindung zu verschiedenen Warteschlangenmanagern herstellen. Subskribenten und Publisher können Teil desselben Clusters sein oder durch eine Publish/Subscribe-Hierarchie verbunden sein. Die Veröffentlichung wird über IBM MQ vom Publisher an den Subskribenten zugestellt.

Eine Nachricht direkt an einen Client senden

Eine Alternative zu einem Client, der eine Subskription erstellt und eine Veröffentlichung empfängt, die mit dem Subskriptionsthema übereinstimmt, können Sie eine Nachricht auch direkt an einen MQTT v3-Client senden. MQTT V3-Clientanwendungen können Nachrichten nicht direkt senden, aber andere Anwendungen, z. B. IBM MQ-Anwendungen, können es.

Die IBM MQ-Anwendung muss die Client-ID (`ClientIdentifier`) des MQTT v3-Clients kennen. Da MQTT v3-Clients keine Warteschlangen haben, wird der Name der Zielwarteschlange als Themaname an die Methode `messageArrived` des MQTT v3-Anwendungsclients übergeben. Erstellen Sie zum Beispiel in einem MQI-Programm einen Objektdeskriptor mit dem Client als `ObjectQmgrName` :

```
MQOD.ObjectQmgrName = ClientIdentifier ;
MQOD.ObjectName = name ;
```

Abbildung 24. MQI-Objektdeskriptor zum Senden einer Nachricht an ein MQTT v3-Clientziel

Wenn die Anwendung unter Verwendung von JMS geschrieben wird, erstellen Sie ein Punkt-zu-Punkt-Ziel; Beispiel:

```
javax.jms.Destination jmsDestination =
(javax.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

Abbildung 25. JMS-Ziel zum Senden einer Nachricht an einen MQTT v3-Client

Wenn Sie eine nicht erwartete Nachricht an einen MQTT-Client senden möchten, verwenden Sie eine Definition einer fernen Warteschlange. Der Name des fernen Warteschlangenmanagers muss in die Client-ID (`ClientIdentifier`) des Clients aufgelöst werden. Die Übertragungswarteschlange muss in `SYSTEM.MQTT.TRANSMIT.QUEUE` aufgelöst werden (siehe [Tabelle 15](#) auf Seite 225). Der Name der fernen Warteschlange kann alles sein. Der Client empfängt ihn als Themenzeichenfolge.

Eingabe		Ausgabe		
Warteschlangenna- me	Name des Warteschlan- genmanagers	Warteschlangenna- me	Name des Warteschlan- genmanagers	Übertragungswar- teschlange
Name der Definiti- on der fernen Warte- schlange	Leerer oder lokaler WS-Managername	Name der fernen Warteschlange, die als Themenzeichen- folge verwendet wird	<code>ClientIdenti- fier</code>	<code>SYSTEM.MQTT. TRANSMIT.QUEUE</code>

Wenn der Client verbunden ist, wird die Nachricht direkt an den MQTT -Client gesendet, der die Methode `messageArrived` aufruft; siehe [MethodemessageArrived](#).

Wenn der Client die Verbindung zu einer persistenten Sitzung getrennt hat, wird die Nachricht in `SYSTEM.MQTT.TRANSMIT.QUEUE` gespeichert. siehe [MQTT statusunabhängige und statusabhängige Sitzun-](#)

gen. Er wird an den Client weitergeleitet, wenn der Client die Verbindung erneut zu der Sitzung wiederherstellt.

Wenn Sie eine nicht persistente Nachricht senden, wird sie mit der Servicequalität `at the 1ichste` an den Client gesendet, `QoS=0`. Wenn Sie eine persistente Nachricht direkt an einen Client senden, wird diese standardmäßig mit der Servicequalität `genau einmal` gesendet, `QoS=2`. Da der Client möglicherweise keinen Persistenzmechanismus hat, kann der Client die Servicequalität verringern, die er für direkt gesendete Nachrichten akzeptiert. Um die Servicequalität für direkt an einen Client gesendete Nachrichten zu verringern, erstellen Sie eine Subskription für das Thema `DEFAULT.QoS`. Geben Sie die maximale Servicequalität an, die der Client unterstützen kann.

Windows

Linux

AIX

MQTT-Client - Identifikation, Berechtigung und

Authentifizierung

Der Telemetrieservice (MQXR) führt Veröffentlichungen oder Subskriptionen für IBM MQ-Themen im Namen von MQTT-Clients über MQTT-Kanäle durch. Der IBM MQ-Administrator konfiguriert die MQTT-Kanalidentität, die für die IBM MQ-Berechtigung verwendet wird. Der Administrator kann eine gemeinsame Identität für den Kanal definieren oder den Benutzernamen oder den `ClientIdentifier` eines Clients verwenden, der mit dem Kanal verbunden ist.

Der Telemetrieservice (MQXR) kann den Client über den Benutzernamen authentifizieren, der vom Client bereitgestellt wird, oder indem er ein Clientzertifikat verwendet. Der Benutzername wird mit einem Kennwort authentifiziert, das vom Client bereitgestellt wird.

Zusammenfassen: Die Client-ID ist die Auswahl der Cliententität. Je nach Kontext wird der Client durch die `Client-ID`, `Benutzername`, eine vom Administrator erstellte gemeinsame Cliententität oder ein Clientzertifikat identifiziert. Die für die Authentizitätsprüfung verwendete Client-ID muss nicht mit der ID identisch sein, die für die Autorisierung verwendet wird.

MQTT-Clientprogramme legen Benutzername und Kennwort fest, die über einen MQTT-Kanal an den Server gesendet werden. Sie können auch die TLS-Eigenschaften festlegen, die zum Verschlüsseln und Authentifizieren der Verbindung erforderlich sind. Der Administrator entscheidet, ob der MQTT-Kanal authentifiziert werden soll und wie er authentifiziert werden soll.

Wenn Sie einen MQTT-Client für den Zugriff auf IBM MQ-Objekte berechtigen möchten, müssen Sie die `Client-ID` oder den Benutzernamen des Clients oder eine gemeinsame Cliententität berechtigen. Wenn Sie einem Client die Verbindung mit IBM MQ ermöglichen möchten, berechtigen Sie den Benutzernamen oder verwenden Sie ein Clientzertifikat. Konfigurieren Sie JAAS für die Authentifizierung des Benutzernamens und konfigurieren Sie TLS für die Authentifizierung eines Clientzertifikats.

Wenn Sie ein Kennwort auf dem Client festlegen, verschlüsseln Sie die Verbindung entweder über VPN oder konfigurieren Sie den MQTT-Kanal für die Verwendung von TLS, um das Kennwort privat zu halten.

Es ist schwierig, Clientzertifikate zu verwalten. Aus diesem Grund wird die Kennwortauthentifizierung häufig zur Authentifizierung von Clients verwendet, wenn die Risiken, die mit der Kennwortauthentifizierung verknüpft sind, akzeptabel sind.

Wenn es eine sichere Möglichkeit gibt, das Clientzertifikat zu verwalten und zu speichern, ist es möglich, sich auf die Zertifikatsauthentifizierung zu verlassen. Es ist jedoch selten der Fall, dass Zertifikate sicher in den Typen von Umgebungen verwaltet werden können, in denen Telemetrie verwendet wird. Stattdessen wird die Authentifizierung von Einheiten, die Clientzertifikate verwenden, durch die Authentifizierung von Clientkennwörtern auf dem Server ergänzt. Aufgrund der zusätzlichen Komplexität ist die Verwendung von Clientzertifikaten auf hochsensible Anwendungen beschränkt. Die Verwendung von zwei Formen der Authentifizierung wird als Zwei-Faktor-Authentifizierung bezeichnet. Sie müssen einen der Faktoren kennen, z. B. ein Kennwort, und die anderen Faktoren, wie z. B. ein Zertifikat, haben.

Bei einer hochempfindlichen Anwendung, wie z. B. einer Chip-und-Pin-Vorrichtung, wird die Vorrichtung während der Fertigung gesperrt, um Manipulationen an der internen Hardware und Software zu verhindern. Ein anerkanntes, zeitbegrenzt Clientzertifikat wird in die Einheit kopiert. Die Einheit wird an der Position implementiert, an der sie verwendet werden soll. Die weitere Authentifizierung wird jedes Mal durchgeführt, wenn die Einheit verwendet wird, entweder mit einem Kennwort oder einem anderen Zertifikat von einer Smart-Card.

Verwenden Sie die Client-ID, den Benutzernamen oder eine allgemeine Clientidentität als Berechtigung für den Zugriff auf IBM MQ-Objekte.

Der IBM MQ-Administrator hat drei Möglichkeiten bei der Auswahl der Identität des MQTT-Kanals. Der Administrator trifft die Auswahl bei der Definition oder Änderung des MQTT-Kanals, der vom Client verwendet wird. Die Identität dient dazu, den Zugriff auf IBM MQ-Themen zu berechtigen. Die Auswahl erfolgt in der folgenden Reihenfolge:

1. Die Client-ID (siehe [USECLNTID](#)).
2. Eine Identität, die der Administrator für den Kanal bereitstellt (der MCAUSER des Kanals). Siehe [MCAUSER](#)).
3. Wenn keine der vorherigen Optionen zutrifft, der vom MQTT-Client übergebene Benutzername (Benutzername ist ein Attribut der Klasse `MqttConnectOptions`. Es muss festgelegt werden, bevor der Client eine Verbindung zum Service herstellt. Der Standardwert ist null).

Vermeiden Sie Probleme.: Die Identität, die von diesem Prozess ausgewählt wird, wird nachfolgend beispielsweise mit dem Befehl `DISPLAY CHSTATUS (MQTT)` als MCAUSER des Clients bezeichnet. Beachten Sie, dass dies nicht unbedingt die gleiche Identität wie der MCAUSER des Kanals ist, auf den in der Auswahl (2) verwiesen wird.

Wählen Sie mit dem IBM MQ-Befehl `setmqaut` aus, welche Objekte und welche Aktionen berechtigt sind, von der Identität verwendet zu werden, die dem MQTT-Kanal zugeordnet ist. Der folgende Code berechtigt zum Beispiel zum Autorisieren einer Kanalidentität `MQTTClient`, die vom Administrator des WS-Managers `QM1` bereitgestellt wird:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

Führen Sie die folgenden Schritte aus, um MQTT-Clients zu berechtigen, Veröffentlichungen und Subskriptionen für IBM MQ-Objekte durchzuführen. Die Schritte folgen vier alternativen Zugriffssteuerungsmustern.

Vorbereitende Schritte

MQTT-Clients werden berechtigt, auf Objekte in IBM MQ zuzugreifen, indem ihnen eine Identität zugeordnet wird, wenn sie eine Verbindung zu einem Telemetrikkanal herstellen. Der IBM MQ-Administrator konfiguriert den Telemetrikkanal mithilfe von IBM MQ Explorer, um einem Client eine von drei Identitätstypen zuzuweisen:

1. ClientIdentifizier
2. Benutzername
3. Ein Name, den der Administrator dem Kanal zuweist.

Bei jedem der Typen muss die Identität vom installierten Berechtigungsservice als Prinzipal für IBM MQ definiert werden. Der Standardberechtigungs-service unter Windows oder Linux ist der Object Authority Manager (OAM, Objektberechtigungsmanager). Wenn Sie den OAM verwenden, muss die Identität als Benutzer-ID definiert sein.

Verwenden Sie die Identität, um einem Client oder einer Gruppe von Clients die Berechtigung zu erteilen, Veröffentlichungen oder Subskriptionen für Themen durchzuführen, die in IBM MQ definiert sind. Wenn ein MQTT-Client ein Thema subskribiert hat, erteilen Sie ihm mithilfe der Identität die Berechtigung, die Veröffentlichungen zu dem Thema zu empfangen.

Es ist schwierig, ein System mit Zehntausenden von MQTT-Clients zu verwalten, die jeweils individuelle Zugriffsberechtigungen erfordern. Eine Lösung besteht darin, gemeinsame Identitäten zu definieren und

einzelnen MQTT-Clients eine der gemeinsamen Identitäten zuzuordnen. Definieren Sie so viele gemeinsame Identitäten wie erforderlich, um unterschiedliche Kombinationen von Berechtigungen zu definieren. Eine andere Lösung besteht darin, einen eigenen Berechtigungsservice zu schreiben, der mit Tausenden von Benutzern einfacher umgehen kann als das Betriebssystem.

Es gibt zwei Möglichkeiten, MQTT-Clients über den OAM mit gemeinsamen Identitäten zu versehen:

1. Definieren Sie mehrere Telemetriedkanäle mit jeweils einer anderen Benutzer-ID, die der Administrator mit IBM MQ Explorer zuordnet. Clients, die unter Verwendung unterschiedlicher TCP/IP-Portnummern eine Verbindung herstellen, sind verschiedenen Telemetriedkanälen zugeordnet und weisen unterschiedliche Identitäten auf.
2. Definieren Sie einen einzelnen Telemetriedkanal, aber lassen Sie jeden Client einen Benutzernamen aus einer kleinen Gruppe von Benutzer-IDs auswählen. Der Administrator konfiguriert den Telemetriedkanal, um den Client `Username` als Identität auszuwählen.

In dieser Task wird die Identität des Telemetriedkanals `mqttUser`, unabhängig davon, wie er festgelegt ist, aufgerufen. Wenn Sammlungen von Clients unterschiedliche Identitäten verwenden, verwenden Sie mehrere `mqttUsers`, eine für jede Sammlung von Clients. Da die Task den OAM verwendet, muss jede `mqttUser` eine Benutzer-ID sein.

Informationen zu diesem Vorgang

In dieser Task haben Sie eine Auswahl von vier Zugriffssteuerungsmustern, die Sie an bestimmte Anforderungen anpassen können. Die Muster unterscheiden sich in ihrer Granularität der Zugriffssteuerung.

- „Keine Zugriffssteuerung“ auf Seite 228
- „Coarse-Grained-Zugriffssteuerung“ auf Seite 228
- „Mittlere Zugriffssteuerung“ auf Seite 229
- „Feinkörnige Zugriffssteuerung“ auf Seite 229

Das Ergebnis der Modelle besteht darin, MQTT-Benutzern (`mqttUsers`) Berechtigungen zuzuweisen, die es ihnen ermöglichen, Veröffentlichungen und Subskriptionen für IBM MQ durchzuführen und Veröffentlichungen von IBM MQ zu empfangen.

Keine Zugriffssteuerung

MQTT-Clients erhalten die IBM MQ-Verwaltungsberechtigung und können jede beliebige Aktion für ein Objekt ausführen.

Vorgehensweise

1. Erstellen Sie eine Benutzer-ID `mqttUser`, die als Identität für alle MQTT-Clients gilt.
2. Fügen Sie `mqttUser` zur Gruppe `mqm` hinzu (siehe [Benutzer unter Windows zu einer Gruppe hinzufügen](#) bzw. [Benutzer unter Linux zu einer Gruppe hinzufügen](#)).

Coarse-Grained-Zugriffssteuerung

MQTT-Clients verfügen über die Berechtigung zum Veröffentlichen und Subskribieren und zum Senden von Nachrichten an MQTT-Clients. Sie verfügen nicht über die Berechtigung zum Ausführen anderer Aktionen oder zum Zugriff auf andere Objekte.

Vorgehensweise

1. Erstellen Sie eine Benutzer-ID `mqttUser`, die als Identität für alle MQTT-Clients gilt.
2. Berechtigen Sie `mqttUser`, zu allen Themen Veröffentlichungen und Subskriptionen durchzuführen und Veröffentlichungen an MQTT-Clients zu senden.

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

Mittlere Zugriffssteuerung

MQTT-Clients sind in verschiedene Gruppen aufgeteilt, um Veröffentlichungen und Subskriptionen zu verschiedenen Themengruppen durchzuführen und Nachrichten an MQTT-Clients zu senden.

Vorgehensweise

1. Erstellen Sie mehrere Benutzer-IDs, *mqttUsers* und mehrere Verwaltungsthemen in der Publish/Subscribe-Themenstruktur.
2. Autorisieren Sie verschiedene *mqttUsers* zu verschiedenen Themen.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub  
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. Erstellen Sie eine Gruppe *mqtt* und fügen Sie alle *mqttUsers* zur Gruppe hinzu.
4. Berechtigen Sie *mqtt* zum Senden von Themen an MQTT -Clients.

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Feinkörnige Zugriffssteuerung

MQTT-Clients werden in ein vorhandenes Zugriffssteuerungssystem integriert, das Gruppen berechtigt, Aktionen für Objekte auszuführen.

Informationen zu diesem Vorgang

Eine Benutzer-ID wird abhängig von den Berechtigungen, die sie benötigt, einer oder mehreren Betriebssystemgruppen zugeordnet. Verwenden Sie dieses Modell, wenn IBM MQ-Anwendungen Veröffentlichungen und Subskriptionen für denselben Themenbereich wie MQTT-Clients durchführen. Die Gruppen werden als Publish X, Subscribe Y und mqtt bezeichnet.

Publish X

Mitglieder von Publish X-Gruppen können in *topicX* veröffentlichen.

Subscribe Y

Mitglieder von Subscribe Y-Gruppen können *topicY* abonnieren.

mqtt

Mitglieder der Gruppe *mqtt* können Veröffentlichungen an MQTT-Clients senden.

Vorgehensweise

1. Erstellen Sie mehrere Gruppen Publish X und Subscribe Y, die mehreren Verwaltungsthemen in der Publish/Subscribe-Themenstruktur zugeordnet sind.
2. Erstellen Sie eine Gruppe mqtt.
3. Erstellen Sie mehrere Benutzer-IDs, *mqttUsers*, und fügen Sie die Benutzer zu einer der Gruppen hinzu, je nachdem, welche Berechtigung sie haben, um sie zu tun.
4. Berechtigen Sie verschiedene Publish X- und Subscribe X-Gruppen für verschiedene Themen und berechtigen Sie die *mqtt*-Gruppe, Nachrichten an MQTT -Clients zu senden.

```
setmqaut -m qMgr -t topic -n topic1 -p Publish X -all +pub  
setmqaut -m qMgr -t topic -n topic1 -p Subscribe X -all +pub +sub  
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

wort

Authentifizieren Sie den Benutzernamen mithilfe des Clientkennworts. Sie können den Client mit einer anderen Identität gegenüber der Identität authentifizieren, die verwendet wird, um den Client zu autorisieren, Themen zu veröffentlichen und zu subscribieren.

Der Telemetrieservice (MQXR) verwendet JAAS, um den Client benutzername zu authentifizieren. JAAS verwendet das Kennwort, das vom MQTT-Client bereitgestellt wird.

Der IBM MQ-Administrator entscheidet, ob der Benutzername authentifiziert werden soll oder nicht, indem er den MQTT-Kanal konfiguriert, zu dem ein Client eine Verbindung herstellt. Clients können verschiedenen Kanälen zugeordnet werden, und jeder Kanal kann so konfiguriert werden, dass er seine Clients auf unterschiedliche Weise authentifiziert. Mit JAAS können Sie konfigurieren, welche Methoden den Client authentifizieren müssen und die optional den Client authentifizieren können.

Die Auswahl der Identität für die Authentifizierung hat keinen Einfluss auf die Auswahl der Identität für die Berechtigung. Es kann sein, dass Sie eine gemeinsame Identität für die Berechtigung für den Verwaltungskomfort einrichten möchten, aber jeder Benutzer authentifiziert wird, um diese Identität zu verwenden. In der folgenden Prozedur werden die Schritte zur Authentifizierung einzelner Benutzer für die Verwendung einer gemeinsamen Identität beschrieben:

1. Der IBM MQ -Administrator setzt die Identität des MQTT -Kanals mithilfe von IBM MQ Explorer auf einen beliebigen Namen, wie z. B. MQTTClientUser.
2. Der IBM MQ-Administrator berechtigt MQTTClient, Veröffentlichungen und Subskriptionen für alle Themen durchzuführen:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. Der MQTT-Clientanwendungsentwickler erstellt ein MqttConnectOptions-Objekt und legt Benutzername und Kennwort fest, bevor eine Verbindung zum Server hergestellt wird.
4. Der Sicherheitsentwickler erstellt ein JAAS LoginModule zur Authentifizierung des Benutzernamens mit dem Kennwort und schließt es in die JAAS -Konfigurationsdatei ein.
5. Der IBM MQ-Administrator konfiguriert den MQTT-Kanal, um den Benutzernamen des Clients mithilfe von JAAS zu authentifizieren.

Verbindungen zwischen dem MQTT-Client und dem Warteschlangenmanager werden immer vom MQTT-Client eingeleitet. Der MQTT-Client ist immer der SSL-Client. Die Clientauthentifizierung für den Server und die Serverauthentifizierung des MQTT-Clients sind beide optional.

Wenn Sie dem Client ein privates signiertes digitales Zertifikat bereitstellen, können Sie den MQTT-Client für WebSphere MQauthentifizieren. Der WebSphere MQ-Administrator kann MQTT-Clients auch zwingen, sich selbst über TLS beim Warteschlangenmanager zu authentifizieren. Eine Clientauthentifizierung kann nur im Rahmen einer gegenseitigen Authentifizierung angefordert werden.

Als Alternative zu SSL können bestimmte virtuelle private Netze (Virtual Private Network, VPN), z. B. IPsec, die Endpunkte einer TCP/IP-Verbindung authentifizieren. In einem VPN wird jedes IP-Paket, das im Netz übertragen wird, verschlüsselt. Sobald eine solche VPN-Verbindung hergestellt ist, haben Sie ein vertrauenswürdiges Netz aufgebaut. MQTT-Clients können auch mit TCP/IP über das VPN mit Telemetriedkanälen verbunden werden.

Die Clientauthentifizierung über TLS setzt voraus, dass der Client über einen geheimen Schlüssel verfügt. Der geheime Schlüssel ist im Falle eines selbst signierten Zertifikats der private Schlüssel des Clients bzw. andernfalls ein Schlüssel, der von einer Zertifizierungsstelle ausgestellt wird. Mit dem Schlüssel wird das digitale Zertifikat des Clients signiert. Jede Person im Besitz des entsprechenden öffentlichen Schlüssels kann das digitale Zertifikat verifizieren. Zertifikate können vertrauenswürdig sein oder - falls sie verkettet sind - durch eine Zertifikatskette bis zu einem Trusted-Root-Zertifikat zurückverfolgt werden.

Bei der Clientverifizierung werden alle Zertifikate in der vom Client bereitgestellten Zertifikatskette an den Server gesendet. Der Server überprüft die Zertifikatskette, bis er ein Zertifikat findet, dem er vertraut. Das vertrauenswürdige Zertifikat ist entweder das öffentliche Zertifikat, das auf Basis eines selbst signierten Zertifikats generiert wird, oder ein Stammzertifikat, das für gewöhnlich von einer Zertifizierungsstelle ausgestellt wird. In einem letzten (optionalen) Schritt kann das vertrauenswürdige Zertifikat mit einer aktuellen Zertifikatswiderrufsliste abgeglichen werden.

Das vertrauenswürdige Zertifikat kann von einer Zertifizierungsstelle ausgegeben und bereits im JRE-Zertifikatsspeicher enthalten sein. Es kann sich aber auch um ein selbst signiertes Zertifikat oder um jedes Zertifikat handeln, das dem Keystore des Telemetriekanals als vertrauenswürdige Zertifikat hinzugefügt wurde.

Anmerkung: Der Telemetriekanal verfügt über eine Kombination aus Keystore/Truststore, die sowohl private Schlüssel für einen oder mehrere Telemetriekanäle enthält, als auch öffentliche Zertifikate, die zur Authentifizierung von Clients erforderlich sind. Da ein SSL-Kanal einen Keystore haben muss und dies dieselbe Datei wie der Truststore des Kanals ist, wird der JRE-Zertifikatsspeicher niemals referenziert. Dies führt dazu, dass Sie das Stammzertifikat in den Keystore für den Kanal stellen müssen, wenn für eine Authentifizierung eines Clients ein Stammzertifikat einer Zertifizierungsstelle erforderlich ist. Dies gilt selbst dann, wenn sich das Stammzertifikat der Zertifizierungsstelle bereits im JRE-Zertifikatsspeicher befindet. Der JRE-Zertifikatsspeicher wird niemals referenziert.

Überlegen Sie sich, welchen Sicherheitsbedrohungen die Clientauthentifizierung entgegenwirken soll, und welche Rollen der Client und Server bei der Abwendung der Sicherheitsbedrohungen spielen. Die Authentifizierung des Clientzertifikats reicht allein nicht aus, um unbefugten Zugriff auf ein System zu verhindern. Wenn eine dritte Person über das Clientgerät verfügt, handelt das Clientgerät nicht unbedingt mit der Berechtigung des Zertifikatseigners. Verlassen Sie sich bei der Abwehr von unerwünschten Angriffen niemals auf eine einzige Maßnahme. Verwenden Sie mindestens eine aus zwei Faktoren bestehende Authentifizierung und sichern Sie dies durch den Besitz eines Zertifikats ab, für das nicht öffentliche Informationen bekannt sein müssen. Verwenden Sie beispielsweise JAAS und authentifizieren Sie den Client mittels eines vom Server ausgestellten Kennworts.

Die größte Sicherheitsbedrohung für das Clientzertifikat ist, dass es in falsche Hände gelangt. Das Zertifikat befindet sich in einem kennwortgeschützten Keystore beim Client. Wie wird es in den Keystore gestellt? Wie erhält der MQTT-Client das Kennwort für den Keystore? Wie sicher ist der Kennwortschutz? Telemetriegeräte sind häufig einfach zu entfernen und können dann in Ruhe manipuliert werden. Muss das Gerät manipulationssicher sein? Die Verteilung und der Schutz clientseitiger Zertifikate gilt als besonders schwierig; diese Aufgabe wird als das Schlüsselverwaltungsproblem bezeichnet.

Eine weitere große Bedrohung besteht darin, dass das Gerät unbeabsichtigt für den Zugriff auf Server missbraucht wird. Wird die MQTT-Anwendung beispielsweise manipuliert, kann unter Verwendung der authentifizierten Clientidentität eine Schwachstelle in der Serverkonfiguration ausgenutzt werden.

Zur Authentifizierung eines MQTT-Clients mit SSL müssen Sie den Telemetriekanal und den Client konfigurieren.

Zugehörige Konzepte

[„Konfiguration des Telemetriekanals für MQTT-Clientauthentifizierung mit TLS“ auf Seite 231](#)

Der IBM MQ -Administrator konfiguriert Telemetriekanäle auf dem Server. Jeder Kanal wird so konfiguriert, dass er eine TCP/IP-Verbindung an einer anderen Portnummer akzeptiert. TLS-Kanäle werden mit kennphrasengeschütztem Zugriff auf Schlüsseldateien konfiguriert. Wird ein TLS-Kanal ohne Kennphrase oder Schlüsseldatei definiert, akzeptiert der Kanal keine TLS-Verbindungen.

Zugehörige Informationen

[MQTT-Clientkonfiguration für Clientauthentifizierung mit TLS](#)

Konfiguration des Telemetriekanals für MQTT-Clientauthentifizierung mit TLS

Der IBM MQ -Administrator konfiguriert Telemetriekanäle auf dem Server. Jeder Kanal wird so konfiguriert, dass er eine TCP/IP-Verbindung an einer anderen Portnummer akzeptiert. TLS-Kanäle werden mit kennphrasengeschütztem Zugriff auf Schlüsseldateien konfiguriert. Wird ein TLS-Kanal ohne Kennphrase oder Schlüsseldatei definiert, akzeptiert der Kanal keine TLS-Verbindungen.

Legen Sie die Eigenschaft `com.ibm.mq.MQTT.ClientAuth` eines TLS-Telemetriekanals auf `REQUIRED` fest, um alle Clients, die eine Verbindung zu diesem Kanal herstellen, zu zwingen, den Nachweis zu erbringen, dass sie digitale Zertifikate überprüft haben. Die Clientzertifikate werden mit Hilfe von Zertifikaten von Zertifizierungsstellen authentifiziert, was zu einem Trusted-Root-Zertifikat führt. Wenn das Clientzertifikat selbst signiert ist oder von einem Zertifikat signiert wird, das von einer Zertifizierungsstelle ausgestellt wurde, müssen die öffentlich signierten Zertifikate der Client- oder Zertifizierungsstelle sicher auf dem Server gespeichert werden.

Stellen Sie das öffentlich signierte Clientzertifikat oder das Zertifikat aus der Zertifizierungsstelle in den Telemetriekanal-Keystore. Auf dem Server werden öffentlich signierte Zertifikate in derselben Schlüsseldatei wie privat signierte Zertifikate gespeichert, und nicht in einem separaten Truststore.

Der Server überprüft die Signatur aller Clientzertifikate, die er mit allen öffentlichen Zertifikaten und Cipher Suites, die er hat, gesendet wird. Der Server überprüft die Schlüsselkette. Der WS-Manager kann so konfiguriert werden, dass er das Zertifikat gegen die Zertifikatswiderrufsliste testen kann. Die Eigenschaft für die Widerrufsnamensliste des Warteschlangenmanagers heißt `SSLCRLNL`.

Wenn eines der Zertifikate, die ein Client sendet, durch ein Zertifikat im Serverschlüsselspeicher verifiziert wird, wird der Client authentifiziert.

Der IBM MQ-Administrator kann denselben Telemetriekanal so konfigurieren, dass JAAS zur Überprüfung von Benutzername oder `Client-ID` des Clients mit dem Client-Kennwort verwendet wird.

Sie können denselben Schlüsselspeicher für mehrere Telemetriekanäle verwenden.

Die Prüfung von mindestens einem digitalen Zertifikat im kennwortgeschützten Client-Keystore auf dem Gerät authentifiziert den Client gegenüber dem Server. Das digitale Zertifikat wird nur zur Authentifizierung durch IBM MQ verwendet. Es wird nicht verwendet, um die TCP/IP-Adresse des Clients zu überprüfen, oder die Identität des Clients für die Autorisierung oder Abrechnung festzulegen. Die vom Server angenommene Identität des Clients ist entweder der Benutzername oder die `Client-ID` des Clients oder eine Identität, die vom IBM MQ-Administrator erstellt wurde.

Sie können TLS-Cipher-Suites auch für die Clientauthentifizierung verwenden. Wenn Sie beabsichtigen, SHA-2 -Cipher-Suites zu verwenden, lesen Sie den Abschnitt [„Systemvoraussetzungen für die Verwendung von SHA-2-Cipher-Suites zusammen mit MQTT-Kanälen“](#) auf Seite 234

Zugehörige Konzepte

[„Telemetriekanal-Konfiguration für Kanalauthentifizierung mit TLS“](#) auf Seite 233

Der IBM MQ -Administrator konfiguriert Telemetriekanäle auf dem Server. Jeder Kanal wird so konfiguriert, dass er eine TCP/IP-Verbindung an einer anderen Portnummer akzeptiert. TLS-Kanäle werden mit kennphasengeschütztem Zugriff auf Schlüsseldateien konfiguriert. Wird ein TLS-Kanal ohne Kennphrase oder Schlüsseldatei definiert, akzeptiert der Kanal keine TLS-Verbindungen.

Zugehörige Informationen

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

[CipherSpecs und CipherSuites](#)

Windows

Linux

AIX

Telemetriekanal-Authentifizierung mit TLS

Verbindungen zwischen dem MQTT-Client und dem Warteschlangenmanager werden immer vom MQTT-Client eingeleitet. Der MQTT-Client ist immer der SSL-Client. Die Clientauthentifizierung für den Server und die Serverauthentifizierung des MQTT-Clients sind beide optional.

Der Client versucht immer, den Server zu authentifizieren, es sei denn, der Client wurde für die Verwendung einer CipherSpec konfiguriert, die anonyme Verbindungen unterstützt. Schlägt die Authentifizierung fehl, wird keine Verbindung hergestellt.

Als Alternative zu SSL können bestimmte virtuelle private Netze (Virtual Private Network, VPN), z. B. IPsec, die Endpunkte einer TCP/IP-Verbindung authentifizieren. In einem VPN wird jedes IP-Paket, das im Netz übertragen wird, verschlüsselt. Sobald eine solche VPN-Verbindung hergestellt ist, haben Sie ein

vertrauenswürdigen Netz aufgebaut. MQTT-Clients können auch mit TCP/IP über das VPN mit Telemetriedatenkanälen verbunden werden.

Bei der Serverauthentifizierung mit SSL wird der Server, an den Sie vertrauliche Informationen senden möchten, mithilfe von Secure Sockets Layer (SSL) authentifiziert. Der Client führt die Prüfungen durch, die mit den vom Server gesendeten Zertifikaten übereinstimmen, mit Zertifikaten, die in seinem Truststore oder in seinem JRE- cacerts Speicher gespeichert sind.

Der JRE-Zertifikatsspeicher ist eine JKS-Datei, cacerts. Sie befindet sich im Pfad JRE Install-Path\lib\security\. Sie wird mit dem Standardkennwort changeit installiert. Sie können Zertifikate, denen Sie vertrauen, entweder im JRE-Zertifikatsspeicher oder im Truststore des Clients speichern. Sie können jedoch nicht beide Speicher verwenden. Verwenden Sie den Truststore des Clients, wenn Sie die öffentlichen Zertifikate, denen der Client vertraut, getrennt von den Zertifikaten aufbewahren möchten, die von anderen Java-Anwendungen verwendet werden. Verwenden Sie den JRE-Zertifikatsspeicher, wenn Sie einen allgemeinen Zertifikatsspeicher für alle Java-Anwendungen nutzen möchten, die auf dem Client ausgeführt werden. Wenn Sie sich für die Verwendung des JRE-Zertifikatsspeichers entscheiden, überprüfen Sie die darin enthaltenen Zertifikate, um sicherzustellen, dass diese wirklich vertrauenswürdig sind.

Sie können die JSSE-Konfiguration ändern, indem Sie einen anderen Trust-Provider bereitstellen. Ein Trust-Provider kann so angepasst werden, dass er verschiedene Zertifikatsprüfungen durchführt. In einigen OGSi-Umgebungen, die den MQTT-Client verwendet haben, wird ein anderer Trust-Provider bereitgestellt.

Zur Authentifizierung eines Telemetriedatenkanals mit TLS müssen Sie den Server und den Client konfigurieren.

Windows Linux AIX **Telemetriedatenkanal-Konfiguration für Kanalauthentifizierung mit TLS**

Der IBM MQ -Administrator konfiguriert Telemetriedatenkanäle auf dem Server. Jeder Kanal wird so konfiguriert, dass er eine TCP/IP-Verbindung an einer anderen Portnummer akzeptiert. TLS-Kanäle werden mit kennphasengeschütztem Zugriff auf Schlüsseldateien konfiguriert. Wird ein TLS-Kanal ohne Kennphrase oder Schlüsseldatei definiert, akzeptiert der Kanal keine TLS-Verbindungen.

Speichern Sie das digitale Zertifikat des Servers, signiert mit seinem privaten Schlüssel, im Keystore, den der Telemetriedatenkanal auf dem Server verwenden wird. Speichern Sie alle Zertifikate in seiner Schlüsselkette im Keystore, wenn Sie die Schlüsselkette an den Client übertragen möchten. Konfigurieren Sie den Telemetriedatenkanal mit dem IBM MQ-Explorer für die Verwendung von TLS. Geben Sie den Pfad zum Keystore und den Verschlüsselungstext für den Zugriff auf den Schlüssel Speicher an. Wenn Sie die TCP/IP-Portnummer des Kanals nicht festlegen, nimmt die TLS-Telemetriedatenkanalanschlussnummer standardmäßig den Wert 8883 an.

Sie können TLS-Cipher-Suites auch für die Kanalauthentifizierung verwenden. Wenn Sie beabsichtigen, SHA-2 -Cipher-Suites zu verwenden, lesen Sie den Abschnitt [„Systemvoraussetzungen für die Verwendung von SHA-2-Cipher-Suites zusammen mit MQTT-Kanälen“](#) auf Seite 234

Zugehörige Konzepte

[„Konfiguration des Telemetriedatenkanals für MQTT-Clientauthentifizierung mit TLS“](#) auf Seite 231

Der IBM MQ -Administrator konfiguriert Telemetriedatenkanäle auf dem Server. Jeder Kanal wird so konfiguriert, dass er eine TCP/IP-Verbindung an einer anderen Portnummer akzeptiert. TLS-Kanäle werden mit kennphasengeschütztem Zugriff auf Schlüsseldateien konfiguriert. Wird ein TLS-Kanal ohne Kennphrase oder Schlüsseldatei definiert, akzeptiert der Kanal keine TLS-Verbindungen.

Zugehörige Informationen

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

[CipherSpecs und CipherSuites](#)

Systemvoraussetzungen für die Verwendung von SHA-2-Cipher-Suites zusammen mit MQTT-Kanälen

Wenn Sie eine Version von Java verwenden, die SHA-2-Cipher-Suites unterstützt, können Sie mithilfe dieser Suites Ihre MQTT-(Telemetrie-)Kanäle und Client-Apps schützen.

Für IBM MQ 8.0, das den Telemetrieservice (MQXR) umfasst, ist die minimale Java-Version Java 7 von IBM, SR6. SHA-2-Cipher-Suites werden standardmäßig in Java 7 von IBM, ab SR4 unterstützt. Sie können daher SHA-2-Cipher-Suites mit dem Telemetrieservice (MQXR) verwenden, um Ihre MQTT-(Telemetrie-)Kanäle zu schützen.

Wenn Sie einen MQTT-Client mit einer anderen JRE ausführen, müssen Sie sicherstellen, dass sie auch die SHA-2-Cipher-Suites unterstützt.

Zugehörige Konzepte

[„Telemetriekanal-Konfiguration für Kanalauthentifizierung mit TLS“ auf Seite 233](#)

Der IBM MQ -Administrator konfiguriert Telemetriekanäle auf dem Server. Jeder Kanal wird so konfiguriert, dass er eine TCP/IP-Verbindung an einer anderen Portnummer akzeptiert. TLS-Kanäle werden mit kennphrasengeschütztem Zugriff auf Schlüsseldateien konfiguriert. Wird ein TLS-Kanal ohne Kennphrase oder Schlüsseldatei definiert, akzeptiert der Kanal keine TLS-Verbindungen.

Zugehörige Informationen

[Telemetrieservice \(MQXR\)](#)

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

Datenschutz auf Telemetriekanälen

Die Vertraulichkeit von MQTT-Veröffentlichungen, die in beide Richtungen über Telemetriekanäle gesendet werden, wird durch die Verwendung von TLS zum Verschlüsseln von Übertragungen über die Verbindung geschützt.

MQTT-Clients, die Verbindungen zu Telemetriekanälen herstellen, verwenden TLS, um die Vertraulichkeit von Veröffentlichungen zu schützen, die über die symmetrische Schlüsselverschlüsselung auf dem Kanal übertragen werden. Da die Endpunkte nicht authentifiziert sind, können Sie die Kanalverschlüsselung nicht allein vertrauen. Kombinieren Sie den Schutz der Privatsphäre mit dem Server oder der gegenseitigen Authentifizierung.

Als Alternative zu SSL können bestimmte virtuelle private Netze (Virtual Private Network, VPN), z. B. IPsec, die Endpunkte einer TCP/IP-Verbindung authentifizieren. In einem VPN wird jedes IP-Paket, das im Netz übertragen wird, verschlüsselt. Sobald eine solche VPN-Verbindung hergestellt ist, haben Sie ein vertrauenswürdiges Netz aufgebaut. MQTT-Clients können auch mit TCP/IP über das VPN mit Telemetriekanälen verbunden werden.

Informationen zu einer typischen Konfiguration mit Kanalverschlüsselung und Serverauthentifizierung finden Sie im Abschnitt [„Telemetriekanal-Authentifizierung mit TLS“ auf Seite 232](#).

Das Verschlüsseln von TLS-Verbindungen ohne die Authentifizierung des Servers macht die Verbindung zu Man-in-the-Middle-Attacken verfügbar. Obwohl die Informationen, die Sie austauschen, vor Lauschangriffen geschützt sind, wissen Sie nicht, mit wem Sie sie austauschen. Wenn Sie das Netz nicht steuern, sind Sie einem Benutzer ausgesetzt, der Ihre IP-Übertragungen abfängt und als Endpunkt maskiert wird.

Sie können eine verschlüsselte TLS-Verbindung erstellen, ohne den Server zu authentifizieren, indem Sie einen Diffie-Hellman-Schlüsselaustausch CipherSpec verwenden, der anonymes TLS unterstützt. Der geheime, gemeinsam genutzte geheime Schlüssel zwischen Client und Server, der für die Verschlüsselung von TLS-Übertragungen verwendet wird, wird ohne Austausch eines privat signierten Serverzertifikats erstellt.

Da anonyme Verbindungen nicht sicher sind, werden die meisten TLS-Implementierungen nicht standardmäßig mit anonymen CipherSpecs verwendet. Wenn eine Clientanforderung für eine TLS-Verbindung von einem Telemetriekanal akzeptiert wird, muss der Kanal über einen Keystore verfügen, der durch eine

Kennphrase geschützt ist. Da TLS-Implementierungen nicht anonyme CipherSpecs verwenden, muss der Keystore standardmäßig ein privat signiertes Zertifikat enthalten, das der Client authentifizieren kann.

Wenn Sie anonyme CipherSpecs verwenden, muss der Server-Keystore vorhanden sein, aber er muss keine privat signierten Zertifikate enthalten.

Eine weitere Möglichkeit, eine verschlüsselte Verbindung herzustellen, besteht darin, den Trust-Provider auf dem Client durch Ihre eigene Implementierung zu ersetzen. Ihr Trust-Provider würde das Serverzertifikat nicht authentifizieren, aber die Verbindung würde verschlüsselt sein.



Achtung: Wenn Sie TLS mit MQTT verwenden, können Sie große Nachrichten verwenden. Es kann jedoch zu Leistungseinbußen kommen. MQTT ist für die Verarbeitung kleiner Nachrichten optimiert (normalerweise zwischen 1KB und 1MB groß).

Windows

Linux

AIX

TLS-Konfiguration von MQTT-Java-Clients und Telemetrienkanälen

Konfigurieren Sie TLS, um den Telemetrienkanal und den MQTT-Java-Client zu authentifizieren, und verschlüsseln Sie die Übertragung von Nachrichten zwischen ihnen. MQTT Java -Clients verwenden Java Secure Socket Extension (JSSE), um Telemetrienkanäle über TLS zu verbinden. Als Alternative zu SSL können bestimmte virtuelle private Netze (Virtual Private Network, VPN), z. B. IPsec, die Endpunkte einer TCP/IP-Verbindung authentifizieren. In einem VPN wird jedes IP-Paket, das im Netz übertragen wird, verschlüsselt. Sobald eine solche VPN-Verbindung hergestellt ist, haben Sie ein vertrauenswürdiges Netz aufgebaut. MQTT-Clients können auch mit TCP/IP über das VPN mit Telemetrienkanälen verbunden werden.

Sie können die Verbindung zwischen einem Java-MQTT-Client und einem Telemetrienkanal so konfigurieren, dass das TLS-Protokoll über TCP/IP verwendet wird. Die Art der Sicherung hängt davon ab, wie Sie TLS für die Verwendung von JSSE konfigurieren. Beginnend mit der sichersten Konfiguration können Sie drei verschiedene Sicherheitsstufen konfigurieren:

1. Erlauben Sie nur vertrauenswürdigen MQTT-Clients, eine Verbindung herzustellen. Verbinden Sie einen MQTT-Client nur mit einem vertrauenswürdigen Telemetrienkanal. Verschlüsseln Sie Nachrichten zwischen dem Client und dem Warteschlangenmanager (siehe [„MQTT-Clientauthentifizierung mit TLS“ auf Seite 230](#)).
2. Verbinden Sie einen MQTT-Client nur mit einem vertrauenswürdigen Telemetrienkanal. Verschlüsseln Sie Nachrichten zwischen dem Client und dem Warteschlangenmanager; siehe [„Telemetrienkanal-Authentifizierung mit TLS“ auf Seite 232](#).
3. Verschlüsseln Sie Nachrichten zwischen dem Client und dem Warteschlangenmanager; siehe [„Datenschutz auf Telemetrienkanälen“ auf Seite 234](#).

JSSE-Konfigurationsparameter

Ändern Sie die JSSE-Parameter so, dass die Art und Weise, wie eine TLS-Verbindung konfiguriert wird, Die JSSE-Konfigurationsparameter sind in drei Gruppen angeordnet:

1. [MQ Telemetry-Kanal](#)
2. [MQTT-Java-Client](#)
3. [JRE](#)

Konfigurieren Sie die Telemetrienkanalparameter mithilfe von IBM MQ Explorer. Legen Sie die Parameter des MQTT-Java-Clients im Attribut `MqttConnectionOptions.SSLProperties` fest. Ändern Sie die JRE-Sicherheitsparameter, indem Sie Dateien im JRE-Sicherheitsverzeichnis sowohl auf dem Client als auch auf dem Server bearbeiten.

MQ Telemetry-Kanal

Legen Sie alle TLS-Parameter des Telemetrienkanals mit IBM MQ Explorer fest.

ChannelName

ChannelName ist ein erforderlicher Parameter auf allen Kanälen.

Der Kanalname identifiziert den Kanal, der einer bestimmten Port-Nummer zugeordnet ist. Die Benennung von Kanälen hilft Ihnen bei der Verwaltung von Gruppen von MQTT-Clients.

PortNumber

PortNumber ist ein optionaler Parameter auf allen Kanälen. Der Standardwert ist 1883 für TCP-Kanäle und 8883 für TLS-Kanäle.

Die TCP/IP-Anschlussnummer, die diesem Kanal zugeordnet ist. MQTT-Clients werden mit einem Kanal verbunden, indem der für den Kanal definierte Port angegeben wird. Wenn der Kanal über TLS-Eigenschaften verfügt, muss der Client eine Verbindung mit dem TLS-Protokoll herstellen, z. B.:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId");
mqttClient.connect();
```

KeyFileName

KeyFileName ist ein erforderlicher Parameter für TLS-Kanäle. Er muss für TCP-Kanäle weggelassen werden.

KeyFileName gibt den Pfad zum Java-Schlüsselspeicher an, der digitale Zertifikate enthält, die Sie bereitstellen. Verwenden Sie JKS, JCEKS oder PKCS12 als den Typ des Schlüsselspeichers auf dem Server.

Geben Sie den Keystore-Typ mithilfe einer der folgenden Dateierweiterungen an:

- .jks
- .jceks
- .p12
- .pkcs12

Es wird angenommen, dass ein Schlüsselspeicher mit einer anderen Dateierweiterung ein JKS-Keystore ist.

Sie können einen Typ von Schlüsselspeicher auf dem Server mit anderen Keystoretypen auf dem Client kombinieren.

Speichern Sie das private Zertifikat des Servers in den Schlüsselspeicher. Das Zertifikat wird als Serverzertifikat bezeichnet. Das Zertifikat kann selbst signiert sein oder Teil einer Zertifikatskette sein, die von einer Signaturberechtigung signiert ist.

Wenn Sie eine Zertifikatskette verwenden, legen Sie die zugeordneten Zertifikate in den Serverschlüsselspeicher.

Das Serverzertifikat und alle Zertifikate in seiner Zertifikatskette werden an Clients gesendet, um die Identität des Servers zu authentifizieren.

Wenn Sie ClientAuth auf Required gesetzt haben, muss der Keystore alle Zertifikate enthalten, die für die Authentifizierung des Clients erforderlich sind. Der Client sendet ein selbst signiertes Zertifikat oder eine Zertifikatskette, und der Client wird durch die erste Überprüfung dieses Materials anhand eines Zertifikats im Keystore authentifiziert. Mit Hilfe einer Zertifikatskette kann ein Zertifikat viele Clients prüfen, auch wenn sie mit unterschiedlichen Clientzertifikaten ausgegeben werden.

PassPhrase

PassPhrase ist ein erforderlicher Parameter für TLS-Kanäle. Er muss für TCP-Kanäle weggelassen werden.

Der Verschlüsselungstext wird zum Schutz des Keystores verwendet.

ClientAuth

`ClientAuth` ist ein optionaler TLS-Parameter. Der Standardwert ist keine Clientauthentifizierung. Er muss für TCP-Kanäle weggelassen werden.

Legen Sie `ClientAuth` fest, wenn der Telemetrieservice (MQXR) den Client authentifizieren soll, bevor er dem Client die Verbindung zum Telemetriekanal ermöglicht.

Wenn Sie `ClientAuth` festlegen, muss der Client mit TLS eine Verbindung zum Server herstellen und den Server authentifizieren. Als Antwort auf die Einstellung von `ClientAuth` sendet der Client sein digitales Zertifikat an den Server und alle anderen Zertifikate in seinem Keystore. Das digitale Zertifikat wird als Clientzertifikat bezeichnet. Diese Zertifikate werden für Zertifikate authentifiziert, die im Channel-Keystore und im JRE- `cacerts` speicher gespeichert sind.

CipherSuite

`CipherSuite` ist ein optionaler TLS-Parameter. Standardmäßig werden alle aktivierten CipherSpecs probiert. Er muss für TCP-Kanäle weggelassen werden.

Wenn Sie eine bestimmte CipherSpec verwenden möchten, setzen Sie `CipherSuite` auf den Namen der CipherSpec, die zum Erstellen der TLS-Verbindung verwendet werden muss.

Der Telemetrieservice und der MQTT-Client vereinbaren eine allgemeine CipherSpec aus allen CipherSpecs, die an beiden Enden aktiviert sind. Wenn eine bestimmte CipherSpec an einem der beiden oder beiden Enden der Verbindung angegeben ist, muss sie mit der CipherSpec am anderen Ende übereinstimmen.

Installieren Sie zusätzliche Chiffriergeräte, indem Sie JSSE zusätzliche Provider hinzufügen.

Federal Information Processing Standards (FIPS)

FIPS ist eine optionale Einstellung. Standardmäßig ist sie nicht festgelegt.

Legen Sie entweder in der Eigenschaftenanzeige des Warteschlangenmanagers oder mithilfe von **`runmqsc`** `SSLFIPS` fest. `SSLFIPS` gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen.

Revocation namelist

Die Widerrufsnamensliste ist eine optionale Einstellung. Standardmäßig ist sie nicht festgelegt.

Legen Sie `SSLCRLNL` in der Eigenschaftenanzeige des Warteschlangenmanagers oder mithilfe von **`runmqsc`** fest. `SSLCRLNL` gibt eine Namensliste mit Authentifizierungsinformationsobjekten an, die zum Angeben von Zertifikatswiderrufspositionen verwendet werden.

Es werden keine anderen WS-Manager-Parameter verwendet, die TLS-Eigenschaften festlegen.

MQTT-Java-Client

Legen Sie TLS-Eigenschaften für den Java-Client in `MqttConnectionOptions.SSLProperties` fest; Beispiel:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

Die Namen und Werte bestimmter Eigenschaften werden in der Klasse `MqttConnectOptions` beschrieben. Links zur Client-API-Dokumentation für die MQTT-Clientbibliotheken finden Sie unter [MQTT client programming reference](#).

Protocol

Das Protokoll ist optional.

Das Protokoll wird in den Verhandlungen mit dem Telemetrieserver ausgewählt. Wenn Sie ein bestimmtes Protokoll benötigen, können Sie eine auswählen. Wenn der Telemetrieserver das Protokoll nicht unterstützt, schlägt die Verbindung fehl.

ContextProvider

ContextProvider ist optional.

KeyStore

KeyStore ist optional. Konfigurieren Sie sie, wenn ClientAuth auf dem Server festgelegt ist, um die Authentifizierung des Clients zu erzwingen.

Speichern Sie das digitale Zertifikat des Clients, signiert mit seinem privaten Schlüssel, in den Schlüsselspeicher. Geben Sie den Schlüsselspeicherpfad und das Kennwort an. Der Typ und der Provider sind optional. JKS ist der Standardtyp, und IBMJCE ist der Standardprovider.

Geben Sie einen anderen Schlüsselspeicherprovider an, um auf eine Klasse zu verweisen, die einen neuen Keystore-Provider hinzufügt. Übergeben Sie den Namen des Algorithmus, der vom Schlüsselspeicherprovider für die Instanziierung von KeyManagerFactory verwendet wird, indem Sie den Namen des Schlüsselmanagers festlegen.

TrustStore

TrustStore ist optional. Sie können alle Zertifikate, auf die Sie vertrauen, in den JRE- cacerts Speicher stellen.

Konfigurieren Sie den Truststore, wenn Sie einen anderen Truststore für den Client verwenden möchten. Sie können den Truststore möglicherweise nicht konfigurieren, wenn der Server ein Zertifikat verwendet, das von einer bekannten Zertifizierungsinstanz ausgestellt wurde, die bereits sein Stammzertifikat in cacerts gespeichert hat.

Fügen Sie das öffentlich signierte Zertifikat des Servers oder des Stammzertifikats zum Truststore hinzu, und geben Sie den Truststore-Pfad und das Kennwort an. JKS ist der Standardtyp, und IBMJCE ist der Standardprovider.

Geben Sie einen anderen Truststore-Provider an, um auf eine Klasse zu verweisen, die einen neuen Truststore-Provider hinzufügt. Übergeben Sie den Namen des Algorithmus, der vom Truststore-Provider für die Instanziierung der TrustManagerFactory verwendet wird, indem Sie den Namen des Trust-Managers festlegen.

JRE

Weitere Aspekte der Java-Sicherheit, die sich auf das Verhalten von TLS sowohl auf dem Client als auch auf dem Server auswirken, werden in der JRE konfiguriert. Die Konfigurationsdateien in Windows befinden sich in *Java Installation Directory*\jre\lib\security. Wenn Sie die JRE verwenden, die mit IBM MQ ausgeliefert wird, lautet der Pfad wie in der folgenden Tabelle angegeben:

Plattform	Dateipfad
Windows	<i>WMQ Installation Directory</i> \java\jre\lib\security
UNIX and Linux-Plattformen	<i>WMQ Installation Directory</i> /java/jre64/jre/lib/security

Well-bekanntes Zertifizierungsstellen

Die Datei cacerts enthält die Stammzertifikate anerkannter Zertifizierungsstellen. Der cacerts wird standardmäßig verwendet, es sei denn, Sie geben einen Truststore an. Wenn Sie den cacerts -Store verwenden oder keinen Truststore bereitstellen, müssen Sie die Liste der Unterzeichner in cacerts überprüfen und bearbeiten, um Ihre Sicherheitsanforderungen zu erfüllen.

Sie können cacerts mit dem IBM MQ-Befehl `stmqikm` öffnen, der das Dienstprogramm IBM Key Management ausführt. Öffnen Sie cacerts als JKS-Datei, indem Sie das Kennwort `changeit` verwenden. Ändern Sie das Kennwort, um die Datei zu sichern.

Sicherheitsklassen konfigurieren

Verwenden Sie die Datei `java.security`, um zusätzliche Sicherheitsprovider und andere Standardsicherheitseigenschaften zu registrieren.

Zulassungs-

Verwenden Sie die Datei `java.policy`, um die Berechtigungen zu ändern, die für Ressourcen erteilt wurden. `javaws.policy` erteilt Berechtigungen für `javaws.jar`.

Verschlüsselungsstärke

Einige JREs werden mit reduzierter Verschlüsselungsstärke ausgeliefert. Wenn Sie keine Schlüssel in Keystores importieren können, kann die Verschlüsselung mit verminderter Stärke die Ursache sein. Versuchen Sie, **ikeyman** mit dem Befehl **strmqikm** zu starten, oder laden Sie starke, aber Dateien mit beschränkter Zuständigkeit von [IBM Developer Kits, Security Information](#) herunter.

Wichtig: Ihr Herkunftsland hat möglicherweise Einschränkungen für den Import, den Besitz, die Verwendung oder den erneuten Export in ein anderes Land der Verschlüsselungssoftware zur Verfügung. Bevor Sie die unbeschränkten Richtliniendateien herunterladen oder verwenden, müssen Sie die Gesetze Ihres Landes überprüfen. Überprüfen Sie die Bestimmungen und die Richtlinien für den Import, den Besitz, die Verwendung und den Wiederexport von Verschlüsselungssoftware, um festzustellen, ob die Software zugelassen ist.

Ändern Sie den Trust-Provider, damit der Client eine Verbindung zu einem beliebigen Server herstellen kann.

Das Beispiel zeigt, wie ein Trust-Provider hinzugefügt und aus dem MQTT-Clientcode referenziert wird. Das Beispiel führt keine Authentifizierung für den Client oder den Server durch. Die resultierende TLS-Verbindung wird verschlüsselt, ohne dass sie authentifiziert wird.

Das Code-Snippet in [Abbildung 26 auf Seite 239](#) setzt den Trust-Provider `AcceptAllProviders` und den Trust Manager für den MQTT-Client.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

Abbildung 26. Code-Snippet MQTT-Client

```
package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
private static final long serialVersionUID = 1L;
public AcceptAllProvider() {
super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
put("TrustManagerFactory.TrustAllCertificates",
AcceptAllTrustManagerFactory.class.getName());
}
}
```

Abbildung 27. `AcceptAllProvider.java`

```
protected static class AcceptAllTrustManagerFactory extends
javax.net.ssl.TrustManagerFactorySpi {
public AcceptAllTrustManagerFactory() {}
protected void engineInit(java.security.KeyStore keystore) {}
protected void engineInit(
javax.net.ssl.ManagerFactoryParameters parameters) {}
protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
}
}
```

Abbildung 28. `AcceptAllTrustManagerFactory.java`

```

protected static class AcceptAllX509TrustManager implements
javax.net.ssl.X509TrustManager {
public void checkClientTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Client authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public void checkServerTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Server authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[0];
}
private static void report(String string) {
System.out.println(string);
}
}

```

Abbildung 29. *AcceptAllX509TrustManager.java*

Windows

Linux

AIX

JAAS-Konfiguration für Telemetrikanal

Konfigurieren Sie JAAS für die Authentifizierung des vom Client gesendeten Benutzername .

Der IBM MQ -Administrator konfiguriert, welche MQTT -Kanäle die Clientauthentifizierung mit JAASerfordern. Geben Sie den Namen einer JAAS-Konfiguration für jeden Kanal an, der die JAAS-Authentifizierung ausführen soll. Kanäle können alle dieselbe JAAS-Konfiguration verwenden, oder sie können verschiedene JAAS-Konfigurationen verwenden. Die Konfigurationen sind in *WMQData directory\mqmrs\qMgrName\mqxr\jaas.config* definiert.

Die Datei *jaas.config* ist nach dem JAAS-Konfigurationsnamen organisiert. Unter jedem Konfigurationsnamen befindet sich eine Liste mit Anmeldekongfigurationen; siehe [Abbildung 30](#) auf Seite 241.

JAAS stellt vier Standardanmeldemodule zur Verfügung. Die Standard-NT- und UNIX-Anmeldemodule haben einen begrenzten Wert.

JndiLoginModule

Authentifiziert einen Verzeichnisservice, der unter JNDI (Java Naming and Directory Interface) konfiguriert ist.

Krb5LoginModule

Authentifiziert die Verwendung von Kerberos-Protokollen.

NTLoginModule

Authentifiziert unter Verwendung der NT-Sicherheitsinformationen für den aktuellen Benutzer.

UnixLoginModule

Authentifiziert die Verwendung der UNIX-Sicherheitsinformationen für den aktuellen Benutzer.

Das Problem bei der Verwendung von *NTLoginModule* oder *UnixLoginModule* ist, dass der Telemetrieeservice (MQXR) mit der Identität *mqm* ausgeführt wird und nicht mit der Identität des MQTT-Kanals. *mqm* ist die Identität, die für die Authentifizierung an *NTLoginModule* oder *UnixLoginModule* übergeben wird, und nicht die Identität des Clients.

Um dieses Problem zu beheben, müssen Sie Ihr eigenes Anmeldemodul schreiben oder die anderen Standardanmeldemodule verwenden. Ein Beispiel für *JAASLoginModule.java* wird mit MQ Telemetry bereitgestellt. Es handelt sich um eine Implementierung der *javax.security.auth.spi.LoginModule* -Schnittstelle. Verwenden Sie diese Option, um Ihre eigene Authentifizierungsmethode zu entwickeln.

Alle neuen *LoginModule*-Klassen, die Sie bereitstellen, müssen sich auf dem Klassenpfad des Telemetrieeservice (MQXR) befinden. Stellen Sie Ihre Klassen nicht in IBM MQ-Verzeichnisse, die sich im Klassen-

pfad befinden. Erstellen Sie Ihre eigenen Verzeichnisse und definieren Sie den gesamten Klassenpfad für den Telemetrieservice (MQXR).

Sie können den Klassenpfad, der vom Telemetrieservice (MQXR) verwendet wird, erweitern, indem Sie den Klassenpfad in der Datei `service.env` definieren. `CLASSPATH` muss aktiviert sein, und die Klassenpfadanweisung kann nur Literale enthalten. Sie können keine Variablen in `CLASSPATH` verwenden, z. B. `CLASSPATH=%CLASSPATH%` ist nicht korrekt. Der Telemetrieservice (MQXR) legt seinen eigenen Klassenpfad fest. Der in `service.env` definierte `CLASSPATH` wird hinzugefügt.

Der Telemetrieservice (MQXR) stellt zwei Callbacks bereit, die den Benutzernamen und das Kennwort für einen Client zurückgeben, der mit dem MQTT-Kanal verbunden ist. Der Benutzername und das Kennwort werden im Objekt `MqttConnectOptions` festgelegt. Im Abschnitt [Abbildung 31](#) auf Seite [242](#) finden Sie ein Beispiel dafür, wie auf den Benutzernamen und das Kennwort zugegriffen wird.

Beispiele

Ein Beispiel für eine JAAS-Konfigurationsdatei mit einer benannten Konfiguration, `MQXRConfig`.

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//      principal=principal@your_realm
//      useDefaultCcache=TRUE
//      renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//      useTicketCache="true"
//      ticketCache="${user.home}/${}tickets";
};
```

Abbildung 30. Beispieldatei `jaas.config`

Beispiel eines JAAS-Anmeldemoduls, codiert für den Empfang des Benutzernamens und Kennworts, die von einem MQTT-Client bereitgestellt werden.

```

public boolean login()
throws javax.security.auth.login.LoginException {
    javax.security.auth.callback.Callback[] callbacks =
    new javax.security.auth.callback.Callback[2];
    callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
    callbacks[1] = new javax.security.auth.callback.PasswordCallback(
    "PasswordCallback", false);
    try {
        callbackHandler.handle(callbacks);
        String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
        .getName();
        char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
        .getPassword();
        // Accept everything.
        if (true) {
            loggedIn = true;
        } else
        throw new javax.security.auth.login.FailedLoginException("Login failed");

        principal= new JAASPrincipal(username);

    } catch (java.io.IOException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    } catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    }

    return loggedIn;
}

```

Abbildung 31. JAASLoginModule.Login() -Beispielmethode

Zugehörige Informationen

[AuthCallback-MQXR-Klasse](#)

Problembehebung: Das JAAS-Anmeldemodul wird vom Telemetrieservice nicht aufgerufen

V 9.0.0 IBM MQ Light verwalten

Sie können MQ Light mit IBM MQ Explorer oder über eine Befehlszeile verwalten. Verwenden Sie den Explorer, um Kanäle zu konfigurieren und die MQ Light-Clients zu überwachen, die mit IBM MQ verbunden sind. Konfigurieren Sie die Sicherheit von MQ Light mit TLS und JAAS.

Vor dem Start

Weitere Informationen zur Installation von AMQP auf Ihrer Plattform finden Sie unter [Zu installierende Installation](#). Installieren Sie die AMQP-Servicekomponente mithilfe der Produktaktualisierung IBM MQ 8.0.0.4, nicht mithilfe des Fixpacks 8.0.0.4. Sie können die AMQP-Komponente nicht in einer Version des Warteschlangenmanagers vor 8.0.0.4 installieren.

Verwaltung mit IBM MQ Explorer

Verwenden Sie den Explorer, um AMQP-Kanäle zu konfigurieren und die MQ Light-Clients zu überwachen, die mit IBM MQ verbunden sind. Sie können die Sicherheit von MQ Light mithilfe von TLS und JAAS konfigurieren.

Verwaltung über die Befehlszeile

Sie können MQ Light über die Befehlszeile mithilfe der IBM MQ [MQSC](#)-Befehle verwalten.

V 9.0.0 Von MQ Light-Clients verwendete IBM MQ-Objekte anzeigen

Sie können die verschiedenen IBM MQ-Ressourcen anzeigen, die von MQ Light-Clients verwendet werden, z. B. Verbindungen und Subskriptionen.

Verbindungen

Wenn der AMQP-Service gestartet wird, werden neue Hconns erstellt und mit dem WS-Manager verbunden. Dieser Pool von Hconns wird verwendet, wenn MQ Light-Clients Nachrichten veröffentlichen. Sie können die Hconns mit dem Befehl **DISPLAY CONN** anzeigen. Beispiel:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'WebSphere MQ Advanced Message Queuing Protocol*')
```

Dieser Befehl zeigt auch alle kundenspezifischen Hconns an. Die Hconns, die ein leeres Client-ID-Attribut haben, sind die Hconns, die im Pool verwendet werden.

Wenn ein MQ Light-Client eine Verbindung zu einem AMQP-Kanal herstellt, wird ein neuer Hconn mit dem Warteschlangenmanager verbunden. Dieser Hconn wird verwendet, um Nachrichten asynchron für die Subskriptionen zu konsumieren, die der MQ Light-Client erstellt hat. Sie können den Hconn, der von einem bestimmten MQ Light-Client verwendet wird, mit dem Befehl **DISPLAY CONN** anzeigen. Beispiel:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

Subskriptionen, die von Clients erstellt werden

Wenn ein MQ Light-Client ein Thema subskribiert, wird eine neue IBM MQ-Subskription erstellt. Der Subskriptionsname enthält die folgenden Informationen:

- Der Name des Clients. Wenn der Client einer gemeinsam genutzten Subskription beigetreten ist, wird der Name der Freigabe verwendet.
- Das Topic-Muster, das der Client abonniert hat
- Ein Präfix. Das Präfix ist `private`, wenn der Client eine nicht gemeinsam genutzte Subskription erstellt hat, oder `share`, wenn der Client mit einer gemeinsam genutzten Subskription verbunden ist.

Um die Subskriptionen anzuzeigen, die von einem bestimmten MQ Light -Client verwendet werden, führen Sie den Befehl **DISPLAY SUB** aus und filtern Sie nach dem Präfix `private`:

```
DISPLAY SUB(':private:*')
```

Um die gemeinsam genutzten Subskriptionen anzuzeigen, die von mehreren Clients verwendet werden, führen Sie den Befehl **DISPLAY SUB** aus und filtern Sie nach dem Präfix `share`:

```
DISPLAY SUB(':share:*')
```

Da gemeinsam genutzte Subskriptionen von mehreren MQ Light-Clients verwendet werden können, möchten Sie möglicherweise die Clients anzeigen, die derzeit Nachrichten aus der gemeinsam genutzten Subskription konsumieren. Sie können dies tun, indem Sie die Hconns auflisten, die derzeit über eine Kennung verfügen, die in der Subskriptionswarteschlange geöffnet ist. Führen Sie die folgenden Schritte aus, um die Clients anzuzeigen, die derzeit eine Freigabe verwenden:

1. Suchen Sie den Namen der Warteschlange, die die gemeinsam genutzte Subskription als Ziel verwendet. Beispiel:

```
DISPLAY SUB(':private:recv_e298452:public') DEST
5 : DISPLAY SUB(':private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D3120202020202020202020707E0A565C2D0020)
SUB(:private:recv_e298452:public)
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```

2. Führen Sie den Befehl **DISPLAY CONN** aus, um die Kennungen zu suchen, die in dieser Warteschlange geöffnet sind:

```
DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME
```

```

EQ SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
 21 : DISPLAY CONN(*) TYPE(HANDLE) WHERE(OBJNAME EQ
SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D312020202020202020)
TYPE(HANDLE)

OBJNAME(SYSTEM.BASE.TOPIC)      OBJTYPE(TOPIC)

OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(QUEUE)

```

3. Zeigen Sie für jede der Kennungen die MQ Light -Client-ID an, die die Kennung geöffnet hat:

```

DISPLAY CONN(707E0A56642B0020) CLIENTID
 23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D312020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
 24 : DISPLAY CONN(707E0A565F290020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D312020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)

```

V 9.0.0 MQ Light-Client - Identifikation, Berechtigung und Authentifizierung

Wie bei anderen IBM MQ-Clientanwendungen können Sie AMQP-Verbindungen auf mehrere Arten sichern.

Sie können die folgenden Sicherheitsfunktionen verwenden, um AMQP-Verbindungen zu IBM MQ zu sichern:

- [Kanalauthentifizierungsdatensätze](#)
- [Verbindungsauthentifizierung](#)
- Kanal-MCA-Benutzerkonfiguration
- IBM MQ-Berechtigungsdefinitionen
- [TLS-Konnektivität](#)

Aus der Sicherheitsperspektive besteht eine Verbindung aus den folgenden zwei Schritten:

- Entscheiden, ob die Verbindung fortgesetzt werden soll
- Entscheiden, welche IBM MQ-Identität die Anwendung für spätere Berechtigungsprüfungen annimmt

In den folgenden Informationen werden verschiedene IBM MQ-Konfigurationen und die Schritte beschrieben, die ausgeführt werden, wenn ein AMQP-Client versucht, eine Verbindung herzustellen. Nicht in allen IBM MQ-Konfigurationen werden alle beschriebenen Schritte verwendet. Einige Konfigurationen verwenden TLS beispielsweise nicht für Verbindungen innerhalb der Unternehmensfirewall und einige Konfigurationen verwenden TLS, verwenden aber keine Clientzertifikate für die Authentifizierung. In vielen Umgebungen werden keine angepassten oder angepassten JAAS-Module verwendet.

Verbindung aufbauen

In den folgenden Schritten wird beschrieben, was passiert, wenn eine Verbindung von einem AMQP-Client hergestellt wird. Die Schritte legen fest, ob die Verbindung fortgesetzt wird und welche IBM MQ-Identität die Anwendung für Berechtigungsprüfungen annimmt:

1. Wenn der Client eine TLS-Verbindung zu IBM MQ öffnet und ein Zertifikat bereitstellt, versucht der Warteschlangenmanager, das Clientzertifikat zu prüfen.
2. Wenn der Client Benutzernamen und Kennwortberechtigungsnaehweise bereitstellt, wird ein AMQP-SASL-Rahmen vom Warteschlangenmanager empfangen, und die MQ-CONNAUTH-Konfiguration wird überprüft.
3. Die Authentifizierungsregeln für den MQ-Kanal werden überprüft (z. B., ob die IP-Adresse und der TLS-Zertifikats-DN gültig sind).
4. Der Kanal MCAUSER wird zugesichert, sofern die Kanalaauthifizierungsregeln nichts anderes bestimmen.
5. Wenn ein JAAS-Modul konfiguriert wurde, wird es aufgerufen.
6. MQ-CONNECT-Berechtigungs-Prüfung für die resultierende MQ-Benutzer-ID.
7. Verbindung, die mit einer angenommenen IBM MQ-Identität aufgebaut wurde.

Nachricht veröffentlichen

In den folgenden Schritten wird beschrieben, was passiert, wenn eine Nachricht von einem AMQP-Client veröffentlicht wird. Die Schritte legen fest, ob die Verbindung fortgesetzt wird und welche IBM MQ-Identität die Anwendung für Berechtigungsprüfungen annimmt:

1. Der AMQP-Link-Zuordnungsrahmen wird beim WS-Manager empfangen. Die IBM MQ-Veröffentlichungsberechtigung für die angegebene Themenzeichenfolge wird auf die MQ-Benutzeridentität geprüft, die während der Verbindung hergestellt wurde.
2. Die Nachricht wird in der angegebenen Themenzeichenfolge veröffentlicht.

Abonnieren eines Topic-Musters

In den folgenden Schritten wird beschrieben, was passiert, wenn ein AMQP-Client ein Themenmuster subskribiert. Die Schritte legen fest, ob die Verbindung fortgesetzt wird und welche IBM MQ-Identität die Anwendung für Berechtigungsprüfungen annimmt:

1. Der AMQP-Link-Zuordnungsrahmen wird beim WS-Manager empfangen. Die IBM MQ-Subskriptionsberechtigung für das angegebene Themenmuster wird auf die MQ-Benutzeridentität geprüft, die während der Verbindung hergestellt wurde.
2. Die Subskription wird erstellt.

MQ Light-Client - Identität und Berechtigung

Verwenden Sie die MQ Light-Client-ID, den MQ Light-Benutzernamen oder eine allgemeine Clientidentität, die auf dem Kanal oder in einer Kanalaauthifizierungsregel definiert ist, um die Berechtigung für den Zugriff auf IBM MQ-Objekte zu erhalten.

Der Administrator kann bei der Definition oder Änderung des AMQP-Kanals die Auswahl treffen, indem er die Einstellung des Warteschlangenmanagers CONNAUTH oder die Kanalaauthifizierungsregeln definiert. Die Identität dient dazu, den Zugriff auf IBM MQ-Themen zu berechtigen. Die Auswahl erfolgt auf der Basis der folgenden:

1. Das Attribut "channel USECLNTID".
2. Das Attribut ADOPTCTX der Warteschlange CONNAUTH des Warteschlangenmanagers.
3. Das MCAUSER-Attribut, das auf dem Kanal definiert ist.
4. Das USERSRC-Attribut einer übereinstimmenden Kanalaauthifizierungsregel.

Vermeiden Sie Probleme.: Die Identität, die durch diesen Prozess ausgewählt wird, wird anschließend beispielsweise mit dem Befehl DISPLAY CHSTATUS (AMQP) als MCAUSER des Clients bezeichnet. Beachten Sie, dass dies nicht unbedingt die gleiche Identität wie der MCAUSER des Kanals ist, auf den in der Auswahl (2) verwiesen wird.

Wählen Sie mit dem IBM MQ-Befehl **setmqaut** aus, welche Objekte und welche Aktionen berechtigt sind, von der Identität verwendet zu werden, die dem AMQP-Kanal zugeordnet ist. Mit den folgenden Befehlen wird z. B. eine Kanal-ID `AMQPClient` autorisiert, die vom Administrator des WS-Managers `QM1` bereitgestellt wird:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPClient -all +pub +sub
```

und

```
setmqaut -m QM1 -t qmgr -p AMQPClient -all +connect
```

V 9.0.0 MQ Light-Clientauthentifizierung über ein Kennwort

Authentifizieren Sie den MQ Light-Benutzernamen mithilfe des Clientkennworts. Sie können den Client mit einer anderen Identität als der Identität authentifizieren, die verwendet wird, um den Client zu autorisieren, Themen zu veröffentlichen und zu abonnieren.

Der AMQP-Service kann MQ CONNAUTH oder JAAS verwenden, um den Clientbenutzernamen zu authentifizieren. Wenn einer dieser Parameter konfiguriert ist, wird das vom Client angegebene Kennwort von der MQ-CONNAUTH-Konfiguration oder dem JAAS-Modul überprüft.

In der folgenden Prozedur werden Beispielschritte für die Authentifizierung einzelner Benutzer gegenüber den lokalen Betriebssystembenutzern und -kennwörtern und bei erfolgreicher Übernahme der gemeinsamen Identität `AMQPUser` beschrieben:

1. Der IBM MQ-Administrator setzt die `MCAUSER`-Identität des AMQP-Kanals mithilfe von IBM MQ Explorer auf einen beliebigen Namen wie `AMQPUser`.
2. Der IBM MQ-Administrator berechtigt `AMQPUser`, Veröffentlichungen und Subskriptionen für alle Themen durchzuführen:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. Der IBM MQ-Administrator konfiguriert eine `IDPWOS CONNAUTH`-Regel, um den Benutzernamen und das Kennwort, die vom Client präsentiert werden, zu überprüfen. Die Regel `CONNAUTH` sollte `CHKCLNT (REQUIRED)` und `ADOPTCTX (NO)` setzen.

Anmerkung: Es wird empfohlen, die Kanalauthentifizierungsregeln zu verwenden und das `MCAUSER`-Kanalattribut einem Benutzer zu setzen, der über keine Berechtigungen verfügt, um mehr Kontrolle über die Verbindungen zum Warteschlangenmanager zu ermöglichen.

V 9.0.0 Datenschutz bei der Veröffentlichung von Kanälen

Die Vertraulichkeit von AMQP-Veröffentlichungen, die in beide Richtungen über die AMQP-Kanäle gesendet werden, wird durch die Verwendung von TLS zum Verschlüsseln von Übertragungen über die Verbindung gesichert.

AMQP-Clients, die eine Verbindung zu AMQP-Kanälen herstellen, verwenden TLS, um die Vertraulichkeit von Veröffentlichungen zu sichern, die auf dem Kanal mit Hilfe der symmetrischen Schlüsselverschlüsselungszeichen übertragen werden. Da die Endpunkte nicht authentifiziert sind, können Sie die Kanalschlüsselung nicht allein vertrauen. Kombinieren Sie den Schutz der Privatsphäre mit dem Server oder der gegenseitigen Authentifizierung.

Als Alternative zur Verwendung von TLS authentifizieren einige Arten von Virtual Private Network (VPN), wie z. B. IPsec, die Endpunkte einer TCP/IP-Verbindung. In einem VPN wird jedes IP-Paket, das im Netz übertragen wird, verschlüsselt. Sobald eine solche VPN-Verbindung hergestellt ist, haben Sie ein vertrauenswürdiges Netz aufgebaut. Sie können AMQP-Clients über TCP/IP über das VPN-Netz mit AMQP-Kanälen verbinden.

Das Verschlüsseln von TLS-Verbindungen ohne die Authentifizierung des Servers macht die Verbindung zu Man-in-the-Middle-Attacken verfügbar. Obwohl die Informationen, die Sie austauschen, vor Lauschangriffen geschützt sind, wissen Sie nicht, mit wem Sie sie austauschen. Wenn Sie das Netz nicht steuern, sind Sie einem Benutzer ausgesetzt, der Ihre IP-Übertragungen abfängt und als Endpunkt maskiert wird.

Sie können eine verschlüsselte TLS-Verbindung erstellen, ohne den Server zu authentifizieren, indem Sie einen Diffie-Hellman-Schlüsselaustausch CipherSpec verwenden, der anonymes TLS unterstützt. Der geheime, gemeinsam genutzte geheime Schlüssel zwischen Client und Server, der für die Verschlüsselung von TLS-Übertragungen verwendet wird, wird ohne Austausch eines privat signierten Serverzertifikats erstellt.

Da anonyme Verbindungen nicht sicher sind, werden die meisten TLS-Implementierungen nicht standardmäßig mit anonymen CipherSpecs verwendet. Wenn eine Clientanforderung für eine TLS-Verbindung von einem AMQP-Kanal akzeptiert wird, muss der Kanal über einen Keystore verfügen, der durch eine Kennphrase geschützt ist. Da TLS-Implementierungen nicht anonyme CipherSpecs verwenden, muss der Keystore standardmäßig ein privat signiertes Zertifikat enthalten, das der Client authentifizieren kann.

Wenn Sie anonyme CipherSpecs verwenden, muss der Server-Keystore vorhanden sein, aber er muss keine privat signierten Zertifikate enthalten.

Eine weitere Möglichkeit, eine verschlüsselte Verbindung herzustellen, besteht darin, den Trust-Provider auf dem Client durch Ihre eigene Implementierung zu ersetzen. Ihr Trust-Provider würde das Serverzertifikat nicht authentifizieren, aber die Verbindung würde verschlüsselt sein.

V 9.0.0 MQ Light-Clients mit TLS konfigurieren

Sie können MQ Light-Clients so konfigurieren, dass sie TLS verwenden, um Daten zu schützen, die im Netz fließen, und die Identität des Warteschlangenmanagers, zu dem der Client eine Verbindung herstellt, zu authentifizieren.

Wenn TLS für die Verbindung von einem MQ Light-Client zu einem AMQP-Kanal verwendet werden soll, müssen Sie sicherstellen, dass der Warteschlangenmanager für TLS konfiguriert wurde. [TLS auf WS-Managern konfigurieren](#) beschreibt, wie der Keystore konfiguriert wird, von dem ein Warteschlangenmanager TLS-Zertifikate liest.

Wenn der WS-Manager mit einem Keystore konfiguriert wurde, müssen Sie die TLS-Attribute im AMQP-Kanal konfigurieren, zu dem die Clients eine Verbindung herstellen. AMQP-Kanäle haben vier Attribute in Bezug auf die TLS-Konfiguration wie folgt:

SSLCAUTH

Das Attribut "SSLCAUTH" wird verwendet, um anzugeben, ob der Warteschlangenmanager einen MQ Light-Client benötigen muss, um ein Clientzertifikat zu präsentieren, um seine Identität zu überprüfen.

SSLCIPH

Das Attribut SSLCIPH gibt die Verschlüsselung an, die der Kanal zum Codieren von Daten in den TLS-Fluss verwenden soll.

SSLPEER

Das Attribut "SSLPEER" wird verwendet, um den definierten Namen (DN) anzugeben, den ein Clientzertifikat erfüllen muss, wenn eine Verbindung zulässig sein soll.

CERTLABL

Das CERTLABL gibt das Zertifikat an, das der WS-Manager dem Client präsentieren soll. Der Schlüsselpeicher des WS-Managers kann mehrere Zertifikate enthalten. Dieses Attribut ermöglicht es Ihnen, das Zertifikat anzugeben, das für Verbindungen zu diesem Kanal verwendet werden soll. Wenn kein CERTLABL angegeben ist, wird das Zertifikat im Schlüsselrepository des Warteschlangenmanagers mit dem Kennsatz des Warteschlangenmanagers CERTLABL-Attribut verwendet.

Wenn Sie Ihren AMQP-Kanal mit den TLS-Attributen konfiguriert haben, müssen Sie den AMQP-Service mit dem folgenden Befehl erneut starten:

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

Wenn ein MQ Light-Client eine Verbindung zu einem AMQP-Kanal herstellt, der durch TLS geschützt ist, überprüft der Client die Identität des Zertifikats, das vom Warteschlangenmanager vorgelegt wird. Dazu müssen Sie Ihren MQ Light-Client mit einem Truststore konfigurieren, der das Zertifikat des Warteschlangenmanagers enthält. Die erforderlichen Schritte hängen vom MQ Light-Client ab, den Sie verwenden.

- Die API-Dokumentation für den MQ Light-Client für Node JS finden Sie unter <https://www.npmjs.com/package/mqlight>.
- Die API-Dokumentation für den MQ Light-Client für Java finden Sie unter <https://mqlight.github.io/java-mqlight/>.
- Die Dokumentation für den MQ Light-Client für Ruby finden Sie unter <https://www.rubydoc.info/github/mqlight/ruby-mqlight/>.
- Die Dokumentation für den MQ Light-Client für Python finden Sie unter <https://python-mqlight.readthedocs.org/en/latest/>.

V 9.0.0 Verbindung zwischen MQ Light-Clients und dem Warteschlangenmanager trennen

Wenn Sie die Verbindung von MQ Light zum Warteschlangenmanager trennen möchten, führen Sie den Befehl `PURGE CHANNEL` aus oder stoppen Sie die Verbindung zum MQ Light-Client.

- Führen Sie den Befehl **PURGE CHANNEL** aus. Beispiel:

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- Stoppen Sie alternativ die Verbindung, die der MQ Light-Client verwendet, um die Verbindung zum Client zu trennen, indem Sie die folgenden Schritte ausführen:

1. Suchen Sie die Verbindung, die der Client verwendet, indem Sie den Befehl **DISPLAY CONN** ausführen. Beispiel:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

Die Befehlsausgabe lautet wie folgt:

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
  40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_28dbb7e)
```

2. Stoppen Sie die Verbindung. Beispiel:

```
STOP CONN(707E0A565F2D0020)
```

Multicast verwalten

Dieser Abschnitt enthält Informationen zu den IBM MQ Multicast-Verwaltungstasks, z. B. zum Reduzieren der Größe von Multicastnachrichten und Aktivieren der Datenkonvertierung.

Erste Schritte mit Multicasting

Verwenden Sie diese Informationen, um sich mit IBM MQ Multicast-Themen und Kommunikationsinformationsobjekten vertraut zu machen.

Informationen zu diesem Vorgang

Die IBM MQ Multicast-Nachrichtenübertragung verwendet das Netz, um Nachrichten zuzustellen, indem Gruppenadressen Themen zugeordnet werden. Mit den folgenden Tasks können Sie schnell testen, ob die erforderliche IP-Adresse und der erforderliche Port für Multicast-Messaging korrekt konfiguriert sind.

Erstellen eines **COMMINFO**-Objekts für Multicasting

Das Objekt Kommunikationsinformationen (COMMINFO) enthält die Attribute, die der Multicastübertragung zugeordnet sind. Weitere Informationen zu den Parametern des Befehls COMMINFO finden Sie in [DEFINE COMMINFO](#).

Verwenden Sie das folgende Befehlszeilenbeispiel, um ein COMMINFO-Objekt für Multicasting zu definieren:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

Hierbei steht *MC1* für den Namen Ihres COMMINFO-Objekts, *Gruppenadresse* für die Multicast-IP-Adresse oder den DNS-Namen der Gruppe, und *Portnummer* für den Port für die Übertragung (Der Standardwert ist 1414).

Es wird ein neues COMMINFO-Objekt mit dem Namen *MC1* erstellt. Dieser Name ist der Name, den Sie beim Definieren eines TOPIC-Objekts im nächsten Beispiel angeben müssen.

TOPIC-Objekt für Multicasting erstellen

Ein Thema ist der Gegenstand der Informationen, die in einer Publish/Subscribe-Nachricht veröffentlicht werden, und ein Thema wird definiert, indem ein TOPIC-Objekt erstellt wird. TOPIC-Objekte verfügen über zwei Parameter, die definieren, ob sie mit Multicasting verwendet werden können. Diese Parameter sind: **COMMINFO** und **MCAST**.

- **COMMINFO** Dieser Parameter gibt den Namen des Multicastkommunikationsinformationsobjekts an. Weitere Informationen zu den Parametern des Befehls COMMINFO finden Sie in [DEFINE COMMINFO](#).
- **MCAST** Dieser Parameter gibt an, ob Multicasting an dieser Position in der Themenstruktur zulässig ist.

Verwenden Sie das folgende Befehlszeilenbeispiel, um ein TOPIC-Objekt für Multicasting zu definieren:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

Es wird ein neues TOPIC-Objekt mit dem Namen *ALLSPORTS* erstellt. Sie hat eine Themenzeichenfolge *Sports*, das zugehörige Kommunikationsinformationsobjekt hat den Namen *MC1* (dies ist der Name, den Sie bei der Definition eines COMMINFO-Objekts im vorherigen Beispiel angegeben haben), und Multicasting ist aktiviert.

Testen des Multicast-Publish/Subscribe

Nachdem die Objekte "TOPIC" und "COMMINFO" erstellt wurden, können sie mit dem `amqspubc`-Beispiel und der `amqssubc`-Stichprobe getestet werden. Weitere Informationen zu diesen Beispielen finden Sie im Abschnitt [Publish/Subscribe-Beispielprogramme](#).

1. Öffnen Sie zwei Befehlszeilenfenster. Die erste Befehlszeile ist für das `amqspubc`-Publizierungsbeispiel und die zweite Befehlszeile für die `amqssubc`-Abonnementsprobe.
2. Geben Sie den folgenden Befehl in die Befehlszeile ein:

```
amqspubc Sports QM1
```

Hierbei steht *Sports* für die Themenzeichenfolge des in einem früheren Beispiel definierten TOPIC-Objekts und *QM1* für den Namen des Warteschlangenmanagers.

3. Geben Sie den folgenden Befehl in die Befehlszeile ein: 2:

Dabei sind *Sports* und *QM1* die gleichen wie in Schritt „2“ auf Seite 249.

- Geben Sie `Hallo Welt` in die Befehlszeile 1 ein. Wenn der Port und die IP-Adresse, die im `COMMINFO`-Objekt angegeben sind, ordnungsgemäß konfiguriert sind; das `amqssubc`-Beispiel, das an dem Port für Veröffentlichungen aus der angegebenen Adresse empfangsbereit ist, gibt `Hallo Welt` an der Befehlszeile 2 aus.

IBM MQ Multicast-Thementopologie

Dieses Beispiel soll das Verständnis der IBM MQ Multicast-Thementopologie erleichtern.

Die IBM MQ Multicast-Unterstützung erfordert, dass jede Unterverzeichnisstruktur eine eigene Multicastgruppe und einen eigenen Datenstrom in der gesamten Hierarchie hat.

Das IP-Adressierungsschema *classful network* umfasst einen designierten Adressraum für die Multicastadresse. Der vollständige IP-Adressenbereich für Multicasting liegt zwischen `224.0.0.0` und `239.255.255.255`, allerdings sind einige dieser Adressen reserviert. Eine Liste der reservierten Adressen erhalten Sie von dem zuständigen Systemadministrator; ebenso finden Sie weitere Informationen hierzu unter <https://www.iana.org/assignments/multicast-addresses>. Es wird empfohlen, die lokal bereichsorientierte Multicastadresse im Bereich von `239.0.0.0` bis `239.255.255.255` zu verwenden.

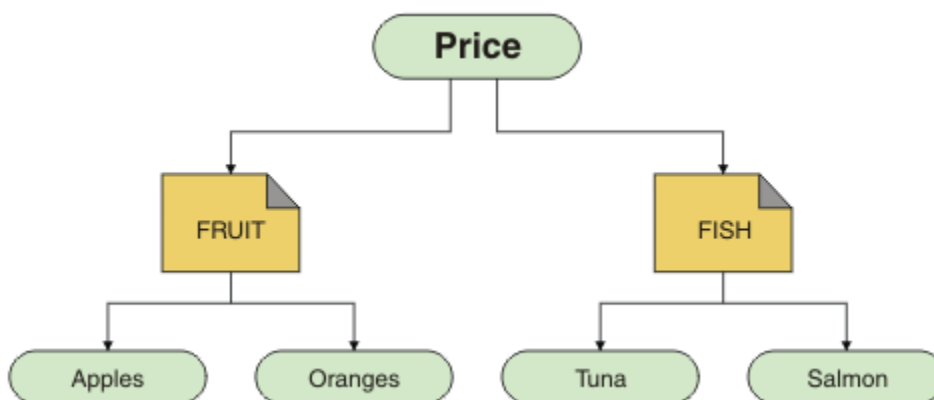
Im folgenden Diagramm gibt es zwei mögliche Multicastdatenströme:

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX)
)
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

Dabei sind `239.XXX.XXX.XXX` und `239.YYY.YYY.YYY` gültige Multicastadressen.

Diese Themendefinitionen werden verwendet, um eine Themenstruktur zu erstellen, wie im folgenden Diagramm dargestellt:

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



Jedes Multicast-Kommunikationsinformationsobjekt (`COMMINFO`) stellt einen anderen Datenstrom dar, da ihre Gruppenadressen unterschiedlich sind. In diesem Beispiel ist das Thema `FRUIT` für die Verwendung des `COMMINFO`-Objekts `MC1` definiert, das Thema `FISH` ist für die Verwendung von `COMMINFO`-Objekt `MC2` definiert, und der Knoten `Price` hat keine Multicastdefinitionen.

In IBM MQ Multicast sind Themenzeichenfolgen auf 255 Zeichen beschränkt. Diese Einschränkung bedeutet, dass für die Namen von Knoten und Blattknoten innerhalb der Baumstruktur gesorgt werden muss. Wenn die Namen von Knoten und Blattknoten zu lang sind, überschreitet die Themenzeichenfolge

möglicherweise 255 Zeichen und gibt den 2425 (0979) (RC2425): MQRC_TOPIC_STRING_ERROR -Ursachencode zurück. Es wird empfohlen, Themenzeichenfolgen so kurz wie möglich zu machen, da längere Themenzeichenfolgen sich nachteilig auf die Leistung auswirken können.

Größe von Multicastnachrichten steuern

Dieser Abschnitt enthält Informationen zum IBM MQ-Nachrichtenformat und darüber, wie die Größe von IBM MQ-Nachrichten reduziert werden kann.

IBM MQ-Nachrichten sind einer Reihe von Attributen zugeordnet, die im Nachrichtendeskriptor enthalten sind. Bei kleinen Nachrichten stellen diese Attribute den größten Teil des Datenverkehrs dar und können erhebliche negative Auswirkungen auf die Übertragungsgeschwindigkeit haben. IBM MQ Multicast ermöglicht es dem Benutzer, zu konfigurieren, welche Attribute (falls vorhanden) zusammen mit der Nachricht übertragen werden.

Das Vorhandensein von Nachrichtenattributen, die keine Themenzeichenfolge sind, hängt davon ab, ob das Objekt COMMINFO angibt, dass sie gesendet werden müssen oder nicht. Wenn ein Attribut nicht übertragen wird, wendet die empfangende Anwendung einen Standardwert an. Die MQMD-Standardwerte sind nicht notwendigerweise mit dem MQMD_DEFAULT-Wert identisch und werden später in Tabelle 17 auf Seite 252 beschrieben.

Das COMMINFO-Objekt enthält das Attribut MCPROP, das steuert, wie viele MQMD-Felder und Benutzereigenschaften mit der Nachricht fließen. Indem Sie für dieses Attribut einen geeigneten Wert festlegen, können Sie die Größe der IBM MQ Multicast-Nachrichten steuern:

MCPROP

Die Multicasteigenschaften steuern, wie viele der MQMD-Eigenschaften und Benutzereigenschaften mit der Nachricht fließen.

ALLE

Alle Benutzereigenschaften und alle Felder des MQMD werden übertragen.

REPLY

Nur Benutzereigenschaften und MQMD-Felder, die sich auf die Beantwortung der Nachrichten beziehen, werden übertragen. Diese Eigenschaften sind:

- MsgType
- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

BENUTZER

Es werden nur die Benutzereigenschaften übertragen.

KEINE

Es werden keine Benutzereigenschaften oder MQMD-Felder übertragen.

COMPAT

Dieser Wert bewirkt, dass die Übertragung der Nachricht in einem kompatiblen Modus zu RMM erfolgt, was eine gewisse Interoperation mit den aktuellen XMS-Anwendungen und IBM Integration Bus-RMM-Anwendungen ermöglicht.

Multicastnachrichten-Attribute

Nachrichtenattribute können aus verschiedenen Bereichen stammen, wie z. B. MQMD, die Felder in den MQRFH2- und Nachrichteneigenschaften.

Die folgende Tabelle zeigt, was geschieht, wenn Nachrichten gemäß dem Wert von MCPROP gesendet werden und der Standardwert verwendet wird, wenn ein Attribut nicht gesendet wird.

Tabelle 17. Messaging-Attribute und ihre Beziehungen zu Multicasting

Attribut	Aktion bei Verwendung von Multicasting	Standardwert, wenn nicht übertragen
TopicString	Immer eingeschlossen	Nicht zutreffend
MQMQ-Struktur-ID	Nicht übertragen	Nicht zutreffend
MQMD-Version	Nicht übertragen	Nicht zutreffend
Bericht	Wird eingeschlossen, wenn nicht der Standardwert	0
MsgType	Wird eingeschlossen, wenn nicht der Standardwert	MQMT_DATAGRAM
Verfall	Wird eingeschlossen, wenn nicht der Standardwert	0
Feedback	Wird eingeschlossen, wenn nicht der Standardwert	0
Encoding	Wird eingeschlossen, wenn nicht der Standardwert	MQENC_NORMAL (equiv)
CodedCharSetId	Wird eingeschlossen, wenn nicht der Standardwert	1208
Format	Wird eingeschlossen, wenn nicht der Standardwert	MQRFH2
Priority	Wird eingeschlossen, wenn nicht der Standardwert	4
Permanenz	Wird eingeschlossen, wenn nicht der Standardwert	MQPER_NOT_PERSISTENT
MsgId	Wird eingeschlossen, wenn nicht der Standardwert	Null
CorrelId	Wird eingeschlossen, wenn nicht der Standardwert	Null
BackoutCount	Wird eingeschlossen, wenn nicht der Standardwert	0
ReplyToQ	Wird eingeschlossen, wenn nicht der Standardwert	Leer
ReplyToQMgr	Wird eingeschlossen, wenn nicht der Standardwert	Leer
UserIdentifier	Wird eingeschlossen, wenn nicht der Standardwert	Leer
AccountingToken	Wird eingeschlossen, wenn nicht der Standardwert	Null
PutAppIType	Wird eingeschlossen, wenn nicht der Standardwert	MQAT_JAVA
PutAppIName	Wird eingeschlossen, wenn nicht der Standardwert	Leer
PutDate	Wird eingeschlossen, wenn nicht der Standardwert	Leer

Tabelle 17. Messaging-Attribute und ihre Beziehungen zu Multicasting (Forts.)

Attribut	Aktion bei Verwendung von Multicasting	Standardwert, wenn nicht übertragen
PutTime	Wird eingeschlossen, wenn nicht der Standardwert	Leer
ApplOriginData	Wird eingeschlossen, wenn nicht der Standardwert	Leer
GroupID	Ausgeschlossen	Nicht zutreffend
MsgSeqNumber	Ausgeschlossen	Nicht zutreffend
Offset	Ausgeschlossen	Nicht zutreffend
MsgFlags	Ausgeschlossen	Nicht zutreffend
OriginalLength	Ausgeschlossen	Nicht zutreffend
UserProperties	Eingeschlossen	Nicht zutreffend

Zugehörige Verweise

[ALTER COMMINFO](#)

Zugehörige Informationen

[DEFINE COMMINFO](#)

Datenkonvertierung für Multicast-Messaging aktivieren

In diesen Informationen erfahren Sie, wie die Datenkonvertierung für die IBM MQ-Multicast-Nachrichtenübertragung funktioniert.

IBM MQ Multicast ist ein gemeinsam genutztes, verbindungsunabhängiges Protokoll. Daher ist es nicht für jeden Client möglich, bestimmte Anforderungen an die Datenkonvertierung zu stellen. Jeder Client, der denselben Multicastdatenstrom subskribiert, empfängt dieselben Binärdaten; wenn eine Konvertierung von IBM MQ-Daten erforderlich ist, wird die Konvertierung deshalb lokal auf jedem Client ausgeführt.

In einer heterogenen Plattforminstallation kann es sein, dass die meisten Clients die Daten in einem Format benötigen, das nicht das native Format der Übertragungsanwendung ist. In dieser Situation können die Werte **CCSID** und **ENCODING** des Multicastobjekts **COMMINFO** verwendet werden, um die Codierung der Nachrichtenübertragung für die Effizienz zu definieren.

IBM MQ Multicast unterstützt die Datenkonvertierung der Nachrichtennutzdaten für folgende integrierte Formate:

- MQADMIN
- MQEVENT
- MQPCF
- MQRFH
- MQRFH2
- MQSTR

Zusätzlich zu diesen Formaten können Sie auch eigene Formate definieren und einen Datenkonvertierungsexit [MQDXP-Data-Conversion Exit](#) Parameter verwenden.

Informationen zur Programmierung von Datenkonvertierungen finden Sie unter [Datenkonvertierung in der MQI für Multicast-Messaging](#).

Weitere Informationen zur Datenkonvertierung finden Sie unter [Datenkonvertierung](#).

Weitere Informationen zu Datenkonvertierungsexits und `ClientExitPath` finden Sie in der Datei [ClientExitPath-Zeilengruppe](#) der Clientkonfigurationsdatei.

Multicast-Anwendungsüberwachung

Dieser Abschnitt enthält Informationen zur Verwaltung und Überwachung von IBM MQ Multicast.

Der Status der aktuellen Publisher und Subskribenten für den Multicast-Datenverkehr (z. B. die Anzahl der gesendeten und empfangenen Nachrichten oder die Anzahl der verlorenen Nachrichten) wird regelmäßig vom Client an den Server übertragen. Wenn der Status empfangen wird, gibt das Attribut **COMMEV** des **COMMINFO**-Objekts an, ob der Warteschlangenmanager eine Ereignisnachricht in die Warteschlange **SYSTEM.ADMIN.PUBSUB.EVENT** einreicht. Die Ereignisnachricht enthält die empfangenen Statusinformationen. Diese Informationen sind eine unschätzbare Diagnose-Hilfe bei der Suche nach der Ursache eines Problems.

Verwenden Sie den MQSC-Befehl **DISPLAY CONN** , um Verbindungsinformationen zu den Anwendungen anzuzeigen, die mit dem Warteschlangenmanager verbunden sind. Weitere Informationen zum Befehl **DISPLAY CONN** finden Sie in [DISPLAY CONN](#) .

Verwenden Sie den MQSC-Befehl **DISPLAY TPSTATUS** , um den -Status Ihrer Publisher und Subskribenten anzuzeigen. Weitere Informationen zum Befehl **DISPLAY TPSTATUS** finden Sie in [DISPLAY TPSTATUS](#) .

COMMEV und der Anzeiger für die Zuverlässigkeit der Multicastnachricht

Der *Zuverlässigkeitsindikator*, der in Verbindung mit dem Attribut **COMMEV** des Objekts **COMMINFO** verwendet wird, ist ein Schlüsselement bei der Überwachung von IBM MQ Multicast-Publishern und -Subskribenten. Der Zuverlässigkeitsindikator (das Feld **MSGREL**, das in den Publish- oder Subscribe-Statusbefehlen zurückgegeben wird) ist ein IBM MQ-Indikator, der den Prozentsatz der Übertragungen angibt, die keine Fehler aufweisen. Manchmal müssen Nachrichten aufgrund eines Übertragungsfehlers erneut übertragen werden, was der Wert von **MSGREL** widerspiegelt. Mögliche Ursachen von Übertragungsfehlern sind langsame Subskribenten, ausgelastete Netze und Netzausfälle. **COMMEV** steuert, ob Ereignisnachrichten für Multicastkennungen generiert werden, die mit dem Objekt **COMMINFO** erstellt werden, und wird auf einen von drei möglichen Werten gesetzt:

INAKTIVIERT

Ereignisnachrichten werden nicht geschrieben.

ENABLED

Ereignisnachrichten werden immer mit einer Häufigkeit geschrieben, die im Parameter **COMMINFO MONINT** definiert ist.

EXCEPTION

Ereignisnachrichten werden geschrieben, wenn die Nachrichtenzuverlässigkeit unter dem Zuverlässigkeitsschwellenwert liegt. Eine Nachrichtenzuverlässigkeitsstufe von 90% oder weniger zeigt an, dass ein Problem mit der Netzkonfiguration aufgetreten ist oder dass eine oder mehrere der Publish/Subscribe-Anwendungen zu langsam ausgeführt werden:

- Der Wert **MSGREL (100, 100)** gibt an, dass es weder in der Kurzzeit noch im Langzeitrahmen zu Problemen gekommen ist.
- Der Wert **MSGREL (80, 60)** gibt an, dass 20% der Nachrichten derzeit Probleme haben, aber es ist auch eine Verbesserung für den Langzeitwert von 60.

Clients können die Übertragung und den Empfang von Multicastverkehr auch dann fortsetzen, wenn die Unicast-Verbindung zum Warteschlangenmanager unterbrochen ist. Daher sind die Daten möglicherweise nicht auf dem neuesten Stand.

Multicast-Nachrichtenzuverlässigkeit

In diesem Abschnitt erfahren Sie, wie das Subskriptions- und Nachrichtenprotokoll von IBM MQ Multicast eingestellt wird.

Ein Schlüsselement für die Überwindung von Übertragungsfehlern bei Multicast ist die Pufferung von übertragenen Daten (ein Nachrichtenprotokoll, das am sendenden Ende der Verbindung vorgehalten wird) durch IBM MQ. Dies bedeutet, dass beim Anwendungsprozess der Einreichung keine Pufferung von Nachrichten erforderlich ist, weil IBM MQ die Zuverlässigkeit bereitstellt. Die Größe dieses Protokolls wird über

das Kommunikationsinformationsobjekt (COMMINFO) konfiguriert, wie in den folgenden Informationen beschrieben. Ein größerer Übertragungspuffer bedeutet, dass bei Bedarf mehr Übertragungsprotokoll übertragen wird, aber aufgrund der Art der Multicast-Übertragung kann eine 100% garantierten Zustellung nicht unterstützt werden.

Das IBM MQ Multicast-Nachrichtenprotokoll wird im Kommunikationsinformationsobjekt (COMMINFO) durch das Attribut **MSGHIST** gesteuert:

MSGHIST

Dieser Wert ist die Menge an Nachrichtenprotokollen in Kilobyte, die vom System zur Bearbeitung erneuter Übertragungen im Falle negativer Rückmeldungen behalten wird.

Der Wert 0 gibt den niedrigsten Grad an Zuverlässigkeit an. Der Standardwert ist 100 KB.

Das neue IBM MQ Multicast-Subskriptionsprotokoll wird im Kommunikationsinformationsobjekt (COMMINFO) durch das Attribut **NSUBHIST** gesteuert:

NSUBHIST

Das neue Abonnentenprotokoll steuert, ob ein Abonnent, der an einem Veröffentlichungs-Stream teilnimmt, so viele Daten wie aktuell verfügbar empfängt, oder ob er nur Veröffentlichungen empfängt, die seit dem Zeitpunkt der Subskription erstellt wurden.

KEINE

Der Wert NONE bewirkt, dass der Sender nur die Veröffentlichung sendet, die aus dem Zeitpunkt der Subskription erstellt wurde. NONE ist der Standardwert.

ALL

Der Wert ALL bewirkt, dass der Sender so viel Protokoll des Themas erneut überträgt, wie bekannt ist. Unter bestimmten Umständen kann diese Situation ein ähnliches Verhalten für ständige Veröffentlichungen liefern.

Anmerkung: Die Verwendung des Werts ALL kann sich unter Umständen bei einem umfangreichen Themenverlauf nachteilig auf die Leistung auswirken, da der gesamte Verlauf erneut übertragen wird.

Zugehörige Informationen

DEFINE COMMINFO

ALTER COMMINFO

Erweiterte Multicast-Tasks

In diesen Informationen finden Sie Informationen zu erweiterten Tasks für die Multicastverwaltung von IBM MQ, wie zum Beispiel die Konfiguration von `.ini`-Dateien und die Interoperabilität mit IBM MQ LLM.

Hinweise zur Sicherheit in einer Multicast-Installation finden Sie im Abschnitt [Multicast-Sicherheit](#).

Überbrückung zwischen Multicast-und Nicht-Multicast-Publish/Subscribe-Domänen

In diesem Abschnitt erfahren Sie, was passiert, wenn ein Nicht-Multicast-Publisher eine Veröffentlichung zu einem IBM MQ Multicast-fähigen Thema durchführt.

Wenn ein Nicht-Multicast-Publisher in einem Thema veröffentlicht, das als **MCAST** aktiviert ist und **BRIDGE** aktiviert ist, überträgt der Warteschlangenmanager die Nachricht direkt an alle Subskribenten, die empfangsbereit sein könnten, über Multicasting. Ein Multicast-Publisher kann nicht in Themen veröffentlichen, die nicht Multicasting-fähig sind.

Vorhandene Themen können Multicasting-fähig sein, indem Sie die Parameter **MCAST** und **COMMINFO** eines Themenobjekts festlegen. Weitere Informationen zu diesen Parametern finden Sie unter [Ursprüngliche Multicastkonzepte](#).

Das Attribut COMMINFO-Objekt **BRIDGE** steuert Veröffentlichungen von Anwendungen, die keine Multicast-Anwendungen verwenden. Wenn **BRIDGE** auf `ENABLED` gesetzt ist und der Parameter **MCAST** des

Themas ebenfalls auf `ENABLED` gesetzt ist, werden Veröffentlichungen von Anwendungen, die nicht Multicast verwenden, zu Anwendungen überbrückt, die nicht verwendet werden. Weitere Informationen zum Parameter **BRIDGE** finden Sie in [DEFINE COMMINFO](#).

INI-Dateien für Multicasting konfigurieren

Verwenden Sie diese Informationen, um die Multicastfelder von IBM MQ in den `.ini`-Dateien zu verstehen.

Zusätzliche IBM MQ Multicast-Konfigurationen können in einer `ini`-Datei vorgenommen werden. Die spezielle `ini`-Datei, die Sie verwenden müssen, hängt von der Art der Anwendung ab:

- Client: Konfigurieren Sie die `MQ_DATA_PATH/mqclient.ini`-Datei.
- Warteschlangenmanager: Konfigurieren Sie die `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini`-Datei.

Dabei steht `MQ_DATA_PATH` für die Position des IBM MQ -Datenverzeichnisses (`/var/mqm/mqclient.ini`) und `QMNAME` für den Namen des Warteschlangenmanagers, für den die `.ini`-Datei gilt.

Die Datei `.ini` enthält Felder, die verwendet werden, um das Verhalten von IBM MQ Multicast zu optimieren:

```
Multicast:
Protocol      = IP | UDP
IPVersion     = IPv4 | IPv6 | ANY | BOTH
LimitTransRate = DISABLED | STATIC | DYNAMIC
TransRateLimit = 100000
SocketTTL    = 1
Batch        = NO
Loop        = 1
Interface    = <IPaddress>
FeedbackMode = ACK | NACK | WAIT1
HeartbeatTimeout = 20000
HeartbeatInterval = 2000
```

Protokoll

UDP

In diesem Modus werden Pakete mit Hilfe des UDP-Protokolls gesendet. Netzelemente können keine Unterstützung in der Multicastverteilung bereitstellen, wie sie im IP-Modus ausgeführt werden. Das Paketformat bleibt mit PGM kompatibel. Dies ist der Standardwert.

IP

In diesem Modus sendet der Sender unformatierte IP-Pakete. Netzelemente mit PGM-Unterstützung helfen bei der zuverlässigen Multicastpaketverteilung. Dieser Modus ist vollständig kompatibel mit dem PGM-Standard.

IPVersion

IPv4

Kommunikation nur über das IPv4-Protokoll. Dies ist der Standardwert.

IPv6

Kommunikation nur über das IPv6-Protokoll.

ANY

Kommunikation über IPv4 und/oder IPv6, abhängig davon, welches Protokoll verfügbar ist.

BOTH

Unterstützt die Kommunikation über IPv4 und IPv6.

LimitTransRate

INAKTIVIERT

Es gibt keine Übertragungsraten-Steuerung. Dies ist der Standardwert.

STATISCH

Implementiert die Steuerung der statischen Übertragungsgeschwindigkeit. Der Sender würde nicht mit einer Rate übertragen, die die durch den Parameter `TransRateLimit` angegebene Rate überschreitet.

DYNAMIC

Der Sender passt seine Übertragungsrate entsprechend dem Feedback, das es von den Empfängern erhält, an. In diesem Fall darf die Übertragungsgeschwindigkeit nicht größer sein als der durch den Parameter `TransRateLimit` angegebene Wert. Der Sender versucht, eine optimale Übertragungsrate zu erreichen.

TransRateLimit

Die Übertragungsgeschwindigkeit in Kbps.

SocketTTL

Der Wert von `SocketTTL` bestimmt, ob der Multicast-Datenverkehr einen Router passieren kann, oder die Anzahl der Router, die er passieren kann.

Batch

Steuert, ob Nachrichten gesendet oder sofort gesendet werden. Es gibt zwei mögliche Werte:

- *NO* Die Nachrichten werden nicht in der Stapelverarbeitung gesendet, sondern sofort gesendet.
- *YES* Die Nachrichten werden stapelweise ausgeführt.

Schleife

Setzen Sie den Wert auf `1`, um die Multicastschleife zu aktivieren. Die Multicastschleife definiert, ob die gesendeten Daten an den Host zurückgeschleift werden oder nicht.

Schnittstelle

Die IP-Adresse der Schnittstelle, auf der Multicastdatenverkehr fließt. Weitere Informationen und Hinweise zur Fehlerbehebung finden Sie in den Abschnitten [Multicastanwendungen auf einem Nicht-Multicast-Netz testen](#) und [Geeignetes Netz für Multicastverkehr festlegen](#).

Feedbackmodus

NACK

Rückmeldung durch negative Bestätigungen. Dies ist der Standardwert.

ACK

Rückmeldung durch positive Bestätigungen.

WAIT1

Rückmeldungen durch positive Bestätigungen, bei denen der Sender nur auf `1` ACK von einem der Empfänger wartet.

HeartbeatTimeout

Das Zeitlimit für die Überwachungssignale in Millisekunden. Der Wert `0` gibt an, dass die Ereignisse des Heartbeat-Zeitlimits vom Empfänger oder Empfänger des Themas nicht ausgelöst werden. Der Standardwert ist `20000`.

HeartbeatInterval

Das Intervall der Überwachungssignale in Millisekunden. Der Wert `0` gibt an, dass keine Überwachungssignale gesendet werden. Das Intervall der Überwachungssignale muss deutlich kleiner als der Wert von **HeartbeatTimeout** sein, um falsche Zeitlimitüberschreitungen bei Überwachungssignalen zu vermeiden. Der Standardwert ist `2000`.

Multicast-Interoperabilität mit IBM MQ Low Latency Messaging

Verwenden Sie diese Informationen, um die Interoperabilität zwischen IBM MQ Multicast und IBM MQ Low Latency Messaging (LLM) zu verstehen.

Eine einfache Übertragung von Nutzdaten ist für eine Anwendung mit LLM möglich, wobei eine andere Anwendung Multicast verwendet, um Nachrichten in beide Richtungen auszutauschen. Obwohl Multicast die LLM-Technologie verwendet, ist das LLM-Produkt selbst nicht eingebettet. Daher ist es möglich, sowohl LLM als auch IBM MQ Multicast zu installieren und die beiden Produkte separat zu betreiben und zu bedienen.

LLM-Anwendungen, die mit Multicasting kommunizieren, müssen möglicherweise Nachrichteneigenschaften senden und empfangen. Die IBM MQ-Nachrichteneigenschaften und MQMD-Felder werden wie in der folgenden Tabelle dargestellt als LLM-Nachrichteneigenschaften mit bestimmten LLM-Nachrichteneigenschaftencodes übertragen:

Tabelle 18. IBM MQ-Nachrichteneigenschaften für Eigenschaftszuordnung von IBM MQ LLM

IBM MQ-Eigenschaft	IBM MQ LLM-Eigenschaftstyp	LLM-Eigenschaftentart	LLM-Eigenschaftscode
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1013
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

Weitere Informationen zu LLM finden Sie in der LLM-Produktdokumentation: [IBM MQ Low Latency Messaging](#).

IBM i **IBM MQ for IBM i verwalten**

Hier finden Sie eine Einführung in die Methoden, die zur Verwaltung von IBM MQ unter IBM i verfügbar sind.

Zu den Verwaltungstasks gehören das Erstellen, Starten, Ändern, Anzeigen, Stoppen und Löschen von Clustern, Prozessen und IBM MQ-Objekten (Warteschlangenmanager, Warteschlangen, Namenslisten, Prozessdefinitionen, Kanäle, Clientverbindungskanäle, Listener, Services und Authentifizierungsinformationsobjekte).

Details zur Verwaltung von IBM MQ for IBM i finden Sie unter den folgenden Links:

- [„IBM MQ for IBM i mithilfe von CL-Befehlen verwalten“](#) auf Seite 259
- [„Alternative Methoden zur Verwaltung von IBM MQ for IBM i“](#) auf Seite 273
- [„Ablaufsteuerung für IBM i“](#) auf Seite 278

Zugehörige Konzepte

[„Verfügbarkeit, Sicherung, Wiederherstellung und Neustart unter IBM i“](#) auf Seite 287

Verwenden Sie diese Informationen, um zu verstehen, wie IBM MQ for IBM i die Journalunterstützung für IBM i verwendet, um die Sicherungs- und Zurückschreibungsstrategie zu unterstützen.

Zugehörige Verweise

„IBM MQ for IBM i in den Quiescemodus versetzen“ auf Seite 333

In diesem Abschnitt wird beschrieben, wie IBM MQ for IBM i in den Quiescemodus (ordnungsgemäße Beendigung) versetzt wird.

Zugehörige Informationen

[Konfigurationsdaten unter IBM i ändern](#)

[Hinweise zu Namen von Warteschlangenmanagerbibliotheken von IBM MQ for IBM i](#)

[Sicherheit unter IBM i einrichten](#)

[Steerroutine der Warteschlange für nicht zustellbare Nachrichten \(DLQ-Handler\) unter IBM i](#)

[Problembestimmung bei IBM MQ for IBM i-Anwendungen](#)

[Installierbare Services und Komponenten unter IBM i](#)

[System- und Standardobjekte unter IBM i](#)

IBM i IBM MQ for IBM i mithilfe von CL-Befehlen verwalten

Dieser Abschnitt enthält Informationen zu den IBM MQ-Befehlen unter IBM i.

Die meisten Gruppen von IBM MQ-Befehlen, einschließlich der Befehle in Verbindung mit Warteschlangenmanagern, Warteschlangen, Themen, Kanälen, Namenslisten, Prozessdefinitionen und Authentifizierungsinformationsobjekten, können mit dem betreffenden **WRK***-Befehl aufgerufen werden.

Der Hauptbefehl in der Gruppe ist **WRKMQM**. Mit diesem Befehl können Sie z. B. eine Liste aller Warteschlangenmanager auf dem System sowie Statusinformationen anzeigen. Alternativ können Sie alle WS-Manager-spezifischen Befehle mit verschiedenen Optionen für jeden Eintrag verarbeiten.

Über den Befehl **WRKMQM** können Sie bestimmte Bereiche jedes Warteschlangenmanagers auswählen, z. B. mit Kanälen, Themen oder Warteschlangen arbeiten und von dort einzelne Objekte auswählen.

IBM MQ-Anwendungsdefinitionen aufzeichnen

Wenn Sie IBM MQ-Anwendungen erstellen oder anpassen, ist es sinnvoll, einen Datensatz mit allen erstellten IBM MQ-Definitionen aufzubewahren. Dieser Datensatz kann für folgende Schritte verwendet werden:

- Wiederherstellungszwecke
- Wartung
- Rollout von IBM MQ-Anwendungen durchführen

Es gibt zwei Methoden, um IBM MQ-Anwendungsdefinitionen aufzuzeichnen:

1. Erstellen von CL-Programmen zum Generieren der IBM MQ-Definitionen für den Server
2. Erstellen von MQSC-Textdateien als SRC-Member zum Generieren der IBM MQ-Definitionen mithilfe der plattformunabhängigen IBM MQ-Befehlssprache.

Weitere Informationen zur Definition von Warteschlangenobjekten finden Sie in den Abschnitten „Scriptbefehle (MQSC)“ auf Seite 11 und „IBM MQ Programmable Command Formats verwenden“ auf Seite 23.

Zugehörige Informationen

[IBM MQ for IBM i CL-Befehlsreferenz](#)

IBM i Vorbereitungen für die Verwendung von IBM MQ for IBM i mit CL-Befehlen

Dieser Abschnitt enthält Informationen zum Starten des IBM MQ-Subsystems und Erstellen eines lokalen Warteschlangenmanagers.

Vorbereitende Schritte

Stellen Sie sicher, dass das IBM MQ-Subsystem aktiv ist (mit dem Befehl STRSBS QMQM/QMQM) und die Jobwarteschlange, die dem Subsystem zugeordnet ist, nicht blockiert ist. Standardmäßig haben sowohl das IBM MQ-Subsystem als auch die Jobwarteschlange in der Bibliothek QMQM den Namen QMQM.

Informationen zu diesem Vorgang

Warteschlangenmanager über die IBM i-Befehlszeile starten

Vorgehensweise

1. Erstellen Sie einen lokalen Warteschlangenmanager, indem Sie den Befehl CRTMQM in einer IBM i-Befehlszeile ausgeben.
Wenn Sie einen WS-Manager erstellen, haben Sie die Möglichkeit, diesen Warteschlangenmanager als Standardwarteschlangenmanager zu verwenden. Der Standardwarteschlangenmanager (von dem es nur einen WS-Manager geben kann) ist der Warteschlangenmanager, auf den ein CL-Befehl angewendet wird, wenn der Parameter des Warteschlangenmanagers (MQMNAME) weggelassen wird.
2. Starten Sie einen lokalen Warteschlangenmanager, indem Sie den Befehl STRMQM in einer IBM i-Befehlszeile ausgeben.
Wenn der Start des Warteschlangenmanagers länger als ein paar Sekunden dauert, zeigt IBM MQ immer wieder Statusnachrichten zum Fortschritt des Startvorgangs an. Weitere Informationen zu diesen Nachrichten finden Sie unter [Nachrichten und Ursachencodes](#).

Nächste Schritte

Sie können einen Warteschlangenmanager stoppen, indem Sie den Befehl ENDMQM in der IBM i-Befehlszeile ausgeben, und einen Warteschlangenmanager steuern, indem Sie andere IBM MQ-Befehle über eine IBM i-Befehlszeile absetzen.

Ferne WS-Manager können nicht über Remotezugriff gestartet werden, sondern müssen in ihren Systemen von lokalen Operatoren erstellt und gestartet werden. Eine Ausnahme gilt dort, wo ferne Betriebseinrichtungen (außerhalb von IBM MQ for IBM i) vorhanden sind, um solche Operationen zu ermöglichen.

Der lokale WS-Administrator kann einen fernen Warteschlangenmanager nicht stoppen.

Anmerkung: Im Rahmen der Stilllegung eines IBM MQ-Systems müssen Sie die aktiven Warteschlangenmanager in den Quiescemodus versetzen. Weitere Informationen hierzu finden Sie unter [„IBM MQ for IBM i in den Quiescemodus versetzen“](#) auf Seite 333.

IBM i IBM MQ for IBM i-Objekte erstellen

Dieser Abschnitt enthält Informationen zu den Methoden zum Erstellen von IBM MQ-Objekten für IBM i.

Vorbereitende Schritte

Im Folgenden werden verschiedene Möglichkeiten der Verwendung von IBM MQ for IBM i über die Befehlszeile vorgeschlagen.

Informationen zu diesem Vorgang

Es gibt zwei Onlinemethoden zum Erstellen von IBM MQ-Objekten:

Vorgehensweise

1. Verwenden Sie einen Erstellungsbefehl, z. B. den Befehl **Create MQM Queue : CRTMQMQ**
2. Verwendung des Befehls "Work with MQM object", gefolgt von F6, z. B.: Der Befehl **Work with MQM Queues : WRKMQMQ**

Nächste Schritte

Eine Liste aller Befehle finden Sie in [IBM MQ for IBM i CL-Befehle](#).

Anmerkung: Alle MQM-Befehle können über das Menü "Message Queue Manager Commands" übergeben werden. Um dieses Menü anzuzeigen, geben Sie `GO CMDMQM` in die Befehlszeile ein, und drücken Sie die Taste `Enter`.

Das System zeigt die Bedienerführungsanzeige automatisch an, wenn Sie einen Befehl in diesem Menü auswählen. Um die Bedienerführungsanzeige für einen Befehl anzuzeigen, den Sie direkt in der Befehlszeile eingegeben haben, drücken Sie `F4`, bevor Sie den `Enter`-Schlüssel drücken.

Erstellen einer lokalen Warteschlange mit dem Befehl CRTMQMQ

Vorgehensweise

1. Geben Sie in der Befehlszeile `CHGMQM` ein und drücken Sie die Taste `F4`.
2. Geben Sie in der Anzeige **Create MQM Queue** (MQM-Warteschlange erstellen) den Namen der Warteschlange ein, die Sie im Feld `Queue name` erstellen möchten. Wenn Sie einen Namen in Groß-/ Kleinschreibung angeben möchten, müssen Sie den Namen in Hochkommas einschließen.
3. Geben Sie `*LCL` im Feld `Queue type` ein.
4. Geben Sie einen WS-Manager-Namen an, es sei denn, Sie verwenden den Standardwarteschlangenmanager und drücken Sie die Taste `Enter`. Sie können einen beliebigen der Werte mit einem neuen Wert überschreiben. Blättern Sie vorwärts, um weitere Felder anzuzeigen. Die für Cluster verwendeten Optionen befinden sich am Ende der Liste der Optionen.
5. Wenn Sie alle Werte geändert haben, drücken Sie die `Enter`-Taste, um die Warteschlange zu erstellen.

Erstellen einer lokalen Warteschlange mit dem Befehl WRKMQM

Vorgehensweise

1. Geben Sie in der Befehlszeile `WRKMQM` ein.
2. Geben Sie den Namen eines Warteschlangenmanagers ein.
3. Wenn Sie die Bedienerführungsanzeige anzeigen möchten, drücken Sie `F4`. Die Bedienerführungsanzeige ist nützlich, um die Anzahl der angezeigten Warteschlangen zu reduzieren, indem Sie einen generischen Warteschlangennamen oder einen generischen Warteschlangentyp angeben.
4. Drücken Sie `Enter` und die Anzeige **Mit MQM-Warteschlangen arbeiten** wird angezeigt. Sie können jeden der Werte mit einem neuen Wert überschreiben. Blättern Sie vorwärts, um weitere Felder anzuzeigen. Die für Cluster verwendeten Optionen befinden sich am Ende der Liste der Optionen.
5. Drücken Sie `F6`, um eine neue Warteschlange zu erstellen. Dies führt Sie zur Anzeige **CRTMQMQ**. Anweisungen zum Erstellen der Warteschlange finden Sie im Abschnitt „Erstellen einer lokalen Warteschlange mit dem Befehl CRTMQMQ“ auf Seite 261. Wenn Sie die Warteschlange erstellt haben, wird die Anzeige **Mit MQM-Warteschlangen arbeiten** erneut angezeigt. Die neue Warteschlange wird der Liste hinzugefügt, wenn Sie `F5=Refresh` drücken.

Ändern von WS-Managerattributen

Informationen zu diesem Vorgang

Um die Attribute des im Befehl `CHGMQM` angegebenen Warteschlangenmanagers zu ändern, geben Sie die Attribute und Werte an, die Sie ändern möchten. Verwenden Sie z. B. die folgenden Optionen, um die Attribute von `jupiter.queue.manager` zu ändern:

Vorgehensweise

Geben Sie in der Befehlszeile `CHGMQM` ein und drücken Sie die Taste `F4`.

Ergebnisse

Der Befehl ändert die verwendete Warteschlange für dead-letter und aktiviert die Sperrereignisse.

IBM i Unter IBM i mit lokalen Warteschlangen arbeiten

Dieser Abschnitt enthält Beispiele für einige der Befehle, die Sie zum Verwalten lokaler Warteschlangen verwenden können. Alle angezeigten Befehle sind auch über die Optionen in der **WRKMQMQ-Befehlsanzeige** verfügbar.

Lokale Warteschlange definieren

Bei einer Anwendung ist der lokale WS-Manager der Warteschlangenmanager, mit dem die Anwendung verbunden ist. Warteschlangen, die vom lokalen WS-Manager verwaltet werden, werden als lokale Warteschlangen für diesen Warteschlangenmanager angegeben.

Verwenden Sie den Befehl **CRTMQMQ QTYPE *LCL**, um eine Definition einer lokalen Warteschlange zu erstellen und um auch die Datenstruktur zu erstellen, die als Warteschlange bezeichnet wird. Sie können die Warteschlangenkenndaten auch von denen der lokalen Standardwarteschlange ändern.

Im folgenden Beispiel wird die definierte Warteschlange `orange.local.queue` mit den folgenden Merkmalen angegeben:

- Es ist aktiviert für Abrufe, ist für die Auswahl inaktiviert und wird auf einer FIFO-Basis (First In First Out) ausgeführt.
- Es handelt sich um eine *normale* Warteschlange, d. a. es handelt sich nicht um eine Initialisierungswarteschlange oder eine Übertragungswarteschlange, und es werden keine Auslösenachrichten generiert.
- Die maximale Warteschlangenlänge beträgt 1000 Nachrichten; die maximale Nachrichtenlänge beträgt 2000 Byte.

Der folgende Befehl führt dies auf dem Standardwarteschlangenmanager aus:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL)
TEXT('Queue for messages from other systems')
PUTENBL(*NO)
GETENBL(*YES)
TRGENBL(*NO)
MSGDLYSEQ(*FIFO)
MAXDEPTH(1000)
MAXMSGLEN(2000)
USAGE(*NORMAL)
```

Anmerkung:

1. USAGE *NORMAL gibt an, dass es sich bei dieser Warteschlange nicht um eine Übertragungswarteschlange handelt.
2. Wenn bereits eine lokale Warteschlange mit dem Namen `orange.local.queue` auf demselben WS-Manager vorhanden ist, schlägt dieser Befehl fehl. Verwenden Sie das Attribut REPLACE *YES, wenn die vorhandene Warteschlangendefinition überschrieben werden soll. Lesen Sie dazu auch [„Attribute der lokalen Warteschlange ändern“](#) auf Seite 263.

Warteschlange für dead-Mail definieren

Jeder WS-Manager muss über eine lokale Warteschlange verfügen, die als Warteschlange für einen dead-letter verwendet werden muss, damit Nachrichten, die nicht an ihr korrektes Ziel zugestellt werden können, für einen späteren Abruf gespeichert werden können. Sie müssen dem WS-Manager explizit die Warteschlange für dead-letter mitteilen. Sie können dies tun, indem Sie eine Warteschlange für nicht zustellbare Nachrichten im Befehl **CRTMQMQ** angeben, oder Sie können den Befehl **CHGMQM** verwenden, um später eine Warteschlange anzugeben. Sie müssen auch die Warteschlange für dead-letter definieren, bevor sie verwendet werden kann.

Im Lieferumfang des Produkts ist eine Beispiel-Warteschlange mit dem Namen 'dead-letter' mit dem Namen SYSTEM.DEAD.LETTER.QUEUE enthalten. Diese Warteschlange wird bei der Erstellung des Warteschlangenmanagers automatisch erstellt. Sie können diese Definition bei Bedarf ändern. Es gibt keine Notwendigkeit, es umzubenennen, obwohl Sie können, wenn Sie möchten.

Eine Warteschlange für nicht zustellbare Nachrichten hat keine besonderen Anforderungen mit Ausnahme der folgenden:

- Es muss sich um eine lokale Warteschlange handeln.
- Das Attribut MAXMSGL (maximale Nachrichtenlänge) muss die Warteschlange aktivieren, damit die größten Nachrichten, die der WS-Manager mit **plus** der Größe des Headers dead-letter (MQDLH) handhaben muss, berücksichtigt werden können.

IBM MQ stellt einen DLQ-Handler (Steuerroutine der Warteschlange für nicht zustellbare Nachrichten) bereit, mit dem Sie angeben können, wie Nachrichten, die in einer Warteschlange für nicht zustellbare Nachrichten gefunden werden, verarbeitet oder entfernt werden sollen. Weitere Informationen finden Sie im Abschnitt [DLQ-Handler von IBM MQ for IBM i](#).

Standardobjektattribute anzeigen

Wenn Sie ein IBM MQ-Objekt definieren, werden alle Attribute, die Sie nicht angeben, aus dem Standardobjekt verwendet. Wenn Sie beispielsweise eine lokale Warteschlange definieren, übernimmt die Warteschlange alle Attribute, die Sie in der Definition aus der lokalen Standardwarteschlange übergehen, die als SYSTEM.DEFAULT.LOCAL.QUEUE bezeichnet wird. Um genau zu sehen, welche Attribute diese Attribute haben, verwenden Sie den folgenden Befehl:

```
DSPMQMQ QNAME(SYSTEM.DEFAULT.LOCAL.QUEUE) MQMNAME(MYQUEUEMANAGER)
```

Lokale Warteschlangendefinition kopieren

Sie können eine Warteschlangendefinition mit dem Befehl CPYMQMQ kopieren. Beispiel:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Dieser Befehl erstellt eine Warteschlange mit den gleichen Attributen wie die ursprüngliche Warteschlange orange.local.queue und nicht die Warteschlange der lokalen Standardwarteschlange.

Sie können auch den Befehl **CPYMQMQ** verwenden, um eine Warteschlangendefinition zu kopieren, aber eine oder mehrere Änderungen an den Attributen des Originals ersetzen. Beispiel:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)  
MAXMSGLEN(1024)
```

Dieser Befehl kopiert die Attribute der Warteschlange orange.local.queue in die Warteschlange third.queue, gibt jedoch an, dass die maximale Nachrichtenlänge in der neuen Warteschlange 1024 Byte und nicht 2000 betragen soll.

Anmerkung: Wenn Sie den Befehl **CPYMQMQ** verwenden, kopieren Sie nur die Warteschlangenattribute, nicht die Nachrichten in der Warteschlange.

Attribute der lokalen Warteschlange ändern

Sie können Warteschlangenattribute auf zwei Arten ändern, indem Sie entweder den Befehl **CHGMQMQ** oder den Befehl **CPYMQMQ** mit dem Attribut REPLACE *YES verwenden. In „[Lokale Warteschlange definieren](#)“ auf Seite 262 haben Sie die Warteschlange orange.local.queue definiert. Wenn Sie zum Beispiel die maximale Nachrichtenlänge in dieser Warteschlange auf 10.000 Byte erhöhen müssen.

- Mit dem Befehl **CHGMQM** :

```
CHGMQM QNAME('orange.local.queue') MQMNAME(MYQUEUEEMANAGER) MAXMSGLEN(10000)
```

Mit diesem Befehl wird ein einzelnes Attribut geändert, das der maximalen Nachrichtenlänge entspricht. Alle anderen Attribute bleiben unverändert.

- Verwendung des Befehls **CRTMQM** mit der Option REPLACE *YES, z. B.:

```
CRTMQM QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEEMANAGER)
MAXMSGLEN(10000) REPLACE(*YES)
```

Dieser Befehl ändert nicht nur die maximale Nachrichtenlänge, sondern auch alle anderen Attribute, die ihre Standardwerte erhalten. Die Warteschlange ist jetzt aktiviert, während sie zuvor gesperrt wurde. Put aktiviert ist der Standardwert, wie er in der Warteschlange SYSTEM.DEFAULT.LOCAL.QUEUE angegeben ist, es sei denn, Sie haben ihn geändert.

Wenn Sie die maximale Nachrichtenlänge für eine vorhandene Warteschlange **erhöhen**, sind vorhandene Nachrichten nicht davon betroffen. Alle neuen Nachrichten müssen jedoch die neuen Kriterien erfüllen.

Löschen einer lokalen Warteschlange

Wenn Sie alle Nachrichten aus einer lokalen Warteschlange mit dem Namen magenta.queue löschen möchten, verwenden Sie den folgenden Befehl:

```
CLRMQM QNAME('magenta.queue') MQMNAME(MYQUEUEEMANAGER)
```

Sie können eine Warteschlange nicht löschen, wenn:

- Es sind nicht festgeschriebene Nachrichten vorhanden, die unter Synchronisationspunkt in die Warteschlange gestellt wurden.
- Eine Anwendung hat momentan die Warteschlange geöffnet.

Lokale Warteschlange löschen

Mit dem Befehl **DLTMQM** können Sie eine lokale Warteschlange löschen.

Eine Warteschlange kann nicht gelöscht werden, wenn sie nicht festgeschriebene Nachrichten enthält oder wenn sie im Gebrauch ist.

Große Warteschlangen aktivieren

IBM MQ unterstützt Warteschlangen, die größer als 2 GB sind. Weitere Informationen zum Aktivieren von IBM i für die Unterstützung großer Dateien finden Sie in der Dokumentation zum Betriebssystem.

Sie finden die IBM iProduktokumentation in [IBM Documentation](#).

Einige Dienstprogramme sind möglicherweise nicht in der Lage, Dateien mit mehr als 2 GB zu bewältigen. Überprüfen Sie vor der Aktivierung der Unterstützung für große Dateien die Dokumentation zum Betriebssystem, um Informationen zu Einschränkungen für diese Unterstützung zu erhalten.

Unter IBM i mit Aliaswarteschlangen arbeiten

Dieser Abschnitt enthält Beispiele für einige der Befehle, die Sie zum Verwalten von Aliaswarteschlangen verwenden können. Alle angezeigten Befehle sind auch über die Optionen in der **WRKMQM-Befehlsanzeige** verfügbar.

Eine Aliaswarteschlange (auch bekannt als Aliasname der Warteschlange) stellt eine Methode zur Umleitung von MQI-Aufrufen bereit. Eine Aliaswarteschlange ist keine echte Warteschlange, sondern eine

Definition, die in eine echte Warteschlange aufgelöst wird. Die Definition der Aliaswarteschlange enthält einen Zielwarteschlangennamen, der durch das Attribut TGTQNAME angegeben wird.

Wenn eine Anwendung eine Aliaswarteschlange in einem MQI-Aufruf angibt, löst der WS-Manager den Namen der realen Warteschlange während der Ausführung auf.

Zum Beispiel wurde eine Anwendung entwickelt, um Nachrichten in eine Warteschlange mit dem Namen `my.alias.queue` zu stellen. Sie gibt den Namen dieser Warteschlange an, wenn sie eine **MQOPEN** -Anforderung und indirekt eine Nachricht in diese Warteschlange eingibt. Der Anwendung ist nicht bekannt, dass es sich bei der Warteschlange um eine Aliaswarteschlange handelt. Für jeden MQI-Aufruf, der diesen Aliasnamen verwendet, löst der Warteschlangenmanager den Namen der realen Warteschlange auf, die entweder eine lokale Warteschlange oder eine ferne Warteschlange sein kann, die in diesem WS-Manager definiert ist.

Wenn Sie den Wert des Attributs TGTQNAME ändern, können Sie MQI-Aufrufe in eine andere Warteschlange umleiten, möglicherweise auf einem anderen Warteschlangenmanager. Dies ist hilfreich bei der Wartung, Migration und Lastverteilung.

Definieren einer Aliaswarteschlange

Mit dem folgenden Befehl wird eine Aliaswarteschlange erstellt:

```
CRTMQMQ QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
MQMNAME(MYQUEUEEMANAGER)
```

Mit diesem Befehl werden MQI-Aufrufe, die `my.alias.queue` in der Warteschlange `yellow.queue` angeben, umgeleitet. Der Befehl erstellt die Zielwarteschlange nicht; die MQI-Aufrufe schlagen fehl, wenn die Warteschlange `yellow.queue` zur Ausführungszeit nicht vorhanden ist.

Wenn Sie die Aliasdefinition ändern, können Sie die MQI-Aufrufe an eine andere Warteschlange umleiten. Beispiel:

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEEMANAGER)
```

Mit diesem Befehl werden MQI-Aufrufe an eine andere Warteschlange (`magenta.queue`) umgeleitet.

Sie können auch Aliaswarteschlangen verwenden, um eine einzelne Warteschlange (die Zielwarteschlange) zu erstellen, um unterschiedliche Attribute für verschiedene Anwendungen zu haben. Definieren Sie dies, indem Sie zwei Aliasnamen definieren, eine für jede Anwendung. Angenommen, es gibt zwei Anwendungen:

- Die Anwendung ALPHA kann Nachrichten in `yellow.queue` einlegen, aber es ist nicht zulässig, Nachrichten von ihr abzurufen.
- Anwendung BETA kann Nachrichten von `yellow.queue` abrufen, darf jedoch keine Nachrichten in diese Datei einlegen.

Sie können dies mit den folgenden Befehlen tun:

```
/* This alias is put enabled and get disabled for application ALPHA */
CRTMQMQ QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQMQ QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEEMANAGER)
```

ALPHA verwendet den Warteschlangennamen `alphas.alias.queue` in seinen MQI-Aufrufen; BETA verwendet den Warteschlangennamen `betas.alias.queue`. Beide greifen auf die gleiche Warteschlange zu, aber auf unterschiedliche Weise.

Sie können das Attribut REPLACE *YES verwenden, wenn Sie Aliaswarteschlangen auf dieselbe Weise definieren, wie Sie diese Attribute mit lokalen Warteschlangen verwenden.

Andere Befehle mit Aliaswarteschlangen verwenden

Sie können die entsprechenden Befehle verwenden, um Attribute der Aliaswarteschlange anzuzeigen oder zu ändern. Beispiel:

```
* Display the alias queue's attributes */
DSPMQMQ QNAME('alphas.alias.queue') MQMNAME(MYQUEUEEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */

CHQMCMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEEMANAGER)
```

IBM i

Unter IBM i mit Modellwarteschlangen arbeiten

Dieser Abschnitt enthält Beispiele für einige der Befehle, die Sie zum Verwalten von Modellwarteschlangen verwenden können. Alle angezeigten Befehle sind auch über die Optionen in der **WRKMCMQ-Befehlsanzeige** verfügbar.

Ein Warteschlangenmanager erstellt eine dynamische Warteschlange, wenn er von einer Anwendung einen MQI-Aufruf empfängt, der einen Warteschlangennamen angibt, der als Modellwarteschlange definiert wurde. Der Name der neuen dynamischen Warteschlange wird vom WS-Manager beim Erstellen der Warteschlange generiert. Eine Modellwarteschlange ist eine Schablone, die die Attribute aller dynamischen Warteschlangen angibt, die aus dieser Warteschlange erstellt wurden.

Modellwarteschlangen stellen eine komfortable Methode für Anwendungen bereit, Warteschlangen zu erstellen, da sie erforderlich sind.

Modellwarteschlange definieren

Sie definieren eine Modellwarteschlange mit einer Gruppe von Attributen in derselben Weise, wie Sie eine lokale Warteschlange definieren. Modellwarteschlangen und lokale Warteschlangen verfügen über dieselbe Gruppe von Attributen, mit der Ausnahme, dass Sie in Modellwarteschlangen angeben können, ob die erstellten dynamischen Warteschlangen temporär oder permanent sind. (Permanente Warteschlangen werden über WS-Manager-Neustarts hinweg gepflegt, temporäre Warteschlangen sind nicht vorhanden). Beispiel:

```
CRTMQMQ QNAME('green.model.queue') QTYPE(*MDL) DFNTYPE(*PERMDYN)
```

Mit diesem Befehl wird eine Modellwarteschlangendefinition erstellt. Aus dem Attribut DFNTYPE sind die tatsächlichen Warteschlangen, die aus dieser Vorlage erstellt werden, permanente dynamische Warteschlangen. Die Attribute, die nicht angegeben sind, werden automatisch aus der Standardwarteschlange `SYSYSTEM.DEFAULT.MODEL.QUEUE` kopiert.

Sie können das Attribut REPLACE *YES verwenden, wenn Sie Modellwarteschlangen auf die gleiche Weise definieren, wie Sie sie mit lokalen Warteschlangen verwenden.

Andere Befehle mit Modellwarteschlangen verwenden

Sie können die entsprechenden Befehle verwenden, um die Attribute einer Modellwarteschlange anzuzeigen oder zu ändern. Beispiel:

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEEMANAGER) QNAME('green.model.queue')
```

```
/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQM QMQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

IBM i **Unter IBM i mit Auslösern arbeiten**

Verwenden Sie diese Informationen, um Informationen zu Auslöserdefinitionen und Prozessdefinitionen zu erhalten.

IBM MQ stellt eine Funktion bereit, um eine Anwendung automatisch zu starten, wenn bestimmte Bedingungen in einer Warteschlange erfüllt sind. Ein Beispiel für die Bedingungen ist, wenn die Anzahl der Nachrichten in einer Warteschlange eine angegebene Zahl erreicht. Diese Funktion wird als *Auslösefunktion* bezeichnet und wird in [Auslöserkanäle](#) detailliert beschrieben.

Das Prinzip der Auslösefunktion

Der WS-Manager definiert bestimmte Bedingungen als Auslöserereignisse. Wenn die Auslösung für eine Warteschlange aktiviert ist und ein Auslöserereignis eintritt, sendet der Warteschlangenmanager eine Auslösenachricht an eine Warteschlange, die als Initialisierungswarteschlange bezeichnet wird. Das Vorhandensein der Auslösenachricht in der Initialisierungswarteschlange zeigt an, dass ein Auslöserereignis aufgetreten ist.

Vom Warteschlangenmanager generierte Auslösenachrichten sind keine persistenten Nachrichten. Dies hat zur Folge, dass die Protokollierung reduziert wird (wodurch die Leistung verbessert wird) und die Duplikate während des Neustarts minimiert werden, so dass die Wiederanlaufzeiten verbessert werden.

Was ist der Auslösemonitor?

Das Programm, das die Initialisierungswarteschlange verarbeitet, wird als Auslösemonitoranwendung bezeichnet, und seine Funktion besteht darin, die Auslösenachricht zu lesen und basierend auf den in der Auslösenachricht enthaltenen Informationen die entsprechende Aktion zu ergreifen. Normalerweise wird diese Aktion ausgeführt, um eine andere Anwendung zu starten, um die Warteschlange zu verarbeiten, die dazu führte, dass die Auslösenachricht generiert wurde. Aus der Sicht des Warteschlangenmanagers gibt es nichts Besonderes an der Auslösemonitoranwendung-es handelt sich um eine andere Anwendung, die Nachrichten aus einer Warteschlange liest (Initialisierungswarteschlange).

Ändern der Jobübergabeattribute des Auslösemonitors

Der Auslösemonitor, der als Befehl **STRMQMTRM** angegeben wurde, übergibt einen Job für jede Auslösenachricht unter Verwendung der Systemstandardjobbeschreibung QDFTJOB. Dies hat die Einschränkung, dass die übergebenen Jobs immer als QDFTJOB bezeichnet werden und die Attribute der Standardjobbeschreibung einschließlich der Bibliotheksliste *SYSVAL haben. IBM MQ stellt eine Methode zum Überschreiben dieser Attribute bereit. Es ist z. B. möglich, die übergebenen Jobs so anzupassen, dass sie aussagekräftigere Jobnamen wie folgt haben:

1. Geben Sie in der Jobbeschreibung die Beschreibung an, die Sie benötigen, z. B. Protokollierungswerte.
2. Geben Sie die Umgebungsdaten der Prozessdefinition an, die im Auslöserprozess verwendet wird:

```
CHGMQM PRCNAME(MY_PROCESS) QMQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

Der Auslösemonitor führt eine SBMJOB mit der angegebenen Beschreibung aus.

Es ist möglich, andere Attribute des SBMJOB zu überschreiben, indem Sie das entsprechende Schlüsselwort und den entsprechenden Wert in der Umgebungsdaten der Prozessdefinition angeben. Die einzige Ausnahme hiervon ist das Schlüsselwort **CMD**, da dieses Attribut vom Auslösemonitor gefüllt wird.

Beispiel für den Befehl zum Angeben der Umgebungsdaten der Prozessdefinition, in dem sowohl der Jobname als auch die Beschreibung geändert werden sollen:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)
JOB(TRIGGER)')
```

Anwendungswarteschlange zum Auslösen definieren

Eine Anwendungswarteschlange ist eine lokale Warteschlange, die von Anwendungen für die Nachrichtenübertragung über die MQI verwendet wird. Für die Triggerung ist es erforderlich, dass eine Reihe von Warteschlangenattributen in der Anwendungswarteschlange definiert wird. Die Triggerung selbst wird durch das Attribut TRGENBL aktiviert.

In diesem Beispiel soll ein Auslöserereignis generiert werden, wenn 100 Nachrichten mit Priorität 5 oder höher in der lokalen Warteschlange `motor.insurance.queue` vorhanden sind, wie folgt:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)
PRCNAME('motor.insurance.quote.process') MAXMSGLEN(2000)
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')
TRGENBL(*YES) TRGTYPE(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

Es folgt eine Auflistung der Parameter:

MQMNAME(MYQUEUEMANAGER)

Der Name des Warteschlangenmanagers.

QNAME('motor.insurance.queue')

Der Name der Anwendungswarteschlange, die definiert wird.

PRCNAME('motor.insurance.quote.process')

Der Name der Anwendung, die von einem Auslösemonitorprogramm gestartet werden soll.

MAXMSGLEN(2000)

Die maximale Länge der Nachrichten in der Warteschlange.

DFTMSGPST(*YES)

Nachrichten in dieser Warteschlange sind standardmäßig persistent.

INITQNAME('motor.ins.init.queue')

Der Name der Initialisierungswarteschlange, in die der WS-Manager die Auslösenachricht einlegen soll.

TRGENBL(*YES)

Der Auslöserattributwert.

TRGTYPE(*DEPTH)

Es wird ein Auslöserereignis generiert, wenn die Anzahl der Nachrichten mit der erforderlichen Priorität (**TRGMSGPTY**) die in **TRGDEPTH** angegebene Zahl erreicht.

TRGDEPTH(100)

Die Anzahl der Nachrichten, die zum Generieren eines Auslöserereignisses erforderlich sind.

TRGMSGPTY(5)

Die Priorität von Nachrichten, die vom Warteschlangenmanager gezählt werden müssen, um zu entscheiden, ob ein Auslöserereignis generiert werden soll. Es werden nur Nachrichten mit Priorität 5 oder höher gezählt.

Initialisierungswarteschlange definieren

Wenn ein Auslöserereignis eintritt, reißt der Warteschlangenmanager eine Auslösenachricht in die Initialisierungswarteschlange ein, die in der Definition der Anwendungswarteschlange angegeben ist. Initialisie-

Warteschlangen haben keine besonderen Einstellungen, aber Sie können die folgende Definition der lokalen Warteschlange `motor.ins.init.queue` als Anleitung verwenden:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.ins.init.queue') QTYPE(*LCL)
GETENBL(*YES) SHARE(*NO) TRGTYPE(*NONE)
MAXMSGL(2000)
MAXDEPTH(1000)
```

Prozessdefinition erstellen

Verwenden Sie den Befehl **CRTMQMPC**, um eine Prozessdefinition zu erstellen. Eine Prozessdefinition ordnet eine Anwendungswarteschlange der Anwendung zu, die Nachrichten aus der Warteschlange verarbeiten soll. Dies erfolgt über das Attribut `PRCNAME` in der Anwendungswarteschlange `motor.insurance.queue`. Mit dem folgenden Befehl wird der erforderliche Prozess `motor.insurance.quote.process` erstellt, der in diesem Beispiel angegeben ist:

```
CRTMQMPC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')
TEXT('Insurance request message processing')
APPTYPE(*OS400) APPID(MQTEST/TESTPROG)
USRDATA('open, close, 235')
```

Es folgt eine Auflistung der Parameter:

MQMNAME(MYQUEUEMANAGER)

Der Name des Warteschlangenmanagers.

PRCNAME('motor.insurance.quote.process')

Der Name der Prozessdefinition.

TEXT('Insurance request message processing')

Eine Beschreibung des Anwendungsprogramms, auf das sich diese Definition bezieht. Dieser Text wird angezeigt, wenn Sie den Befehl **DSPMQMPC** verwenden. Dadurch können Sie feststellen, was der Prozess tut. Wenn Sie Leerzeichen in der Zeichenfolge verwenden, müssen Sie die Zeichenfolge in einfache Anführungszeichen setzen.

APPTYPE(*OS400)

Der Typ der Anwendung, die gestartet werden soll.

APPID(MQTEST/TESTPROG)

Der Name der ausführbaren Datei der Anwendung, die als vollständig qualifizierter Dateiname angegeben ist.

USRDATA('open, close, 235')

Benutzerdefinierte Daten, die von der Anwendung verwendet werden können.

Prozessdefinition anzeigen

Verwenden Sie den Befehl **DSPMQMPC**, um die Ergebnisse Ihrer Definition zu untersuchen. Beispiel:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPC('motor.insurance.quote.process')
```

Sie können auch den Befehl **CHGMQMPC** verwenden, um eine vorhandene Prozessdefinition zu ändern, und den Befehl **DLTMQMPC**, um eine Prozessdefinition zu löschen.

Kommunikation zwischen zwei IBM MQ-Systemen unter IBM i

Dieses Codebeispiel zeigt, wie zwei IBM MQ for IBM i-Systeme mithilfe von CL-Befehlen so konfiguriert werden, dass sie miteinander kommunizieren können.

Die Systeme werden als `SYSTEMA` und `SYSTEMB` bezeichnet, und das verwendete Übertragungsprotokoll ist TCP/IP.

Gehen Sie wie folgt vor:

1. Erstellen Sie einen Warteschlangenmanager unter SYSTEMA, und rufen Sie ihn QMGRA1 auf.

```
CRTMQM  MQMNAME(QMGRA1) TEXT('System A - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Starten Sie diesen Warteschlangenmanager.

```
STRMQM  MQMNAME(QMGRA1)
```

3. Definieren Sie die IBM MQ-Objekte auf SYSTEMA, die Sie zum Senden von Nachrichten an einen Warteschlangenmanager auf SYSTEMB benötigen.

```
/* Transmission queue */
CRTMQMQ  QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +
MQMNAME(QMGRA1) TEXT('Transmission Queue +
to QMGRB1') MAXDEPTH(5000) USAGE(*TMQ)

/* Remote queue that points to a queue called TARGETB */
/* TARGETB belongs to queue manager QMGRB1 on SYSTEMB */
CRTMQMQ  QNAME(TARGETB.ON.QMGRB1) QTYPE(*RMT) +
MQMNAME(QMGRA1) TEXT('Remote Q pointing +
at Q TARGETB on QMGRB1 on Remote System +
SYSTEMB') RMTQNAME(TARGETB) +
RMTMQMNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)

/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +
MQMNAME(QMGRA1) TRPTYPE(*TCP) +
TEXT('Sender Channel From QMGRA1 on +
SYSTEMA to QMGRB1 on SYSTEMB') +
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. Erstellen Sie einen Warteschlangenmanager unter SYSTEMB, und rufen Sie ihn QMGRB1 auf.

```
CRTMQM  MQMNAME(QMGRB1) TEXT('System B - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

5. Starten Sie den WS-Manager unter SYSTEMB.

```
STRMQM  MQMNAME(QMGRB1)
```

6. Definieren Sie die IBM MQ-Objekte, die Sie zum Empfangen von Nachrichten vom Warteschlangenmanager auf SYSTEMA benötigen.

```
/* Local queue to receive messages on */
CRTMQMQ  QNAME(TARGETB) QTYPE(*LCL) MQMNAME(QMGRB1) +
TEXT('Sample Local Queue for QMGRB1')

/* Receiver channel of the same name as the sender channel on SYSTEMA */
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*RCVR) +
MQMNAME(QMGRB1) TRPTYPE(*TCP) +
TEXT('Receiver Channel from QMGRA1 to +
QMGRB1')
```

7. Starten Sie schließlich einen TCP/IP-Listener auf SYSTEMB, damit der Kanal gestartet werden kann. In diesem Beispiel wird der Standardport 1414 verwendet.

```
STRMQMLSR MQMNAME(QMGRB1)
```

Sie sind jetzt bereit, Testnachrichten zwischen SYSTEMA und SYSTEMB zu senden. Verwenden Sie eines der bereitgestellten Beispiele, um eine Reihe von Nachrichten in Ihre ferne Warteschlange auf SYSTEMA zu stellen.

Starten Sie den Kanal unter SYSTEMA entweder mit dem Befehl **STRMQMCHL** oder verwenden Sie den Befehl **WRKMQMCHL** und geben Sie eine Startanforderung (Option 14) für den Senderkanal ein.

Der Kanal sollte in den Status 'RUNNING' wechseln und die Nachrichten werden an die Warteschlange TARGETB unter SYSTEMB gesendet.

Überprüfen Sie Ihre Nachrichten, indem Sie den folgenden Befehl ausgeben:

```
WRKMQMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

IBM i Beispielressourcendefinitionen unter IBM i

Im Folgenden finden Sie das IBM i-CL-Beispielprogramm AMQSAMP4.

```
/* **** */
/*
/* Program name: AMQSAMP4
/*
/* Description: Sample CL program defining MQM queues
/* to use with the sample programs
/* Can be run, with changes as needed, after
/* starting the MQM
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1993, 2023. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/*
/* **** */
/*
/* Function:
/*
/* AMQSAMP4 is a sample CL program to create or reset the
/* MQI resources to use with the sample programs.
/*
/* This program, or a similar one, can be run when the MQM
/* is started - it creates the objects if missing, or resets
/* their attributes to the prescribed values.
/*
/*
/* Exceptions signaled: none
/* Exceptions monitored: none
/*
/* AMQSAMP4 takes a single parameter, the Queue Manager name
/*
/* **** */
QSYS/PGM PARM(&QMGRNAME)

/* **** */
/* Queue Manager Name Parameter
/* **** */
QSYS/DCL VAR(&QMGRNAME) TYPE(*CHAR)

/* **** */
/* EXAMPLES OF DIFFERENT QUEUE TYPES
/*
/* Create local, alias and remote queues
/*
/* Uses system defaults for most attributes
/*
/* **** */
/* Create a local queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.LOCAL') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Sample local queue') /* description */+
SHARE(*YES) /* Shareable */+
DFTMSGPST(*YES) /* Persistent messages OK */
```

```

/* Create an alias queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ALIAS')      +
MQMNAME(&QMGRNAME)                        +
QTYPE(*ALS) REPLACE(*YES)                 +
+
TEXT('Sample alias queue')                 +
DFTMSGPST(*YES) /* Persistent messages OK */+
TGTONAME('SYSTEM.SAMPLE.LOCAL')

/* Create a remote queue - in this case, an indirect reference */
/* is made to the sample local queue on OTHER queue manager */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REMOTE')      +
MQMNAME(&QMGRNAME)                        +
QTYPE(*RMT) REPLACE(*YES)                 +
+
TEXT('Sample remote queue')/* description */+
DFTMSGPST(*YES) /* Persistent messages OK */+
RMTQNAME('SYSTEM.SAMPLE.LOCAL')           +
RMTMQMNAME(OTHER) /* Queue is on OTHER */

/* Create a transmission queue for messages to queues at OTHER */
/* By default, use remote node name */
CRTMQMQ QNAME('OTHER') /* transmission queue name */+
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                 +
TEXT('Transmission queue to OTHER') +
USAGE(*TMQ) /* transmission queue */

/*****
/* SPECIFIC QUEUES AND PROCESS USED BY SAMPLE PROGRAMS */
/* Create local queues used by sample programs */
/* Create MQI process associated with sample initiation queue */
*****/
/* General reply queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REPLY')      +
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                 +
+
TEXT('General reply queue')                 +
DFTMSGPST(*NO) /* Not Persistent */

/* Queue used by AMQSINQ4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.INQ')        +
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                 +
+
TEXT('Queue for AMQSINQ4')                 +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO) /* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.INQPROCESS')       +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSSET4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.SET')        +
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                 +
+
TEXT('Queue for AMQSSET4')                 +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.SETPROCESS')       +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSECH4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ECHO')      +
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                 +
+
TEXT('Queue for AMQSECH4')                 +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+

```



```

PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Initiation Queue used by AMQSTRG4, sample trigger process */
CRTMQMQ QNAME('SYSTEM.SAMPLE.TRIGGER') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Trigger queue for sample programs')

/* MQI Processes associated with triggered sample programs */
/*
/***** Note - there are versions of the triggered samples *****/
/***** in different languages - set APPID for these *****/
/***** process to the variation you want to trigger *****/
/*
CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSINQ4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSINQ4') /* C */ +
/* APPID('QMOM/AMQOINQ4') /* COBOL */ +
/* APPID('QMOM/AMQ3INQ4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSSET4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSSET4') /* C */ +
/* APPID('QMOM/AMQOSET4') /* COBOL */ +
/* APPID('QMOM/AMQ3SET4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSECH4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSECH4') /* C */ +
/* APPID('QMOM/AMQOECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****
/*
/* Normal return.
/*
/*****
SNDPGMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM

/*****
/*
/* END OF AMQSAMP4
/*
/*****

```

IBM i Alternative Methoden zur Verwaltung von IBM MQ for IBM i

Die Verwendung von CL-Befehlen ist die bevorzugte Methode zur Verwaltung von IBM MQ for IBM i. Sie können jedoch verschiedene andere Verwaltungsmethoden wie MQSC-Befehle, PCF-Befehle und die Fernverwaltung verwenden.

In der Regel verwenden Sie IBM i-CL-Befehle, um IBM MQ for IBM i zu verwalten. Eine Übersicht über diese Befehle finden Sie im Abschnitt „IBM MQ for IBM i mithilfe von CL-Befehlen verwalten“ auf Seite 259.

Sie können den Betrieb von Warteschlangenmanagern mithilfe von IBM MQ-Instrumentierungsereignissen überwachen. Im Abschnitt [Instrumentierungsereignisse](#) finden Sie Informationen zu IBM MQ-Instrumentierungsereignissen und zu ihrer Verwendung.

Sie können jede der in den folgenden Unterabschnitten beschriebenen Verwaltungsmethoden als Alternative zu IBM i-CL-Befehlen verwenden:

IBM i Lokale und ferne Verwaltung unter IBM i

Sie verwalten IBM MQ für IBM i-Objekte lokal oder über Fernzugriff.

Lokale Verwaltung bedeutet, dass Verwaltungstasks auf allen Warteschlangenmanagern ausgeführt werden, die Sie auf Ihrem lokalen System definiert haben. In IBM MQ können Sie dies als lokale Verwaltung in Betracht ziehen, da es keine IBM MQ-Kanäle gibt, d. h., die Kommunikation wird vom Betriebssystem verwaltet. Um diesen Tasktyp ausführen zu können, müssen Sie sich entweder am fernen System anmelden und die Befehle von dort aus ausführen oder einen Prozess erstellen, der die Befehle für Sie absetzen kann.

IBM MQ unterstützt die Verwaltung von einem zentralen Punkt aus, was auch als *Fernverwaltung* bezeichnet wird. Die ferne Verwaltung besteht aus dem Senden von PCF-Steuernachrichten (PCF = Programmable Command Format) an den SYSTEM.ADMIN.COMMAND.QUEUE auf dem Ziel-WS-Manager.

Es gibt eine Reihe von Möglichkeiten, PCF-Nachrichten zu generieren. Diese sind:

1. Schreiben eines Programms mit Hilfe von PCF-Nachrichten. Weitere Informationen finden Sie in [„Verwaltung mithilfe von PCF-Befehlen unter IBM i“](#) auf Seite 276.
2. Schreiben eines Programms mit Hilfe der MQAI, die PCF-Nachrichten sendet. Weitere Informationen finden Sie in [„Verwendung von MQAI zur Vereinfachung der Verwendung von PCFs“](#) auf Seite 35.
3. Verwendung des IBM MQ Explorer, der mit IBM MQ for Windows verfügbar ist und es Ihnen ermöglicht, über eine grafische Benutzerschnittstelle die richtigen PCF-Nachrichten zu generieren. Weitere Informationen finden Sie in [„IBM MQ Explorer mit IBM MQ for IBM i verwenden“](#) auf Seite 276.
4. Verwenden Sie **STRMQMMQSC**, um Befehle indirekt an einen fernen Warteschlangenmanager zu senden. Weitere Informationen finden Sie in [„Verwaltung mithilfe von MQSC-Befehlen unter IBM i“](#) auf Seite 274.

Sie können beispielsweise einen fernen Befehl absetzen, um eine Warteschlangendefinition in einem fernen Warteschlangenmanager zu ändern.

Einige Befehle können auf diese Weise nicht abgesetzt werden, insbesondere die Erstellung oder den Start von Warteschlangenmanagern und den Start von Befehlsservern. Um diesen Tasktyp ausführen zu können, müssen Sie sich entweder am fernen System anmelden und die Befehle von dort absetzen oder einen Prozess erstellen, der die Befehle für Sie absetzen kann.

IBM i Verwaltung mithilfe von MQSC-Befehlen unter IBM i

In diesem Abschnitt finden Sie Informationen zu MQSC-Befehlen und Sie erfahren, wie Sie die Befehle zur Verwaltung von IBM MQ für IBM i verwenden können.

IBM MQ-Scriptbefehle (MQSC) sind in einer lesbaren Form geschrieben (EBCDIC-Text). Sie verwenden MQSC-Befehle zum Verwalten von WS-Manager-Objekten, einschließlich des Warteschlangenmanagers selbst, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services, Themen und Authentifizierungsinformationsobjekte.

MQSC-Befehle werden mit dem CL-Befehl **STRMQMMQSC** von IBM MQ an einen Warteschlangenmanager ausgegeben. Bei dieser Methode handelt es sich nur um eine Stapelmethode, die die Eingabe aus einer physischen Quellendatei im Serverbibliothekssystem übernimmt. Der Standardname für diese physische Quellendatei ist QMQSC.



Achtung: Verwenden Sie die Bibliothek QTEMP nicht als Quellenbibliothek in STRMQMMQSC, da die Verwendung der Bibliothek QTEMP begrenzt ist. Sie müssen eine andere Bibliothek als Eingabedatei für den Befehl verwenden.

IBM MQ for IBM i unterstützt keine Quellendatei mit dem Namen QMQSC. Um MQSC-Befehle zu verarbeiten, müssen Sie die QMQSC -Quellendatei in einer Bibliothek Ihrer Wahl erstellen, indem Sie den folgenden Befehl ausgeben:

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT('IBM MQ - MQSC Source')
```

Die MQSC-Quelle wird in Teildateien in dieser Quellendatei gehalten. Geben Sie den folgenden Befehl ein, um mit den Mitgliedern zu arbeiten:

```
WRKMBRPDM MYLIB/QMQSC
```

Sie können jetzt neue Mitglieder hinzufügen und vorhandene verwalten.

Sie können MQSC-Befehle auch interaktiv eingeben, indem Sie RUNMQSC eingeben oder:

1. Geben Sie den Namen des Warteschlangenmanagers ein und drücken Sie die Enter -Taste, um auf die **WRKMQM** -Ergebnisanzeige zuzugreifen.
2. Wählen Sie F23=More options in dieser Anzeige aus.
3. Wählen Sie Option 26 für einen aktiven Warteschlangenmanager in der in [Abbildung 32 auf Seite 275](#) gezeigten Anzeige aus.

Zum Beenden einer solchen MQSC-Sitzung geben Sie endein.

[Abbildung 32 auf Seite 275](#) ist ein Auszug aus einer WebSphere MQ-Scriptbefehlsdatei, der einen WebSphere MQ-Scriptbefehl (DEFINE QLOCAL) mit seinen Attributen zeigt.

```
.  
.   
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +  
DESCR(' ') +  
PUT(ENABLED) +  
DEFPRTY(0) +  
DEFPSIST(NO) +  
GET(ENABLED) +  
MAXDEPTH(5000) +  
MAXMSGL(1024) +  
DEFSOPT(SHARED) +  
NOHARDENBO +  
USAGE(NORMAL) +  
NOTRIGGER;  
.   
.
```

Abbildung 32. Auszug aus der MQSC-Befehlsdatei myprog.in

Um Portierbarkeit zwischen IBM MQ-Umgebungen zu gewährleisten, müssen Sie die Zeilenlänge in MQSC-Befehlsdateien auf 72 Zeichen begrenzen. Das Pluszeichen gibt an, dass der Befehl in der nächsten Zeile fortgesetzt wird.

Objektattribute, die in MQSC angegeben sind, werden in diesem Abschnitt in Großbuchstaben (z. B. RQMNAME) angezeigt, obwohl die Groß-/Kleinschreibung nicht beachtet werden muss.

Anmerkung:

1. Das Format einer MQSC-Datei hängt nicht von ihrer Position im Dateisystem ab.
2. Die Namen von MQSC-Attributen sind auf acht Zeichen begrenzt.
3. MQSC-Befehle sind auf anderen Plattformen verfügbar, einschließlich z/OS.

Eine Beschreibung der einzelnen MQSC-Befehle und ihrer Syntax finden Sie im Abschnitt [„Scriptbefehle \(MQSC\)“](#) auf Seite 11.

Verwaltung mithilfe von PCF-Befehlen unter IBM i

Der Zweck von IBM MQ-PCF-Befehlen (Programmable Command Format) besteht darin, Verwaltungstasks in einem Verwaltungsprogramm zu programmieren. Auf diese Weise können Sie Warteschlangen und Prozessdefinitionen erstellen und WS-Manager aus einem Programm ändern.

PCF-Befehle decken den gleichen Funktionsumfang ab, der von MQSC-Befehlen bereitgestellt wird. Im Gegensatz zu MQSC-Befehlen befinden sich PCF-Befehle und ihre Antworten jedoch nicht in einem Textformat, das Sie lesen können.

Sie können ein Programm schreiben, um PCF-Befehle an jeden WS-Manager im Netz von einem einzelnen Knoten aus auszugeben. Auf diese Weise können Sie Verwaltungsaufgaben zentralisieren und automatisieren.

Jeder PCF-Befehl ist eine Datenstruktur, die in den Anwendungsdatenteil einer IBM MQ-Nachricht eingebettet ist. Jeder Befehl wird mit der MQPUT-Funktion MQPUT auf die gleiche Weise an den Zielwarteschlangenmanager gesendet, wie jede andere Nachricht. Der Befehlsserver auf dem WS-Manager, der die Nachricht empfängt, interpretiert ihn als Befehlsnachricht und führt den Befehl aus. Um die Antworten abzurufen, gibt die Anwendung einen MQGET-Aufruf aus und die Antwortdaten werden in einer anderen Datenstruktur zurückgegeben. Die Anwendung kann dann die Antwort verarbeiten und entsprechend handeln.

Kurz, dies sind einige der Dinge, die der Anwendungsprogrammierer angeben muss, um eine PCF-Befehlsnachricht zu erstellen:

Nachrichtendeskriptor

Dies ist ein Standardnachrichtendeskriptor von IBM MQ, in dem Folgendes gilt:

- Der Nachrichtentyp (*MsgType*) ist MQMT_REQUEST.
- Nachrichtenformat (*Format*) ist MQFMT_ADMIN.

Anwendungsdaten

Enthält die PCF-Nachricht (einschließlich des PCF-Headers), in der Folgendes gilt:

- Der PCF-Nachrichtentyp (*Type*) gibt MQCFT_COMMAND an.
- Die Befehls-ID gibt den Befehl an, beispielsweise *Change Queue* (MQCMD_CHANGE_Q).

Escape-PCFs sind PCF-Befehle, die MQSC-Befehle innerhalb des Nachrichtentexts enthalten. Sie können PCFs verwenden, um Befehle an einen fernen Warteschlangenmanager zu senden. Weitere Informationen finden Sie im Abschnitt [„Verwendung von MQAI zur Vereinfachung der Verwendung von PCFs“](#) auf Seite 35.

Eine vollständige Beschreibung der PCF-Datenstrukturen und deren Implementierung finden Sie unter [Strukturen für Befehle und Antworten](#).

IBM MQ Explorer mit IBM MQ for IBM i verwenden

Dieser Abschnitt enthält Informationen zur Verwaltung von IBM MQ for IBM i mit dem IBM MQ Explorer.

IBM MQ for Windows (x86 -Plattform) und IBM MQ for Linux (x86 -und x86-64 -Plattformen) stellen eine Verwaltungsschnittstelle namens IBM MQ Explorer bereit, um Verwaltungstasks als Alternative zur Verwendung von CL-, Steuer-oder MQSC-Befehlen auszuführen.

Der IBM MQ Explorer ermöglicht Ihnen die lokale oder ferne Verwaltung Ihres Netzes von einem Computer, auf dem Windows (x86-Plattform) oder Linux (x86- und x86-64-Plattformen) ausgeführt wird, indem Sie auf den IBM MQ Explorer auf den Warteschlangenmanagern und Clustern zeigen, die Sie interessieren.

Mit dem IBM MQ Explorer können Folgendes tun:

- Starten und stoppen Sie einen WS-Manager (nur auf Ihrer lokalen Maschine).
- Definieren, Anzeigen und Ändern der Definitionen von IBM MQ-Objekten, wie z. B. Warteschlangen, Themen und Kanäle.

- Durchsuchen Sie die Nachrichten in einer Warteschlange.
- Starten und stoppen Sie einen Kanal.
- Statusinformationen zu einem Kanal anzeigen.
- Anzeigen von Warteschlangenmanagern in einem Cluster.
- Überprüfen Sie, welche Anwendungen, Benutzer oder Kanäle eine bestimmte Warteschlange geöffnet haben.
- Erstellen Sie mit dem Assistenten **Neuen Cluster erstellen** einen neuen WS-Manager-Cluster.
- Fügen Sie einen Warteschlangenmanager mit dem Assistenten **WS-Manager zum Cluster hinzufügen** zu einem Cluster hinzu.
- Verwalten Sie das Authentifizierungsinformationsobjekt, das mit der TLS-Kanalsicherheit (Transport Layer Security) verwendet wird.

Mit der Online-Anleitung können Sie:

- Definieren und steuern Sie verschiedene Ressourcen, einschließlich Warteschlangenmanager, Warteschlangen, Kanäle, Prozessdefinitionen, Clientverbindungskanäle, Empfangsprogramme, Topics, Services, Namenslisten und Cluster.
- Starten oder Stoppen eines Warteschlangenmanagers und der zugehörigen Prozesse.
- Zeigen Sie die Warteschlangenmanager und die zugehörigen Objekte auf Ihrer Workstation oder von anderen Workstations an.
- Überprüfen Sie den Status von Warteschlangenmanagern, Clustern und Kanälen.

Stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind, bevor Sie versuchen, den IBM MQ Explorer für die Verwaltung von IBM MQ auf einer Servermaschine zu verwenden. Überprüfen Sie Folgendes:

1. Für jeden verwalteten Warteschlangenmanager wird ein Befehlsserver ausgeführt, der auf dem Server mit dem CL-Befehl **STRMQCSVR** gestartet wurde.
2. Für jeden fernen WS-Manager ist ein geeigneter TCP/IP-Listener vorhanden. Dies ist der IBM MQ -Listener, der vom Befehl **STRMQLSR** gestartet wurde.
3. Der Serververbindungskanal mit dem Namen SYSTEM.ADMIN.SVRCONN ist auf jedem fernen WS-Manager vorhanden. Sie müssen diesen Kanal selbst erstellen. Es ist obligatorisch, dass jeder ferne WS-Manager verwaltet wird. Ohne sie ist die Fernverwaltung nicht möglich.
4. Überprüfen Sie, ob die SYSTEM.MQEXPLORER.REPLY.MODEL-Warteschlange vorhanden ist.

IBM i

Befehlsserver für Fernverwaltung unter IBM i verwalten

Dieser Abschnitt enthält Informationen zur Fernverwaltung des IBM MQ for IBM i-Befehlsservers.

Jedem WS-Manager kann ein Befehlsserver zugeordnet sein. Ein Befehlsserver verarbeitet alle ankommenden Befehle von fernen Warteschlangenmanagern oder PCF-Befehlen aus Anwendungen. Er stellt die Befehle für den Warteschlangenmanager zur Verarbeitung dar und gibt abhängig vom Ursprung des Befehls einen Beendigungscode oder eine Bedienernachricht zurück.

Ein Befehlsserver ist für die gesamte Verwaltung mit PCFs, der MQAI und auch für die Fernverwaltung obligatorisch.

Anmerkung: Für die Fernverwaltung müssen Sie sicherstellen, dass der Zielwarteschlangenmanager aktiv ist. Andernfalls können die Nachrichten, die Befehle enthalten, den WS-Manager nicht verlassen, von dem sie ausgegeben werden. Stattdessen werden diese Nachrichten in die lokale Übertragungswarteschlange eingereiht, die dem fernen Warteschlangenmanager dient. Vermeiden Sie diese Situation, wenn dies überhaupt möglich ist.

Es gibt separate Steuerbefehle zum Starten und Stoppen des Befehlsservers. Sie können mit dem IBM MQ Explorer die in den folgenden Abschnitten beschriebenen Operationen ausführen.

Starten und Stoppen des Befehlsservers

Verwenden Sie den folgenden CL-Befehl, um den Befehlsserver zu starten:

```
STRMQCSVR MQMNAME('saturn.queue.manager')
```

Hierbei steht `saturn.queue.manager` für den Warteschlangenmanager, für den der Befehlsserver gestartet wird.

Verwenden Sie einen der folgenden CL-Befehle, um den Befehlsserver zu stoppen:

1.

```
ENDMQCSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRLD)
```

um einen gesteuerten Stopp auszuführen, wobei `saturn.queue.manager` für den Warteschlangenmanager steht, für den der Befehlsserver gestoppt wird. Dies ist die Standardoption. Dies bedeutet, dass der `OPTION(*CNTRLD)` übergangen werden kann.

2.

```
ENDMQCSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)
```

um einen sofortigen Stopp auszuführen, wobei `saturn.queue.manager` für den Warteschlangenmanager steht, für den der Befehlsserver gestoppt wird.

Status des Befehlsservers anzeigen

Stellen Sie für die Fernverwaltung sicher, dass der Befehlsserver auf dem Zielwarteschlangenmanager ausgeführt wird. Wenn der Befehl nicht aktiv ist, können ferne Befehle nicht verarbeitet werden. Alle Nachrichten, die Befehle enthalten, werden in die Befehlswarteschlange `SYSTEM.ADMIN.COMMAND.QUEUE` des Zielwarteschlangenmanagers eingereicht.

Um den Status des Befehlsservers für einen WS-Manager anzuzeigen, der hier `saturn.queue.manager` aufgerufen wird, lautet der CL-Befehl wie folgt:

```
DSPMQCSVR MQMNAME('saturn.queue.manager')
```

Geben Sie diesen Befehl auf der Zielmaschine aus. Wenn der Befehlsserver aktiv ist, wird die in [Abbildung 33](#) auf Seite 278 gezeigte Anzeige aufgerufen:

```
Display MQM Command Server (DSPMQCSVR)
```

```
Queue manager name . . . . . > saturn.queue.manager
```

```
MQM Command Server Status. . . . > RUNNING
```

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

Abbildung 33. Anzeige 'MQM-Befehlsserver anzeigen'

IBM i Ablaufsteuerung für IBM i

In diesem Abschnitt wird beschrieben, wie IBM MQ Arbeitsanforderungen verarbeitet, und es werden die verfügbaren Optionen für die Priorisierung und Steuerung der Jobs, die IBM MQ zugeordnet sind, beschrieben.

Warnung

Ändern Sie die IBM MQ-Arbeitsverwaltungsobjekte nur dann, wenn Sie die Konzepte von IBM i und IBM MQ-Ablaufsteuerung vollständig verstanden haben.

Weitere Informationen zu Subsystemen und Jobbeschreibung finden Sie unter *Work Management* in der IBM i-Produktdokumentation. Achten Sie besonders auf die Abschnitte zu [Startjobs](#) und [Stapeljobs](#).

IBM MQ for IBM i beinhaltet die IBM i UNIX-Umgebung und IBM i-Threads. Nehmen Sie **keine** Änderungen an den Objekten im Integrated File System (IFS) vor.

Im normalen Betrieb startet ein IBM MQ-Warteschlangenmanager eine Reihe von Stapeljobs, um verschiedene Aufgaben auszuführen. Diese Stapeljobs werden standardmäßig im QMQM-Subsystem ausgeführt, das bei der Installation von IBM MQ erstellt wird.

Der Begriff Ablaufsteuerung bezieht sich auf den Prozess der Anpassung von IBM MQ-Jobs, um die optimale Leistung des Systems zu nutzen oder die Verwaltung zu vereinfachen.

Sie können z. B.:

- Ändern Sie die Ausführungspriorität von Jobs, um einen Warteschlangenmanager reaktionsschneller zu machen als ein anderer.
- Die Ausgabe einer Reihe von Jobs an eine bestimmte Ausgabewarteschlange umleiten.
- Stellen Sie sicher, dass alle Jobs eines bestimmten Typs in einem bestimmten Subsystem ausgeführt werden.
- Isolieren Sie Fehler in einem Subsystem.

Die Ablaufsteuerung erfolgt durch die Erstellung oder Änderung der Jobbeschreibungen, die den IBM MQ-Jobs zugeordnet sind. Sie können das Arbeitsmanagement für folgende Aufgaben konfigurieren:

- Eine vollständige IBM MQ-Installation.
- Einzelne Warteschlangenmanager.
- Einzelne Jobs für einzelne WS-Manager.

IBM i IBM MQ-Tasks für IBM i

Die folgende Tabelle enthält eine Liste der IBM MQ for IBM i-Jobs mit einer Kurzbeschreibung jedes einzelnen Jobs.

Wenn ein Warteschlangenmanager aktiv ist, werden einige oder alle der folgenden Stapeljobs angezeigt, die unter dem Benutzerprofil QMQM im IBM MQ-Subsystem ausgeführt werden. Die Jobs werden in [Tabelle 19 auf Seite 279](#) kurz beschrieben.

Sie können alle mit einem Warteschlangenmanager verbundenen Jobs mit Auswahl 22 in der Anzeige **Mit WS-Manager arbeiten** (WRKMQM) anzeigen. Sie können Empfangsprogramme mit dem Befehl WRKMQLSR anzeigen.

Jobname	Funktion
AMQALMPX	Der Prüfpunktprozessor, der Journalprüfpunkte in regelmäßigen Abständen übernimmt.
AMQZMUC0	Utility-Manager. Dieser Job führt kritische WS-Manager-Dienstprogramme aus, z. B. den Journalkettenmanager.
AMQZXMA0	Der Ausführungscontroller, der der erste Job ist, der vom Warteschlangenmanager gestartet wird. Er verarbeitet MQCONN-Anforderungen und startet Agentenprozesse zur Verarbeitung von IBM MQ-API-Aufrufen.
AMQZFUMA	Objektberechtigungsmanager (OAM).

Tabelle 19. IBM MQ-Tasks (Forts.)

Jobname	Funktion
AMQZLAA0	WS-Manager-Agenten, die die meisten Arbeiten für Anwendungen ausführen, die mit MQCNO_STANDARD_BINDING eine Verbindung zum Warteschlangenmanager herstellen.
AMQZLSA0	Warteschlangenmanageragent.
AMQZMUFO	Utility-Manager
AMQZMGRO	Prozesscontroller. Dieser Job wird verwendet, um Empfangsprogramme und Services zu starten und zu verwalten.
AMQZMUR0	Utility-Manager. Dieser Job führt kritische WS-Manager-Dienstprogramme aus, z. B. den Journalkettenmanager.
AMQFQPUB	Publish/Subscribe-Dämon in Warteschlange.
AMQFCXBA	Broker-Worker-Job.
RUNMQBRK	Broker-Steuerjob.
AMQRMPPA	Job für Kanalprozess-Pooling.
AMQCRSTA	TCP/IP-aufgerufene Kanalantworterin.
AMQCRS6B	LU62 -Empfängerkanal und Clientverbindung (siehe Anmerkung).
AMQRRMFA	Repository-Manager für Cluster.
AMQCLMAA	TCP/IP-Listener ohne Thread.
AMQPCSEA	PCF-Befehlsprozessor, der PCF-und Fernverwaltungsanforderungen verarbeitet.
RUNMQTRM	Auslösemonitor.
RUNMQDLQ	Warteschlange für nicht zustellbare Nachrichten.
RUNMQCHI	Der Kanalinitiator.
RUNMQCHL	Senderkanaljob, der für jeden Senderkanal gestartet wird.
RUNMQLSR	Threaded TCP/IP-Listener.
AMQRCMLA	Kanal-MQSC-und PCF-Befehlsprozessor.

Anmerkung: Der LU62 -Empfängerjob wird im DFV-Subsystem ausgeführt und übernimmt die Laufzeiteigenschaften aus den Routing-und DFV-Einträgen, die zum Starten des Jobs verwendet werden. Weitere Informationen finden Sie unter [Initiated end \(Receiver\)](#).

Arbeitsverwaltungsobjekte unter IBM i

Bei der Installation von IBM MQ werden verschiedene Objekte zur Unterstützung der Ablaufsteuerung in der QMQM-Bibliothek bereitgestellt. Dies sind die Objekte, die für die Ausführung von IBM MQ-Jobs in einem eigenen Subsystem erforderlich sind.

Für zwei der IBM MQ-Stapeljobs werden Beispieljobbeschreibungen zur Verfügung gestellt. Wenn für einen IBM MQ-Job keine bestimmte Jobbeschreibung angegeben ist, wird er mit der Standardjobbeschreibung QMQMJOB0 ausgeführt.

Die Arbeitsverwaltungsobjekte, die bei der Installation von IBM MQ bereitgestellt werden, sind in Tabelle 20 auf Seite 281 aufgelistet und die Objekte, die für einen Warteschlangenmanager erstellt wurden, sind in [Tabelle 21 auf Seite 281](#) aufgelistet.

Anmerkung: Die Arbeitsverwaltungsobjekte können in der Bibliothek QMQM gefunden werden, und die WS-Manager-Objekte befinden sich in der WS-Manager-Bibliothek.

Tabelle 20. Arbeitsverwaltungs-objekte

Name	Typ	Beschreibung
AMQALMPX	*JOB	Die Jobbeschreibung, die vom Prüfpunktprozess verwendet wird.
AMQZLAA0	*JOB	Jobbeschreibung, die von den IBM MQ-Agentenprozessen verwendet wird
AMQZLSA0	*JOB	Warteschlangenmanageragent für isolierte Bindungen
AMQZXMA0	*JOB	Jobbeschreibung, die von IBM MQ-Ausführungscontrollern verwendet wird
QMQM	*SBS	Subsystem, in dem alle IBM MQ-Jobs ausgeführt werden.
QMQM	*JOB	Die Jobwarteschlange, die dem angegebenen Subsystem zugeordnet ist.
QMQMJOB	*JOB	Die IBM MQ-Standardjobbeschreibung, die verwendet wird, wenn für einen Job keine bestimmte Jobbeschreibung angegeben ist.
QMQMSG	*MSG	Standardnachrichtenwarteschlange für IBM MQ-Jobs
QMQMRUN20	*CLS	Klassenbeschreibung für IBM MQ-Jobs mit hoher Priorität
QMQMRUN35	*CLS	Klassenbeschreibung für IBM MQ-Jobs mit mittlerer Priorität
QMQMRUN50	*CLS	Klassenbeschreibung für IBM MQ-Jobs mit niedriger Priorität

Tabelle 21. Für einen Warteschlangenmanager erstellte Arbeitsverwaltungsobjekte

Name	Typ	Beschreibung
AMQA000000	*JRNRCV	Lokaler Journalempfänger
AMQAJRN	*JRN	Lokales Journal
AMQJRNINF	*USRSPC	Benutzerbereich, der mit den neuesten Journalempfängern aktualisiert wird, die für den Start und die Datenträgerwiederherstellung eines Warteschlangenmanagers erforderlich sind. Dieser Benutzerbereich kann von einer Anwendung abgefragt werden, um festzustellen, welche Journalempfänger archiviert werden müssen und die sicher gelöscht werden können.
AMQAJRNMSG	*MSG	Nachrichtenwarteschlange des lokalen Journals
AMQCRC6B	*PGM	Programm zum Starten der LU6.2-Verbindung
AMQRFOLD	*DATEI	Migrierte WS-Manager-Kanaldefinitionsdatei
QMQMSG	*MSG	Nachrichtenwarteschlange des Warteschlangenmanagers

IBM i Verwendung von IBM MQ-Arbeitsverwaltungsobjekten unter IBM i

In diesem Abschnitt wird beschrieben, wie IBM MQ Arbeitsverwaltungsobjekte verwendet, und es werden Konfigurationsbeispiele bereitgestellt.



Achtung: Ändern Sie die Einstellungen des Jobwarteschlangeneintrags nicht im QMQM-Subsystem, um die Anzahl der im Subsystem zulässigen Jobs nach Priorität zu begrenzen. Wenn Sie dies versuchen, können wesentliche IBM MQ-Jobs möglicherweise nicht ausgeführt werden, nachdem

sie übergeben wurden; dies kann dazu führen, dass der Start des Warteschlangenmanagers fehlschlägt.

Um zu verstehen, wie die Ablaufsteuerung konfiguriert wird, müssen Sie zunächst verstehen, wie IBM MQ Jobbeschreibungen verwendet.

Die Jobbeschreibung, die zum Starten des Jobs verwendet wird, steuert viele Attribute des Jobs. Beispiel:

- Die Jobwarteschlange, in der sich der Job in der Warteschlange befindet und auf welchem Subsystem der Job ausgeführt wird.
- Die Routing-Daten, die zum Starten des Jobs und der Klasse verwendet werden, die der Job für seine Laufzeitparameter verwendet.
- Die Ausgabewarteschlange, die der Job für Druckdateien verwendet.

Der Prozess zum Starten eines IBM MQ-Jobs kann in drei Schritte unterteilt werden:

1. IBM MQ wählt eine Jobbeschreibung aus.

IBM MQ ermittelt mit folgendem Verfahren, welche Jobbeschreibung für einen Stapeljob verwendet werden muss:

- a. Suchen Sie in der WS-Manager-Bibliothek nach einer Jobbeschreibung mit demselben Namen wie der Job. Weitere Informationen zur Warteschlangenmanagerbibliothek finden Sie unter [Informationen zu IBM MQ for IBM i-Warteschlangenmanagerbibliotheksnamen](#).
- b. Suchen Sie in der WS-Manager-Bibliothek nach der Standardjobbeschreibung QMQMJOB.
- c. Suchen Sie in der QMQM-Bibliothek nach einer Jobbeschreibung mit demselben Namen wie der Job.
- d. Verwenden Sie die Standardjobbeschreibung QMQMJOB in der Bibliothek QMQM.

2. Der Job wird an die Jobwarteschlange übergeben.

Jobbeschreibungen, die mit IBM MQ bereitgestellt werden, sind standardmäßig so konfiguriert, dass Jobs in die Jobwarteschlange QMQM in der Bibliothek QMQM gestellt werden. Die QMQM-Jobwarteschlange ist dem mitgelieferten QMQM-Subsystem zugeordnet, so dass die Jobs standardmäßig im QMQM-Subsystem ausgeführt werden.

3. Der Job wird in das Subsystem eingegeben und durchläuft die Routing-Schritte.

Wenn der Job in das Subsystem eingeht, werden die in der Jobbeschreibung angegebenen Leitwegdaten verwendet, um Leitwegeinträge für den Job zu finden.

Die Routing-Daten müssen mit einem der im QMQM-Subsystem definierten Routing-Einträge übereinstimmen, und dies definiert, welche der angegebenen Klassen (QMQRUN20, QMQRUN35 oder QMQRUN50) vom Job verwendet werden.

Anmerkung: Wenn IBM MQ-Jobs nicht gestartet werden, stellen Sie sicher, dass das Subsystem aktiv ist und die Jobwarteschlange nicht gesperrt ist.

Wenn Sie die IBM MQ-Arbeitsverwaltungsobjekte geändert haben, stellen Sie sicher, dass alle Zuordnungen richtig sind. Wenn Sie z. B. eine andere Jobwarteschlange als QMQM/QMQM in der Jobbeschreibung angeben, stellen Sie sicher, dass ein ADDJOBQE für das Subsystem ausgeführt wird, d.h. QMQM.

Sie können für jeden in [Tabelle 19 auf Seite 279](#) dokumentierten Job eine Jobbeschreibung erstellen. Verwenden Sie dabei folgendes Arbeitsblatt als Beispiel:

```
What is the queue manager library name? _____
Does job description AMQZXMA0 exist in the queue manager library? Yes No
Does job description QMQMJOB exist in the queue manager library? Yes No
Does job description AMQZXMA0 exist in the QMQM library? Yes No
Does job description QMQMJOB exist in the QMQM library? Yes No
```

Wenn Sie Nein zu all diesen Fragen beantworten, erstellen Sie eine globale Jobbeschreibung QMQMJOB in der QMQM-Bibliothek.

IBM MQ-Nachrichtenwarteschlange

Es wird in jeder Warteschlangenmanagerbibliothek eine IBM MQ-Nachrichtenwarteschlange mit dem Namen QMQMSG erstellt. Betriebssystemnachrichten werden an diese Warteschlange gesendet, wenn die Jobs des Warteschlangenmanagers beendet werden und IBM MQ Nachrichten an die Warteschlange sendet. Zum Beispiel, um zu berichten, welche Journalempfänger beim Start benötigt werden. Bewahren Sie die Anzahl der Nachrichten in dieser Nachrichtenwarteschlange in einer überschaubaren Größe auf, um die Überwachung einfacher zu machen.

IBM i Standardsystembeispiele für IBM i

Diese Beispiele zeigen, wie eine nicht geänderte IBM MQ-Installation funktioniert, wenn einige der Standardjobs beim Start des Warteschlangenmanagers übergeben werden.

Zunächst wird der Job AMQZXMA0 des Ausführungscontrollers gestartet.

1. Geben Sie den Befehl **STRMQM** für den Warteschlangenmanager TESTQM aus.
2. IBM MQ durchsucht die Warteschlangenmanagerbibliothek QMTESTQM zunächst nach der Jobbeschreibung AMQZXMA0 und dann nach der Jobbeschreibung QMQMJOB. Da keine dieser Jobbeschreibungen vorhanden ist, sucht IBM MQ in der Produktbibliothek QMQM nach Jobbeschreibung AMQZXMA0. Diese Jobbeschreibung ist vorhanden, so dass sie zum Übergeben des Jobs verwendet wird.
3. In der Jobbeschreibung wird die standardmäßige IBM MQ-Jobwarteschlange verwendet, d. h., der Job wird an die Jobwarteschlange QMQM/QMQM übergeben.
4. Die Routing-Daten in der Jobbeschreibung AMQZXMA0 sind QMQMRUN20, so dass das System die Routing-Einträge des Subsystems nach einem, das mit diesen Daten übereinstimmt, durchsucht. Standardmäßig verfügt der Leitweeintrag mit der Folgenummer 9900 über Vergleichsdaten, die mit QMQMRUN20 übereinstimmen, so dass der Job mit der Klasse gestartet wird, die in diesem Leitweeintrag definiert ist, der auch als QMQMRUN20 bezeichnet wird.
5. Die Klasse QMQM/QMQMRUN20 hat Priorität auf 20 gesetzt, so dass der AMQZXMA0-Job im Subsystem QMQM mit derselben Priorität wie die meisten interaktiven Jobs auf dem System ausgeführt wird.

Als Nächstes wird der AMQALMPX-Prüfpunkt-Prozess-Job gestartet.

1. IBM MQ durchsucht die Warteschlangenmanagerbibliothek QMTESTQM zunächst nach der Jobbeschreibung AMQALPMX und dann nach der Jobbeschreibung QMQMJOB. Da keine dieser Jobbeschreibungen vorhanden ist, sucht IBM MQ in der Produktbibliothek QMQM nach den Jobbeschreibungen AMQALMPX und QMQMJOB. Jobbeschreibung AMQALMPX ist nicht vorhanden, aber QMQMJOB wird ausgeführt, daher wird QMQMJOB zum Übergeben des Jobs verwendet.
Anmerkung: Die Jobbeschreibung QMQMJOB wird immer für IBM MQ-Jobs verwendet, für die keine eigene Jobbeschreibung vorhanden ist.
2. In der Jobbeschreibung wird die standardmäßige IBM MQ-Jobwarteschlange verwendet, d. h., der Job wird an die Jobwarteschlange QMQM/QMQM übergeben.
3. Die Leitwegdaten in der Jobbeschreibung QMQMJOB sind QMQMRUN35, so dass das System die Routing-Einträge des Subsystems nach einem, das mit diesen Daten übereinstimmt, durchsucht. Der Leitweeintrag mit der Folgenummer 9910 weist standardmäßig Vergleichsdaten auf, die mit QMQMRUN35 übereinstimmen, so dass der Job mit der Klasse gestartet wird, die in diesem Leitweeintrag definiert ist, der auch als QMQMRUN35 bezeichnet wird.
4. Die Klasse QMQM/QMQMRUN35 hat Priorität auf 35 gesetzt, so dass der AMQALMPX-Job im Subsystem QMQM mit einer niedrigeren Priorität als die meisten interaktiven Jobs auf dem System ausgeführt wird, aber höhere Priorität als die meisten Stapeljobs hat.

In diesem Abschnitt erfahren Sie, wie Sie IBM MQ-Jobbeschreibungen ändern und erstellen können, um die Laufzeitattribute von IBM MQ-Jobs zu ändern.

Der Schlüssel für die Flexibilität der IBM MQ-Ablaufsteuerung liegt in dem zweistufigen Verfahren, mit dem IBM MQ nach Jobbeschreibungen sucht:

- Wenn Sie Jobbeschreibungen in einer WS-Manager-Bibliothek erstellen oder ändern, überschreiben diese Änderungen die globalen Jobbeschreibungen in QMQM, aber die Änderungen sind lokal und betreffen nur den jeweiligen Warteschlangenmanager.
- Wenn Sie globale Jobbeschreibungen in der Bibliothek QMQM erstellen oder ändern, wirken sich diese Jobbeschreibungen auf alle Warteschlangenmanager im System aus, sofern sie nicht lokal für einzelne WS-Manager außer Kraft gesetzt werden.

1. Im folgenden Beispiel wird die Priorität von Kanalsteuerungsjobs für einen einzelnen WS-Manager erhöht.

Um den Repository-Manager und die Kanalinitiatorjobs, AMQRRMFA und RUNMQCHI, so schnell wie möglich für Warteschlangenmanager TESTQM auszuführen, führen Sie die folgenden Schritte aus:

- a. Erstellen Sie in der Warteschlangenmanagerbibliothek lokale Duplikate der Jobbeschreibung QMQM/QMQMJOBDD mit den Namen der IBM MQ-Prozesse, die gesteuert werden sollen. Beispiel:

```
CRTDUPOBJ OBJ(QMQMJOBDD) FROMLIB(QMQM) OBJTYPE(*JOBDD) TOLIB(QMTESTQM)
NEWOBJ (RUNMQCHI)
CRTDUPOBJ OBJ(QMQMJOBDD) FROMLIB(QMQM) OBJTYPE(*JOBDD) TOLIB(QMTESTQM)
NEWOBJ (AMQRRMFA)
```

- b. Ändern Sie den Parameter für die Routing-Daten in der Jobbeschreibung, um sicherzustellen, dass die Jobs die Klasse QMQMRUN20 verwenden.

```
CHGJOBDD JOBDD(QMTESTQM/RUNMQCHI) RTGDTA('QMQMRUN20')
CHGJOBDD JOBDD(QMTESTQM/AMQRRMFA) RTGDTA('QMQMRUN20')
```

Die Jobs 'AMQRRMFA' und 'RUNMQCHI' für Warteschlangenmanager TESTQM jetzt:

- Die neuen lokalen Jobbeschreibungen in der WS-Manager-Bibliothek verwenden
- Führen Sie mit Priorität 20 aus, da die Klasse QMQMRUN20 verwendet wird, wenn die Jobs das Subsystem betreten.

2. Im folgenden Beispiel wird eine neue Ausführungsprioritätsklasse für das Subsystem QMQM definiert.

- a. Erstellen Sie eine doppelte Klasse in der Bibliothek QMQM, um anderen Warteschlangenmanagern den Zugriff auf die Klasse zu ermöglichen, indem Sie den folgenden Befehl ausgeben:

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ (QMQMRUN10)
```

- b. Ändern Sie die Klasse so, dass sie die neue Ausführungspriorität hat, indem Sie den folgenden Befehl ausgeben:

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. Fügen Sie die neue Klassendefinition dem Subsystem hinzu, indem Sie den folgenden Befehl ausgeben:

```
ADDRTGE SBSDD(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)
CLS(QMQM/QMQMRUN10)
```

Anmerkung: Sie können einen beliebigen numerischen Wert für die Leitwegfolgennummer angeben, aber die Werte müssen in einer sequenziellen Reihenfolge angegeben werden. Diese Folgennummer

teilt dem Subsystem die Reihenfolge mit, in der Leitwegeinträge nach einer Leitwegdatenübereinstimmung durchsucht werden sollen.

- d. Ändern Sie die lokale oder globale Jobbeschreibung so, dass sie die neue Prioritätsklasse verwendet, indem Sie den folgenden Befehl ausgeben:

```
CHGJOB JOB(QMQMLibname/QMQMJOB) RTGDTA('QMQRUN10')
```

Nun verwenden alle WS-Manager-Jobs, die dem Warteschlangenmanager QMLibraryname zugeordnet sind, eine Ausführungspriorität von 10.

3. Im folgenden Beispiel wird ein Warteschlangenmanager in einem eigenen Subsystem ausgeführt.

Führen Sie die folgenden Schritte aus, um alle Jobs für den WS-Manager TESTQM im Subsystem QBATCH auszuführen:

- a. Erstellen Sie ein lokales Duplikat der Jobbeschreibung QMQM/QMQMJOB in der WS-Manager-Bibliothek mit dem Befehl.

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
```

- b. Ändern Sie den Jobwarteschlangenparameter in der Jobbeschreibung, um sicherzustellen, dass die Jobs die Jobwarteschlange QBATCH verwenden.

```
CHGJOB JOB(QMTESTQM/QMQMJOB) JOBQ(*LIBL/QBATCH)
```

Anmerkung: Die Jobwarteschlange ist der Subsystembeschreibung zugeordnet. Wenn Sie feststellen, dass sich die Jobs in der Jobwarteschlange befinden, überprüfen Sie, ob die Jobwarteschlangendefinition auf dem SBS definiert ist. Den Befehl DSPSBS für das Subsystem verwenden und die Option 6 (Jobwarteschlangeneinträge) verwenden.

Alle Jobs für WS-Manager TESTQM jetzt:

- Verwenden Sie die neue lokale Standardjobbeschreibung in der WS-Manager-Bibliothek.
- Werden an die Jobwarteschlange QBATCH übergeben.

Gehen Sie wie folgt vor, um sicherzustellen, dass Jobs ordnungsgemäß weitergeleitet und priorisiert werden

- Erstellen Sie Routing-Einträge für die IBM MQ-Jobs im Subsystem QBATCH oder
- Verlassen Sie sich auf einen Catch-All-Routing-Eintrag, der QCMD aufruft, unabhängig davon, welche Routing-Daten verwendet werden.

Diese Option kann nur verwendet werden, wenn die Option für die maximale Anzahl aktiver Jobs für die Jobwarteschlange QBATCH auf *NOMAX gesetzt ist. Der Systemstandardwert ist 1.

4. Im folgenden Beispiel wird ein anderes IBM MQ-Subsystem erstellt.

- a. Erstellen Sie ein doppeltes Subsystem in der QMQM-Bibliothek, indem Sie den folgenden Befehl ausgeben:

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBS) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. Entfernen Sie die QMQM-Jobwarteschlange, indem Sie den folgenden Befehl ausgeben:

```
RMVJOBQE SBS(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. Erstellen Sie eine neue Jobwarteschlange für das Subsystem, indem Sie den folgenden Befehl ausgeben:

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. Fügen Sie dem Subsystem einen Jobwarteschlangeneintrag hinzu, indem Sie den folgenden Befehl ausgeben:

```
ADDJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. Erstellen Sie ein Duplikat QMQMJOB in der WS-Manager-Bibliothek, indem Sie den folgenden Befehl ausgeben:

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMlibraryname)
```

- f. Ändern Sie die Jobbeschreibung so, dass sie die neue Jobwarteschlange verwendet, indem Sie den folgenden Befehl ausgeben:

```
CHGJOB JOB(QMlibraryname/QMQMJOB) JOBQ(QMQM/QMQM2)
```

- g. Starten Sie das Subsystem, indem Sie den folgenden Befehl ausgeben:

```
STRSBS SBSDB(QMQM/QMQM2)
```

Anmerkung:

- a. Sie können das Subsystem in einer beliebigen Bibliothek angeben. Wenn das Produkt aus irgendeinem Grund erneut installiert wird oder die QMQM-Bibliothek ersetzt wird, werden alle Änderungen entfernt, die Sie vorgenommen haben.
 - b. Alle WS-Manager-Jobs, die dem Warteschlangenmanager QMlibraryname zugeordnet sind, werden jetzt unter dem Subsystem QMQM2 ausgeführt.
5. Das folgende Beispiel erfasst die gesamte Ausgabe für einen Jobtyp.

Führen Sie die folgenden Schritte aus, um alle Prüfpunktprozesse AMQALMPX, Jobprotokolle für mehrere WS-Manager in einer einzigen Ausgabewarteschlange zu erfassen:

- a. Erstellen Sie eine Ausgabewarteschlange, z. B.

```
CRTOUTQ OUTQ(MYLIB/CHKPTLOGS)
```

- b. Erstellen Sie ein globales Duplikat der Jobbeschreibung QMQM/QMQMJOB mit dem Namen des IBM MQ-Prozesses, der gesteuert werden soll, z. B. wie folgt:

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) NEWOBJ(AMQALMPX)
```

- c. Ändern Sie den Parameter für die Ausgabewarteschlange in der Jobbeschreibung so, dass er auf die neue Ausgabewarteschlange verweist, und ändern Sie die Jobprotokollierungsstufe so, dass alle Nachrichten in das Jobprotokoll geschrieben werden.

```
CHGJOB JOB(QMQM/AMQALMPX) OUTQ(MYLIB/CHKPTLOGS) LOG(4 00 *SECLVL)
```

Alle AMQALMPX-Jobs von IBM MQ für alle Warteschlangenmanager verwenden die neue globale Jobbeschreibung AMQALMPX, sofern in der lokalen Warteschlangenmanagerbibliothek keine Überschreibung durch lokale Jobbeschreibungen erfolgt.

Alle Spool-Dateien des Jobprotokolls für diese Jobs werden jetzt in die Ausgabewarteschlange CHKPTLOGS in der Bibliothek MYLIB geschrieben.

Anmerkung:

- a. Das vorherige Beispiel funktioniert nur, wenn der Parameter QPJOBLOG oder eine beliebige Druckdatei den Wert *JOB für den Ausgabewarteschlangenparameter hat. Im obigen Beispiel benötigt die Datei QSYS/QPDJOBLOG die Angabe OUTQ in *JOB.

- b. Um eine Systemdruckdatei zu ändern, den Befehl CHGPRTF (Druckdatei ändern) verwenden. Beispiel:

```
CHGPRTF PRTF(QJOBLOG) OUTQ(*JOB)
```

Die Option *JOB gibt an, dass die Jobbeschreibungen verwendet werden müssen.

- c. Sie können alle Spooldateien, die den IBM MQ-Jobs zugeordnet sind, an eine bestimmte Ausgabe-warteschlange senden. Stellen Sie jedoch sicher, dass die verwendete Druckdatei den entsprechenden Wert für den Parameter OUTQ hat.

IBM i Verfügbarkeit, Sicherung, Wiederherstellung und Neustart unter IBM i

Verwenden Sie diese Informationen, um zu verstehen, wie IBM MQ for IBM i die Journalunterstützung für IBM i verwendet, um die Sicherungs- und Zurückschreibungsstrategie zu unterstützen.

Lesen Sie diesen Abschnitt erst, wenn Sie mit den standardmäßigen Sicherungs- und Wiederherstellungsverfahren in IBM i sowie mit der Verwendung von Journalen und den zugehörigen Journalempfängern in IBM i vertraut sind. Weitere Informationen zu diesen Themen finden Sie unter [Sicherung und Wiederherstellung](#).

Um die Sicherungs- und Wiederherstellungsstrategie zu verstehen, müssen Sie zunächst verstehen, wie IBM MQ for IBM i die zugehörigen Daten im Dateisystem von IBM i und im Integrated File System (IFS) organisiert.

IBM MQ for IBM i speichert seine Daten in einer einzelnen Bibliothek für jede Warteschlangenmanagerinstanz und in Datenstromdateien im IFS-Dateisystem.

Die WS-Manager-spezifischen Bibliotheken enthalten Journale, Journalempfänger und Objekte, die für die Steuerung der Arbeitsverwaltung des Warteschlangenmanagers erforderlich sind. Die IFS-Verzeichnisse und -Dateien enthalten IBM MQ-Konfigurationsdateien, die Beschreibungen von IBM MQ-Objekten und die darin enthaltenen Daten.

Jede Änderung an diesen Objekten, die über einen Systemfehler hinweg wiederherstellbar ist, wird in einem Journal aufgezeichnet, *bevor* es auf das entsprechende Objekt angewendet wird. Dies hat zur Folge, dass solche Änderungen durch die Wiedergabe der im Journal aufgezeichneten Informationen wiederhergestellt werden können.

Sie können IBM MQ for IBM i für die Verwendung mehrerer Warteschlangenmanagerinstanzen auf verschiedenen Servern konfigurieren, um eine höhere Verfügbarkeit des Warteschlangenmanagers zu gewährleisten und im Falle eines Server- oder Warteschlangenmanagerausfalls die Wiederherstellung zu beschleunigen.

IBM i Warteschlangenmanagerjournale unter IBM i

In diesem Abschnitt wird beschrieben, wie IBM MQ for IBM i während seines Betriebs Journale zur Steuerung von Aktualisierungen lokaler Objekte verwendet.

Jede WS-Manager-Bibliothek enthält ein Journal für diesen Warteschlangenmanager, und das Journal hat den Namen QM *GRLIB*/AMQ *A* JRN. Dabei ist QM *GRLIB* der Name der WS-Manager-Bibliothek, und *A* ein Buchstabe, *A* im Fall eines einzigen Instanz-WS-Managers, der für die Warteschlangenmanagerinstanz eindeutig ist.

QM *GRLIB* verwendet den Namen QM, gefolgt von dem Namen des Warteschlangenmanagers in einer eindeutigen Form. Beispiel: Ein WS-Manager mit dem Namen TEST hat eine WS-Manager-Bibliothek mit dem Namen QMTEST. Die WS-Manager-Bibliothek kann bei der Erstellung eines Warteschlangenmanagers mit dem Befehl **CRTMQM** angegeben werden.

Journalen sind Journalempfänger zugeordnet, die die Informationen enthalten, die im Journal aufgezeichnet werden. Die Empfänger sind Objekte, an die Informationen nur angehängt werden können, und füllen schließlich die Daten aus.

Journalempfänger verwenden einen wertvollen Plattenspeicherplatz mit Daten, die nicht mehr aktuell sind. Sie können die Informationen jedoch in den permanenten Speicher stellen, um dieses Problem zu minimieren. Ein Journalempfänger wird zu einem bestimmten Zeitpunkt an das Journal angehängt. Wenn der Journalempfänger seine vorgegebene Schwellenwertgröße erreicht, wird er freigegeben und durch einen neuen Journalempfänger ersetzt. Sie können den Schwellenwert für Journalempfänger angeben, wenn Sie einen Warteschlangenmanager mit dem Parameter **CRTMQM** und dem Parameter **THRESHOLD** erstellen.

Die zum lokalen Journal von IBM MQ für IBM i gehörigen Journalempfänger sind in jeder Warteschlangenmanagerbibliothek vorhanden. Für Sie gilt folgende Namenskonvention:

```
AMQ Arnnnnn
```

Dabei gilt:

A

ist ein Buchstabe A-Z. Es ist A für einzelne Instanz-WS-Manager. Sie variiert je nach Instanzen eines Multi-Instanz-Warteschlangenmanagers.

nnnnn

ist die Dezimalzahl 00000 to 99999, die um 1 für das nächste Journal in der Folge erhöht wird.

r

ist dezimal 0 to 9, das um 1 erhöht wird, wenn ein Empfänger zurückgespeichert wird.

Die Reihenfolge der Journale basiert auf dem Datum. Die Benennung des nächsten Journals basiert jedoch auf den folgenden Regeln:

1. AMQA \pm nnnnn wechselt zu AMQA \pm (nnnnn+1), und nnnnn wird umgebrochen, wenn es 99999 erreicht. AMQA099999 geht z. B. zu AMQA000000 und AMQA999999 geht zu AMQA900000.
2. Wenn bereits ein Journal mit einem von Regel 1 generierten Namen vorhanden ist, wird die Nachricht CPI70E3 an die Nachrichtenwarteschlange von QSYSOPR gesendet und die automatische Empfängerumschaltung wird gestoppt.

Der derzeit angehängbare Empfänger wird weiterhin verwendet, bis Sie das Problem untersuchen und einen neuen Empfänger manuell anhängen.

3. Wenn in der Sequenz kein neuer Name verfügbar ist (d. a. sind alle möglichen Journalnamen auf dem System vorhanden), müssen Sie beide der folgenden Schritte ausführen:
 - a. Löschen nicht mehr benötigter Journale (siehe Abschnitt „[Journalverwaltung unter IBM i](#)“ auf Seite 293).
 - b. Notieren Sie die Journaländerungen in den letzten Journalempfänger mit (**RCDMQMIMG**) und wiederholen Sie den vorherigen Schritt. Auf diese Weise können die alten Journalempfängernamen wiederverwendet werden.

Das Journal AMQAJRN verwendet die Option MNGRCV(*SYSTEM), um das Betriebssystem zu aktivieren, damit Journalempfänger automatisch geändert werden können, wenn der Schwellenwert erreicht wird. Weitere Informationen darüber, wie das System Empfänger verwaltet, finden Sie in *IBM i Backup and Recovery*.

Der Standardschwellenwert des Journalempfängers ist 100.000 KB. Sie können diesen Wert bei der Erstellung des Warteschlangenmanagers auf einen größeren Wert setzen. Der Anfangswert des Attributs LogReceiverSize wird in die Stanza LogDefaults der mq.s.ini-Datei geschrieben.

Wenn ein Journalempfänger sich über den angegebenen Schwellenwert hinaus erstreckt, wird der Empfänger freigegeben und ein neuer Journalempfänger erstellt, wobei Attribute vom vorherigen Empfänger übernommen werden. Änderungen an den Attributen LogReceiverSize oder LogASP nach der Erstellung eines Warteschlangenmanagers werden ignoriert, wenn das System automatisch einen neuen Journalempfänger anordnet.

Weitere Informationen zur Konfiguration des Systems finden Sie im Abschnitt [Konfigurationsinformationen in IBM i ändern](#).

Wenn Sie die Größe von Journalempfängern ändern müssen, nachdem der Warteschlangenmanager erstellt wurde, erstellen Sie einen neuen Journalempfänger, und legen Sie seinen Eigner mithilfe der folgenden Befehle auf QMQM fest:

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +  
TEXT('MQM LOCAL JOURNAL RECEIVER')  
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

Dabei gilt:

QMGRLIB

Ist der Name der WS-Manager-Bibliothek

A

Ist die Instanzkennung (in der Regel A).

rnnnnn

Ist der nächste Journalempfänger in der zuvor beschriebenen Namensfolge

xxxxxx

Ist der neue Schwellenwert für Empfänger (in KB)

Anmerkung: Die maximale Größe des Empfängers richtet sich nach dem Betriebssystem. Um diesen Wert zu überprüfen, prüfen Sie das Schlüsselwort THRESHOLD im Befehl **CRTJRNRCV** .

Schließen Sie den neuen Empfänger jetzt mit dem Befehl an das AMQAJRN -Journal an:

```
CHGJRN JRN(QMGRLIB/AMQ A JRN) JRNRCV(QMGRLIB/AMQ Annnnnn)
```

Ausführlichere Informationen zur Verwaltung dieser Journalempfänger finden Sie im Abschnitt „[Journalverwaltung unter IBM i](#)“ auf Seite 293.

IBM i Verwendung von Warteschlangenmanagerjournalen unter IBM i

In diesem Abschnitt wird beschrieben, wie IBM MQ for IBM i während seines Betriebs Journale zur Steuerung von Aktualisierungen lokaler Objekte verwendet.

Persistente Aktualisierungen von Nachrichtenwarteschlangen werden in zwei Phasen durchgeführt. Die Datensätze, die die Aktualisierung darstellen, werden zuerst in das Journal geschrieben, und anschließend wird die Warteschlangendatei aktualisiert.

Die Journalempfänger können daher mehr auf dem neuesten Stand als die Warteschlangendateien sein. Mithilfe von Prüfpunkten wird in IBM MQ sichergestellt, dass die Neustartverarbeitung an einem konsistenten Punkt beginnt.

Ein Prüfpunkt ist ein Zeitpunkt, an dem der in dem Journal beschriebene Datensatz mit dem Datensatz in der Warteschlange identisch ist. Der Prüfpunkt selbst besteht aus der Reihe der Journalsätze, die zum erneuten Starten des Warteschlangenmanagers erforderlich sind. Der Status aller Transaktionen (d.h. Arbeitseinheiten), die zum Zeitpunkt des Prüfpunkts aktiv sind, ist beispielsweise der Status aller Transaktionen.

Prüfpunkte werden automatisch von IBM MQ generiert. Sie werden beim Starten und Herunterfahren des Warteschlangenmanagers und nach der Protokollierung einer bestimmten Anzahl von Operationen ausgeführt.

Sie können einen Warteschlangenmanager zwingen, einen Prüfpunkt zu übernehmen, indem Sie den Befehl RCDMQMIMG für alle Objekte in einem Warteschlangenmanager ausgeben und die Ergebnisse wie folgt anzeigen:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

Wenn die Warteschlangen weitere Nachrichten verarbeiten, wird der Prüfpunktsatz mit dem aktuellen Status der Warteschlangen nicht konsistent.

Beim Neustart sucht IBM MQ nach dem letzten Prüfpunktsatz im Protokoll. Diese Informationen werden in der Prüfpunktdatei gehalten, die am Ende jedes Prüfpunkts aktualisiert wird. Der Prüfpunktsatz stellt den letzten Konsistenzpunkt zwischen dem Protokoll und den Daten dar. Die Daten von diesem Prüfpunkt werden verwendet, um die Warteschlangen so wiederherzustellen, wie sie an der Prüfpunktzeit vorhanden waren. Wenn die Warteschlangen erneut erstellt werden, wird das Protokoll dann vorgespielt, um die Warteschlangen wieder in den Zustand zu bringen, in dem sie sich vor Systemausfall oder in der Nähe befanden.

Die Verwendung des Journals in IBM MQ wird anhand des Beispiels einer lokalen Warteschlange namens TESTQ in Warteschlangenmanager TEST deutlich. Dies wird durch die IFS-Datei dargestellt:

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

Die Aktionen, die stattfinden, wenn eine bestimmte Nachricht in diese Warteschlange eingereicht und anschließend aus der Warteschlange abgerufen wird, sind in [Abbildung 34 auf Seite 290](#) zu sehen.

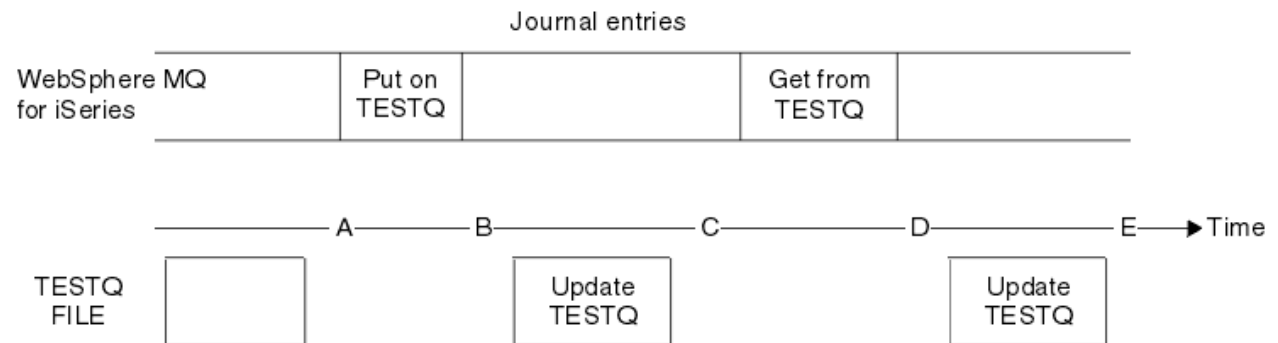


Abbildung 34. Reihenfolge von Ereignissen beim Aktualisieren von MQM-Objekten

Die fünf Punkte A bis E, die im Diagramm dargestellt sind, stellen Punkte in der Zeit dar, die die folgenden Status definieren:

- A** Die IFS-Dateidarstellung der Warteschlange ist mit den im Journal enthaltenen Informationen konsistent.
- B** Ein Journaleintrag wird in das Journal geschrieben, in dem eine Put -Operation in der Warteschlange definiert wird.
- C** Die entsprechende Aktualisierung wird in die Warteschlange gestellt.
- D** Ein Journaleintrag wird in das Journal geschrieben, in dem eine Get -Operation aus der Warteschlange definiert wird.
- E** Die entsprechende Aktualisierung wird in die Warteschlange gestellt.

Der Schlüssel für die Wiederherstellungsfähigkeiten von IBM MQ for IBM i ist, dass der Benutzer die IFS-Dateidarstellung von TESTQ zum Zeitpunkt A speichern und anschließend die IFS-Dateidarstellung von TESTQ zum Zeitpunkt E wiederherstellen kann, indem das gesicherte Objekt wiederhergestellt wird und die Einträge im Journal ab dem Zeitpunkt A wiederholt werden.

Diese Strategie wird von IBM MQ for IBM i verwendet, um persistente Nachrichten nach einem Systemausfall wiederherzustellen. IBM MQ merkt sich einen bestimmten Eintrag in den Journalempfängern und stellt sicher, dass beim Start die Einträge in den Journalen ab diesem Zeitpunkt wiederholt werden. Dieser Starteintrag wird in regelmäßigen Abständen neu berechnet, sodass IBM MQ beim nächsten Start nur die mindestens notwendigen Wiederholungen durchführen muss.

IBM MQ ermöglicht eine individuelle Wiederherstellung von Objekten. Alle persistenten Informationen, die sich auf ein Objekt beziehen, werden in den lokalen IBM MQ for IBM i-Journalen aufgezeichnet. Jedes

IBM MQ-Objekt, das beschädigt oder manipuliert wird, kann aus den Informationen im Journal vollständig wiederhergestellt werden.

Weitere Informationen dazu, wie das System Empfänger verwaltet, finden Sie im Abschnitt [„Verfügbarkeit, Sicherung, Wiederherstellung und Neustart unter IBM i“](#) auf Seite 287.

IBM i Medienimages unter IBM i

Unter IBM i ist ein Medienimage eine vollständige Kopie eines IBM MQ-Objekts, die im Journal aufgezeichnet wird. Einige beschädigte oder beschädigte Objekte können automatisch aus ihrem Medienimage wiederhergestellt werden.

Ein IBM MQ-Objekt von langer Dauer kann eine große Anzahl von Journaleinträgen darstellen, bis zurück zu dem Punkt, an dem es erstellt wurde. Um dies zu vermeiden, gibt es in IBM MQ for IBM i das Konzept eines Medienimages eines Objekts.

Dieses Medienimage ist eine vollständige Kopie des IBM MQ-Objekts, die im Journal aufgezeichnet wird. Wenn ein Image eines Objekts erstellt wird, kann das Objekt wiederhergestellt werden, indem die Journaleinträge ab diesem Image erneut wiedergegeben werden. Der Eintrag im Journal, der den Wiedergabepunkt für jedes IBM MQ-Objekt darstellt, wird als Medienwiederherstellungseintrag bezeichnet. IBM MQ protokolliert Folgendes:

- Medienwiederherstellungseintrag für jedes WS-Manager-Objekt.
- Ältester Eintrag aus dieser Gruppe (Details siehe Fehlernachricht AMQ7462 in [„Journalverwaltung unter IBM i“](#) auf Seite 293)

Die Images des *CTLG-Objekts und des *MQM-Objekts werden regelmäßig ausgeführt, da diese Objekte für den Neustart des Warteschlangenmanagers von entscheidender Bedeutung sind.

Bilder von anderen Objekten werden aufgenommen, wenn sie bequem sind. Standardmäßig werden Images aller Objekte erstellt, wenn ein WS-Manager mit dem Befehl **ENDMQM** und dem Parameter ENDCCTJOB (*YES) beendet wird. Diese Operation kann für sehr große Warteschlangenmanager sehr viel Zeit in Anspruch nehmen. Wenn Sie die Systemabschaltung schnell beenden müssen, geben Sie den Parameter RCDMQMIMG (*NO) mit ENDCCTJOB (*YES) an. In solchen Fällen wird empfohlen, ein vollständiges Datenträgerimage in den Journalen aufzuzeichnen, nachdem der WS-Manager erneut gestartet wurde. Verwenden Sie dazu den folgenden Befehl:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ zeichnet automatisch ein Image eines Objekts auf, wenn es einen geeigneten Punkt findet, an dem ein Objekt durch einen kurzen Eintrag im Journal kompakt beschrieben werden kann. Dies kann jedoch bei einigen Objekten, z. B. Warteschlangen, die eine große Anzahl von Nachrichten enthalten, nie passieren.

Es ist nicht ratsam, das Datum des ältesten Medienwiederherstellungseintrags unnötig lange beizubehalten. Verwenden Sie stattdessen den IBM MQ-Befehl RCDMQMIMG, mit dem Sie von ausgewählten Objekten manuell ein Image erstellen können.

Wiederherstellung von Medienimages

IBM MQ stellt einige Objekte automatisch aus ihrem Medienimage wieder her, wenn festgestellt wird, dass sie manipuliert wurden oder beschädigt sind. Dies gilt insbesondere für die speziellen *MQM- und *CTLG-Objekte als Teil des normalen WS-Manager-Starts. Wenn eine Synchronisationspunkttransaktion zum Zeitpunkt der letzten Beendigung des Warteschlangenmanagers unvollständig war, wird jede betroffene Warteschlange ebenfalls automatisch wiederhergestellt, um die Startoperation abzuschließen.

Sie müssen andere Objekte manuell mit dem IBM MQ -Befehl RCRMQMOBJ wiederherstellen. Mit diesem Befehl werden die Einträge im Journal wiederholt, um das IBM MQ-Objekt erneut zu erstellen. Wenn ein IBM MQ-Objekt beschädigt wird, bestehen die einzigen gültigen Aktionen darin, es zu löschen oder mithilfe dieser Methode erneut zu erstellen. Beachten Sie jedoch, dass nicht persistente Nachrichten auf diese Weise nicht wiederhergestellt werden können.

IBM i **Prüfpunkte unter IBM MQ for IBM i**

Prüfpunkte werden zu verschiedenen Zeitpunkten ausgeführt, um einen bekannten konsistenten Startpunkt für die Wiederherstellung bereitzustellen.

Der Prüfpunktprozess AMQALMPX ist für die Übernahme des Prüfpunkts an den folgenden Punkten verantwortlich:

- Start des Warteschlangenmanagers (STRMQM).
- Beendigung des Warteschlangenmanagers (ENDMQM).
- Nach Ablauf eines Zeitraums seit dem letzten Prüfpunkt (der Standardzeitraum ist 30 Minuten) und eine Mindestanzahl von Protokollsätzen seit dem vorherigen Prüfpunkt geschrieben wurde (der Standardwert ist 100).
- Nachdem eine Reihe von Protokollsätzen geschrieben wurde. Der Standardwert ist 10 000.
- Nachdem die Journalschwellengröße überschritten wurde und ein neuer Journalempfänger automatisch erstellt wurde.
- Wenn ein vollständiges Medienimage aufgenommen wird mit:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

IBM i **Sicherungen von IBM MQ for IBM i-Daten**

In diesem Abschnitt werden die zwei Typen der IBM MQ-Sicherung für jeden einzelnen Warteschlangenmanager beschrieben.

Für jeden Warteschlangenmanager kommen zwei Typen der IBM MQ-Sicherung in Betracht:

- Daten- und Journalsicherung.

Um sicherzustellen, dass beide Datengruppen konsistent sind, müssen Sie dies nur nach dem Herunterfahren des Warteschlangenmanagers tun.

- Journalsicherung.

Sie können diese Schritte ausführen, während der Warteschlangenmanager aktiv ist.

Für beide Methoden müssen Sie die Namen des IFS-Verzeichnisses des Warteschlangenmanagers und der WS-Manager-Bibliothek suchen. Sie finden diese Informationen in der IBM MQ-Konfigurationsdatei (mq5.ini). Weitere Informationen finden Sie in der [Zeilengruppe 'QueueManager'](#).

Führen Sie die folgenden Prozeduren aus, um beide Sicherungstypen zu erstellen:

Daten- und Journalsicherung eines bestimmten Warteschlangenmanagers

Anmerkung: Verwenden Sie bei der Ausführung des Warteschlangenmanagers keine Sicherung im aktiven Betrieb. Eine solche Anforderung kann erst dann ausgeführt werden, wenn alle COMMIT-Definitionen mit anstehenden Änderungen festgeschrieben oder rückgängig gemacht wurden. Wenn dieser Befehl verwendet wird, wenn der Warteschlangenmanager aktiv ist, werden die Kanalverbindungen möglicherweise nicht normal beendet. Verwenden Sie immer die folgende Prozedur.

1. Erstellen Sie einen leeren Journalempfänger mit dem folgenden Befehl:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Zeichnen Sie mit dem Befehl **RCDMQMIMG** ein MQM-Image für alle IBM MQ-Objekte auf und erzwingen Sie dann mit folgendem Befehl einen Prüfpunkt:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Beenden Sie die Kanäle, und stellen Sie sicher, dass der Warteschlangenmanager nicht aktiv ist. Wenn der WS-Manager aktiv ist, stoppen Sie ihn mit dem Befehl **ENDMQM**.

4. Sichern Sie die WS-Manager-Bibliothek, indem Sie den folgenden Befehl ausgeben:

```
SAVLIB LIB(QMTEST)
```

5. Sichern Sie die IFS-Verzeichnisse des Warteschlangenmanagers, indem Sie den folgenden Befehl ausgeben:

```
SAV DEV(...) OBJ((' /QIBM/UserData/mqm/qmgrs/test'))
```

Journalisierung eines bestimmten Warteschlangenmanagers

Da alle relevanten Informationen in den Journalen gespeichert sind, können Teilsicherungen durch Sichern der Journalempfänger ausgeführt werden, solange Sie eine vollständige Sicherung zu einem bestimmten Zeitpunkt ausführen. Diese Datensätze zeichnen alle Änderungen seit dem Zeitpunkt der Gesamtsicherung ab und werden durch Absetzen der folgenden Befehle ausgeführt:

1. Erstellen Sie einen leeren Journalempfänger mit dem folgenden Befehl:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Zeichnen Sie mit dem Befehl **RCDMQMIMG** ein MQM-Image für alle IBM MQ-Objekte auf und erzwingen Sie dann mit folgendem Befehl einen Prüfpunkt:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Sichern Sie die Journalempfänger mit dem folgenden Befehl:

```
SAVOBJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

Eine einfache Sicherungsstrategie besteht darin, jede Woche eine Gesamtsicherung der IBM MQ-Bibliotheken sowie eine tägliche Journalisierung durchzuführen. Dies hängt natürlich davon ab, wie Sie Ihre Sicherungsstrategie für Ihr Unternehmen eingerichtet haben.

Journalverwaltung unter IBM i

Im Rahmen Ihrer Backup-Strategie kümmern Sie sich um Ihre Journalempfänger. Es kann aus folgenden Gründen sinnvoll sein, Journalempfänger aus den IBM MQ-Bibliotheken zu entfernen:

- Um Speicherbereich freizugeben, gilt dies für alle Journalempfänger.
- Gehen Sie wie folgt vor, um die Leistung beim Starten zu verbessern (STRMQM)
- Gehen Sie wie folgt vor, um die Leistung beim erneuten Erstellen von Objekten zu verbessern (RCRMQMOBJ)

Bevor Sie einen Journalempfänger löschen, müssen Sie darauf achten, dass Sie eine Sicherungskopie haben und dass der Journalempfänger nicht mehr benötigt wird.

Jornalempfänger können aus der WS-Manager-Bibliothek *nach* entfernt werden, nachdem sie von den Journalen abgehängt und gespeichert wurden, sofern sie für die Wiederherstellung verfügbar sind, wenn sie für eine Wiederherstellungsoperation benötigt werden.

Das Konzept der Journalverwaltung ist in [Abbildung 35 auf Seite 294](#) dargestellt.

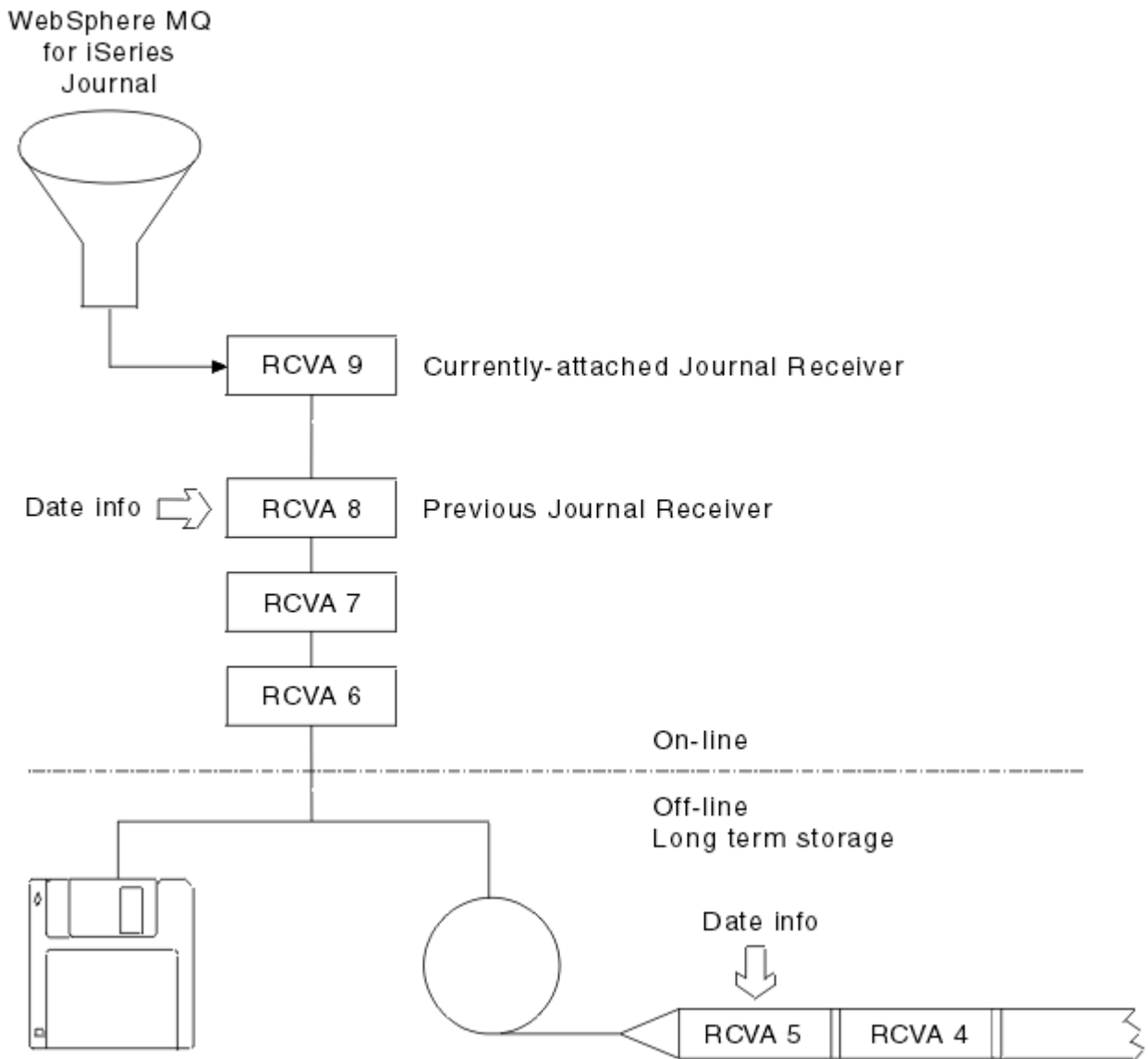


Abbildung 35. Journalführung in IBM i

Es ist wichtig zu wissen, wie weit IBM MQ wahrscheinlich in den Journalen zurück gehen muss, um zu bestimmen, wann ein Journalempfänger, der gesichert wurde, aus der Warteschlangenmanagerbibliothek entfernt werden kann und wann die Sicherung selbst gelöscht werden kann.

IBM MQ gibt zwei Nachrichten an die Nachrichtenwarteschlange des Warteschlangenmanagers (QMOMSG in der Warteschlangenmanagerbibliothek) aus, die dabei helfen, diese Zeit zu bestimmen. Diese Nachrichten werden ausgegeben, wenn sie gestartet werden, wenn sie einen lokalen Journalempfänger ändert und Sie RCDMQIMG verwenden, um einen Prüfpunkt zu erzwingen. Die beiden Nachrichten lauten wie folgt:

AMQ7460

Startwiederherstellungspunkt. Diese Nachricht definiert Datum und Uhrzeit des Starteintrags, ab dem IBM MQ das Journal im Falle eines Startwiederherstellungsvorgangs wiederholt. Wenn der Journalempfänger, der diesen Datensatz enthält, in den IBM MQ-Bibliotheken verfügbar ist, enthält diese Nachricht auch den Namen des Journalempfängers, der den Datensatz enthält.

AMQ7462

Ältester Datenträger-Wiederherstellungseintrag. Diese Nachricht definiert das Datum und die Uhrzeit des ältesten Eintrags, der verwendet werden soll, um ein Objekt aus seinem Datenträgerimage neu zu erstellen.

Der angegebene Journalempfänger ist der älteste erforderliche Journalempfänger. Alle anderen IBM MQ-Journalempfänger mit einem älteren Erstellungsdatum werden nicht mehr benötigt. Wenn nur Sterne angezeigt werden, müssen Sie Sicherungen ab dem angegebenen Datum wiederherstellen, um zu bestimmen, welcher der älteste Journalempfänger ist.

Wenn diese Nachrichten protokolliert werden, schreibt IBM MQ auch ein Benutzeradressbereichsobjekt in die Warteschlangenmanagerbibliothek, das nur einen Eintrag enthält: den Namen des ältesten Journalempfängers, der auf dem System gespeichert werden muss. Dieser Benutzerbereich wird als AMQJR-NINF bezeichnet, und die Daten werden in das folgende Format geschrieben:

```
JJJJJJJJJJLLLLLLLLLLLLYYYYMMDDHHMMSSmmm
```

Dabei gilt:

JJJJJJJJJJ

Ist der älteste Empfängername, den IBM MQ noch benötigt.

LLLLLLLLLL

Ist der Name des Journalempfängerbibliotheksnamens.

YYYY

Ist das Jahr des ältesten Journaleintrags, den IBM MQ benötigt.

MM

Ist der Monat des ältesten Journaleintrags, den IBM MQ benötigt.

DD

Ist der Tag des ältesten Journaleintrags, den IBM MQ benötigt.

HH

Ist die Stunde des ältesten Journaleintrags, den IBM MQ benötigt.

SS

Ist die Sekunde des ältesten Journaleintrags, den IBM MQ benötigt.

mmm

Ist die Millisekunde des ältesten Journaleintrags, den IBM MQ benötigt.

Wenn der älteste Journalempfänger aus dem System gelöscht wurde, enthält dieser Benutzerbereich Sterne (*) für den Journalempfängernamen.

Anmerkung: Die regelmäßige Ausführung von RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) kann die Startzeit für IBM MQ einsparen und die Anzahl der lokalen Journalempfänger reduzieren, die Sie für die Sicherung und Wiederherstellung benötigen.

IBM MQ for IBM i bezieht sich nicht auf die Journalempfänger, es sei denn, es führt einen Wiederherstellungsvorgang für einen Systemstart oder zum erneuten Erstellen eines Objekts aus. Wenn festgestellt wird, dass ein Journal, das erforderlich ist, nicht vorhanden ist, wird die Nachricht AMQ7432 in die Nachrichtenwarteschlange des Warteschlangenmanagers (QMOMMSG) ausgegeben, in der die Zeit und das Datum des Journaleintrags, der für die Ausführung des Wiederherstellungsdurchgangs erforderlich ist, gemeldet werden.

Wenn dies der Fall ist, stellen Sie alle Journalempfänger zurück, die nach diesem Datum von der Sicherung abgehängt wurden, um den Wiederanlauf erfolgreich zu ermöglichen.

Bewahren Sie den Journalempfänger, der den Starteintrag enthält, und alle nachfolgenden Journalempfänger auf, die in der WS-Manager-Bibliothek verfügbar sind.

Bewahren Sie den Journalempfänger mit dem ältesten Media Recovery Entry und allen nachfolgenden Journalempfängern auf, die zu allen Zeiten verfügbar sind und entweder in der WS-Manager-Bibliothek oder in der Sicherungsdatei vorhanden sind.

Wenn Sie einen Prüfpunkt erzwingen:

- Wenn der in AMQ7460 angegebene Journalempfänger nicht fortgeschritten ist, weist dies darauf hin, dass eine unvollständige Arbeitseinheit vorhanden ist, die festgeschrieben oder zurückgesetzt werden muss.

- Wenn der in AMQ7462 angegebene Journalempfänger nicht hoch ist, weist dies darauf hin, dass ein oder mehrere beschädigte Objekte vorhanden sind.

IBM i **Vollständigen Warteschlangenmanager (Daten und Journale) unter IBM i wiederherstellen**

Verwenden Sie diese Informationen, um einen oder mehrere Warteschlangenmanager von einer Sicherung oder von einer fernen Maschine zurückzuschreiben.

Gehen Sie wie folgt vor, um einen oder mehrere IBM MQ-Warteschlangenmanager aus einer Sicherung wiederherzustellen:

1. Versetzen Sie die IBM MQ-Warteschlangenmanager in den Quiescemodus
2. Suchen Sie die neueste Sicherungsgruppe, die aus Ihrer letzten vollständigen Sicherung besteht, und sichern Sie anschließend die Journalempfänger.
3. Führen Sie eine RSTLIB-Operation aus der Gesamtsicherung durch, um die IBM MQ-Datenbibliotheken in ihrem Zustand zum Zeitpunkt der vollständigen Sicherung wiederherzustellen, indem Sie folgende Befehle ausgeben:

```
RSTLIB LIB(QMQLIB1) .....
RSTLIB LIB(QMQLIB2) .....
```

Wurde ein Journalempfänger teilweise in einer Journalsicherung gesichert und in einem nachfolgenden Backup vollständig gesichert, wird nur der vollständig gesicherte gesichert. Journale einzeln in chronologischer Reihenfolge wiederherstellen.

4. Führen Sie mit folgendem Befehl eine RST-Operation durch, um die IFS-Verzeichnisse von IBM MQ im IFS-Dateisystem wiederherzustellen:

```
RST DEV(...) OBJ((' /QIBM/UserData/mqm/qmgrs/testqm')) ...
```

5. Starten Sie den Nachrichten-WS-Manager. Dies wiederholt alle Journalsätze, die seit der vollständigen Sicherung geschrieben wurden, und stellt alle IBM MQ-Objekte in dem konsistenten Zustand zum Zeitpunkt der Journalsicherung wieder her.

Wenn Sie einen vollständigen Warteschlangenmanager auf einer anderen Maschine zurückschreiben wollen, verwenden Sie die folgende Prozedur, um alle Daten aus der WS-Manager-Bibliothek wiederherzustellen. (Wir verwenden TEST als Beispiel-WS-Manager-Name.)

1. CRTMQM TEST
2. DLTLIB LIB(QMTEST)
3. RSTLIB SAVLIB(QMTEST) DEV(*SAVF) SAVF(QMGRLIBSAV)
4. Löschen Sie die folgenden IFS-Dateien:

```
/QIBM/UserData/mqm/qmgrs/TEST/QMQMCHKPT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMQMOBJCAT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMANAGER
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.AUTH.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CHANNEL.INITQ/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.COMMAND.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.REPOSITORY.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.TRANSMIT.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.PENDING.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.ADMIN.COMMAND.QUEUE/q
```

5. STRMQM TEST
6. RCRMQMOBJ OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(TEST)

Journalempfänger für einen bestimmten Warteschlangenmanager unter IBM i wiederherstellen

Verwenden Sie diese Informationen, um die verschiedenen Methoden zum Zurückspeichern von Journalempfängern zu verstehen.

Die häufigste Maßnahme besteht darin, einen Sicherungsjournalempfänger in eine WS-Manager-Bibliothek zurückzuspeichern, wenn ein Empfänger, der entfernt wurde, wieder für eine nachfolgende Wiederherstellungsfunktion benötigt wird.

Dies ist eine einfache Aufgabe, die es erfordert, dass die Journalempfänger mit dem IBM i-Standardbefehl RSTOBJ wiederhergestellt werden:

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNRCV) .....
```

Eine Reihe von Journalempfängern muss möglicherweise zurückgeschrieben werden, und nicht nur ein einzelner Empfänger. Beispielsweise ist AMQA000007 der älteste Empfänger in den IBM MQ -Bibliotheken und sowohl AMQA000005 als auch AMQA000006 müssen zurückgeschrieben werden.

In diesem Fall die Empfänger einzeln in umgekehrter chronologischer Reihenfolge zurückschreiben. Dies ist nicht immer notwendig, aber gute Praxis. In schwerwiegenden Situationen müssen Sie möglicherweise den IBM i-Befehl WRKJRNA verwenden, um die wiederhergestellten Journalempfänger dem Journal zuzuordnen.

Bei der Zurückspeicherung von Journalen erstellt das System automatisch einen angehängten Journalempfänger mit einem neuen Namen in der Journalempfängersequenz. Der generierte neue Name kann jedoch mit einem Journalempfänger identisch sein, den Sie zurückschreiben müssen. Es ist ein manueller Eingriff erforderlich, um dieses Problem zu beheben. Um einen neuen Namen Journalempfänger in Folge zu erstellen, und das neue Journal vor dem Zurückschreiben des Journalempfängers erstellen.

Sie können beispielsweise das Problem mit dem gespeicherten Journal AMQAJRN und den folgenden Journalempfängern berücksichtigen:

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

Wenn das Journal AMQAJRN in eine WS-Manager-Bibliothek zurückgespeichert wird, erstellt das System automatisch den Journalempfänger AMQA000000. Dieser automatisch generierte Empfänger steht in Konflikt mit einem der vorhandenen Journalempfänger (AMQA000000), den Sie zurückschreiben wollen, den Sie nicht zurückschreiben können.

Die Lösung ist:

1. Der nächste Journalempfänger wird manuell erstellt (siehe Abschnitt „Warteschlangenmanagerjournale unter IBM i“ auf Seite 287):

```
CRTJRNRCV JRNRCV(QMQRLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Erstellen Sie das Journal manuell mit dem Journalempfänger:

```
CRTJRN JRN(QMGLIB/AMQAJRN) MNGRCV(*SYSTEM) +
JRNRVC(QMGLIB/AMQA9000001) MSGQ(QMGLIB/AMQAJRNMSG)
```

3. Stellen Sie die lokalen Journalempfänger AMQA000000 in AMQA900000 wieder her.

IBM i Mehrinstanz-Warteschlangenmanager unter IBM i erstellen

Warteschlangenmanager mit mehreren Instanzen verbessern die Verfügbarkeit, indem sie automatisch zu einem Standby-Server wechseln, wenn der aktive Server ausfällt. Der aktive Server und der Standby-Server sind mehrere Instanzen desselben Warteschlangenmanagers; sie verwenden dieselben WS-Manager-Daten. Wenn die aktive Instanz fehlschlägt, müssen Sie ihr Journal in die Bereitschaftsdatenbank übertragen, die die Warteschlange übernimmt, damit der WS-Manager seine Warteschlangen erneut erstellen kann.

Konfigurieren Sie die IBM i-Systeme, auf denen Sie Multi-Instanz-Warteschlangenmanager ausführen, sodass, wenn die aktive Warteschlangenmanagerinstanz ausfällt, das von ihr verwendete Journal für die Standby-Instanz, die die Ausführung übernimmt, verfügbar ist. Sie können eigene Konfigurations- und Verwaltungstasks entwerfen, um das Journal aus der aktiven Instanz für die Instanz verfügbar zu machen, die die Instanz übernimmt. Wenn die Nachrichten nicht verloren gehen sollen, muss Ihr Entwurf sicherstellen, dass das Bereitschaftsjournal mit dem aktiven Journal am Fehlerpunkt konsistent ist. Sie können Ihr Design von einer der beiden Konfigurationen anpassen, die mit Beispielen in nachfolgenden Abschnitten beschrieben werden, die die Konsistenz wahren.

1. Spiegeln Sie das Journal aus dem System, auf dem die aktive WS-Manager-Instanz ausgeführt wird, auf die Systeme, auf denen Standby-Instanzen ausgeführt werden.
2. Das Journal in einen unabhängigen Zusatzspeicherpool (IASP) stellen, der von dem System, auf dem die aktive Instanz ausgeführt wird, auf eine Standby-Instanz übertragen werden kann.

Die erste Lösung erfordert keine zusätzliche Hardware oder Software, da sie Basis-ASPs verwendet. Die zweite Lösung erfordert umschaltbare IASPs, die eine IBM i-Clustering-Unterstützung benötigen, die als gegen Aufpreis erhältliche Lizenz (IBM i License Product 5761-SS1 Option 41) verfügbar ist.

IBM i Zuverlässigkeit und Verfügbarkeit unter IBM i

WS-Manager mit mehreren Instanzen zielen darauf ab, die Verfügbarkeit von Anwendungen zu verbessern. Mit technologischen und physikalischen Bedingungen benötigen Sie verschiedene Lösungen, um die Anforderungen an die Wiederherstellung nach einem Katastrophenfall, die Unterstützung von Warteschlangenmanagern und den fortlaufenden Betrieb zu erfüllen.

Bei der Konfiguration von Zuverlässigkeit und Verfügbarkeit ist der Handel mit einer Vielzahl von Faktoren, die sich in vier verschiedenen Konstruktionspunkten ergeben, zu ersehen:

Wiederherstellung nach einem Katastrophenfall

Optimiert für die Wiederherstellung nach einer größeren Katastrophe, die alle Ihre lokalen Ressourcen zerstört.

Das Disaster-Recovery unter IBM i basiert häufig auf der geographischen Spiegelung von IASP.

Sicherung

Optimiert für die Wiederherstellung nach einem lokalisierten Fehler, häufig einem menschlichen Fehler oder einem unvorhergesehenem technischen Problem.

IBM MQ stellt Backup-Warteschlangenmanager bereit, um Warteschlangenmanager in regelmäßigen Abständen zu sichern. Sie können auch die asynchrone Replikation von WS-Manager-Journalen verwenden, um die Aktualität der Sicherung zu verbessern.

Verfügbarkeit

Optimiert für die Wiederherstellung von Operationen, die das Auftreten eines nahezu unterbrechungsfreien Service nach vorhersehbaren technischen Ausfällen, wie z. B. einem Server oder einem Plattenausfall, schnell ermöglichen.

Die Wiederherstellung wird in der Regel in Minuten gemessen, wobei die Erkennung manchmal länger ist als der Wiederherstellungsprozess. Ein Multi-Instanz-Warteschlangenmanager unterstützt Sie bei der Konfiguration von *Verfügbarkeit*.

Kontinuierlicher Betrieb

Optimiert für die Bereitstellung eines ununterbrochenen Service.

Kontinuierliche Operationslösungen müssen das Erkennungsproblem lösen und fast immer mit der Übergabe derselben Arbeit über mehrere Systeme verbunden sein und entweder mit dem ersten Ergebnis arbeiten oder wenn die Korrektheit eine wichtige Überlegung ist, die mindestens zwei Ergebnisse vergleicht.

Ein Multi-Instanz-Warteschlangenmanager unterstützt Sie bei der Konfiguration von *Verfügbarkeit*. Es ist jeweils eine Instanz des Warteschlangenmanagers aktiv. Das Umschalten auf eine Standby-Instanz dauert von etwas mehr als zehn Sekunden bis zu einem Viertelminuten oder mehr, je nachdem, wie das System konfiguriert, geladen und optimiert wird.

Ein Multi-Instanz-Warteschlangenmanager kann das Auftreten eines nahezu unterbrechungsfreien Service bei der Verwendung mit wiederanschliessbaren IBM MQ MQI clients angeben, die in der Lage sind, die Verarbeitung fortzusetzen, ohne dass das Anwendungsprogramm zwangsläufig einen Ausfall des Warteschlangenmanagers kennt. Informationen hierzu finden Sie im Artikel [Automatische Clientneueverbindung](#).



Komponenten einer Hochverfügbarkeitslösung unter IBM i

Konstruieren Sie eine Hochverfügbarkeitslösung unter Verwendung von Warteschlangenmanagern mit mehreren Instanzen, indem Sie einen stabilen Netzspeicher für WS-Manager-Daten, die Journalreplikation oder den stabilen IASP-Speicher für WS-Manager-Journale bereitstellen und wiederverbindbare Clients verwenden, von Anwendungen, die als wiederanlauffähige WS-Manager-Services konfiguriert sind.

Ein Warteschlangenmanager mit mehreren Instanzen reagiert auf die Erkennung des Warteschlangenmanagerfehlers, indem er den Start einer anderen WS-Manager-Instanz auf einem anderen Server wiederaufnimmt. Um den Start abzuschließen, benötigt die Instanz Zugriff auf die Daten des gemeinsam genutzten Warteschlangenmanagers im Netzspeicher und auf die zugehörige Kopie des lokalen WS-Manager-Journals.

Um eine Hochverfügbarkeitslösung zu erstellen, müssen Sie die Verfügbarkeit der WS-Manager-Daten, die Verwaltung des lokalen WS-Manager-Journals verwalten und entweder wiederverbindbare Clientanwendungen erstellen oder Ihre Anwendungen als WS-Manager-Services implementieren, damit sie automatisch erneut gestartet werden, wenn der Warteschlangenmanager wieder aufgenommen wird. Eine automatische Wiederherstellung einer Client-Verbindung wird von IBM MQ classes for Java nicht unterstützt.

WS-Manager-Daten

Platzieren Sie WS-Manager-Daten in den vernetzten Speicher, der gemeinsam genutzt wird, hoch verfügbar und zuverlässig ist, möglicherweise unter Verwendung von Platten der RAID-Stufe 1 oder höher. Das Dateisystem muss die Voraussetzungen für ein gemeinsam genutztes Dateisystem für Warteschlangenmanager mit mehreren Instanzen erfüllen. Weitere Informationen zu den Voraussetzungen für gemeinsam genutzte Dateisysteme finden Sie im Abschnitt [Anforderungen für gemeinsam genutzte Dateisysteme](#). Network File System Version 4 (NFS4) ist ein Protokoll, das diese Anforderungen erfüllt.

WS-Manager-Journ

Außerdem müssen Sie die IBM i-Journale konfigurieren, die von den Warteschlangenmanagerinstanzen verwendet werden, sodass die Standby-Instanz in der Lage ist, deren Warteschlangenmanagerdaten in einem konsistenten Zustand wiederherzustellen. Bei einem ununterbrochenen Service bedeutet dies, dass Sie die Journale in ihren Status zurückschreiben müssen, wenn die aktive Instanz fehlgeschlagen ist. Anders als bei Backup- oder Disaster Recovery-Lösungen reicht die Wiederherstellung von Journalen bis zu einem früheren Prüfpunkt nicht aus.

Eine physische gemeinsame Nutzung von Journalen durch mehrere IBM i-Systeme ist in einem gemeinsam genutzten Netzspeicher nicht möglich. Zum Zurückschreiben von WS-Manager-Journalen in den kon-

sistenten Status am Fehlerpunkt müssen Sie entweder das physische Journal, das lokal ist, an die aktive WS-Manager-Instanz übertragen, wenn die neue Instanz, die aktiviert wurde, fehlgeschlagen ist, oder ein Spiegelungsspiegel des Journals zum Ausführen von Standby-Instanzen verwendet wird. Das gespiegelte Journal ist ein fernes Journal-Replikat, das exakt mit dem lokalen Journal, das zur fehlgeschlagenen Instanz gehört, synchronisiert wurde.

Drei Konfigurationen sind Ausgangspunkt für das Entwerfen der Verwaltung der Journale für einen Multi-Instanz-Warteschlangenmanager.

1. Verwenden Sie die synchronisierte Journalreplikation (Journal spiegeln) aus dem ASP der aktiven Instanz in die ASPs der Standby-Instanzen.
2. Sie übertragen einen IASP, den Sie für die Aufnahme des Warteschlangenmanagerjournals von der aktiven Instanz in die Standby-Instanz konfiguriert haben, die als aktive Instanz übernommen wird.
3. Synchronisierte sekundäre IASP-Spiegel verwenden.

Weitere Informationen zum Einreihen von Warteschlangenmanagerdaten in einen iASP finden Sie unter [ASP-Optionen im IBM MQ IBM i-Befehl CRTMQM](#).

Siehe auch [Hochverfügbarkeit](#) in IBM Documentation.

Anwendungen

Um einen Client zu erstellen, der die Verbindung zum Warteschlangenmanager automatisch wiederherstellt, wenn der Standby-Warteschlangenmanager wiederaufgenommen wird, verbinden Sie Ihre Anwendung mit dem Warteschlangenmanager mit MQCONNX und geben Sie MQCNO_RECONNECT_Q_MGR im Feld **MQCNO** Optionen an. Weitere Informationen zum Entwerfen von Clientanwendungen für die Wiederherstellung finden Sie im Abschnitt [Hochverfügbarkeitsprogrammprogramme für drei Beispielprogramme, die wiederverbindbare Clients verwenden, und Anwendungswiederherstellung](#).

 *Gemeinsame Nutzung über ein Netz für Warteschlangenmanagerdaten mithilfe von NetServer unter IBM i einrichten*

Richten Sie eine gemeinsame Nutzung über das Netz auf einem IBM i-Server zum Speichern von Warteschlangenmanagerdaten ein. Richten Sie Verbindungen von zwei Servern aus, die zum Zugriff auf die Netzfregabe in den Host-WS-Manager-Instanzen gehen.

Vorbereitende Schritte

- Für diese Aufgabe sind drei IBM i-Server erforderlich. Die Netzwirkfreigabe wird auf einem der Server, GAMMA, definiert. Die anderen beiden Server, ALPHA und BETA, sollen eine Verbindung zu GAMMA herstellen.
- Installieren Sie IBM MQ auf allen drei Servern.
- Installieren Sie den System i Navigator (siehe [System i Navigator](#)).

Informationen zu diesem Vorgang

- Erstellen Sie das WS-Manager-Verzeichnis auf GAMMA, und legen Sie das korrekte Eigentumsrecht und die Berechtigungen für die Benutzerprofile QMQM und QMQMADM fest. Das Verzeichnis und die Berechtigung lassen sich ohne großen Aufwand durch die Installation von IBM MQ auf GAMMA erstellen.
- Erstellen Sie mit System i Navigator eine Freigabe für das Warteschlangenmanagerdatenverzeichnis auf GAMMA.
- Erstellen Sie Verzeichnisse auf ALPHA und BETA, die auf die Freigabe verweisen.

Vorgehensweise

1. Erstellen Sie auf GAMMA das Verzeichnis, in dem die WS-Manager-Daten als Eigner des QMQM -Benutzerprofils und QMQMADM als Primärgruppe enthalten sind.

Tipp:

Eine schnelle und zuverlässige Möglichkeit, das Verzeichnis mit den richtigen Berechtigungen zu erstellen, besteht darin, IBM MQ auf GAMMA zu installieren.

Wenn IBM MQ später nicht mehr auf GAMMA ausgeführt werden soll, deinstallieren Sie IBM MQ. Nach der Deinstallation bleibt das Verzeichnis /QIBM/UserData/mqm/qmgrs auf GAMMA mit dem Benutzerprofil des Eigners QMQM und QMQMADM der Primärgruppe erhalten.

Die Task verwendet das /QIBM/UserData/mqm/qmgrs -Verzeichnis auf GAMMA für die Freigabe.

2. Starten Sie den System i Navigator-Assistenten **Verbindung hinzufügen** und stellen Sie eine Verbindung zum GAMMA-System her.
 - a) Doppelklicken Sie auf das Symbol **System i Navigator** auf Ihrem Windows-Desktop.
 - b) Klicken Sie auf **Ja** , um eine Verbindung zu erstellen.
 - c) Folgen Sie den Anweisungen im Assistenten **Verbindung hinzufügen** und erstellen Sie eine Verbindung vom IBM i-System zu GAMMA.

Die Verbindung zu GAMMA wird zu **Meine Verbindungen** hinzugefügt.

3. Fügen Sie eine neue Dateifreigabe auf GAMMA hinzu.
 - a) Klicken Sie im **System i Navigator** -Fenster auf den Ordner File Shares in My Connections/GAMMA/File Systems.
 - b) Klicken Sie im Fenster **My Tasks** auf **Manage IBM i NetServer shares**.

Ein neues Fenster, **IBM i NetServer-GAMMA** , wird auf Ihrem Desktop geöffnet und zeigt gemeinsam genutzte Objekte an.
 - c) Klicken Sie mit der rechten Maustaste Shared Objects Ordner> **Datei > Neu > Datei**.

Es wird ein neues Fenster, **IBM i NetServer File Share-GAMMA** , geöffnet.
 - d) Geben Sie der Freigabe einen Namen, z. B. WMQ .
 - e) Legen Sie die Zugriffssteuerung auf Lesen/Schreiben fest.
 - f) Wählen Sie den **Pfadnamen** aus, indem Sie zu dem zuvor erstellten /QIBM/UserData/mqm/qmgrs -Verzeichnis navigieren, und klicken Sie auf **OK** .

Das Fenster **IBM i NetServer File Share-GAMMA** wird geschlossen, und WMQ wird im Fenster für gemeinsam genutzte Objekte aufgelistet.

4. Klicken Sie mit der rechten Maustaste auf **WMQ** im Fenster für gemeinsam genutzte Objekte. Klicken Sie auf **File > Permissions**.

Ein Fenster wird geöffnet, **Qmgrs Permissions-GAMMA** , für das Objekt /QIBM/UserData/mqm/qmgrs.

- a) Überprüfen Sie die folgenden Berechtigungen für QMQM, wenn sie noch nicht festgelegt sind:

- Read
- Write
- Execute
- Management
- Existence
- Alter
- Reference

- b) Überprüfen Sie die folgenden Berechtigungen für QMQMADM, wenn sie noch nicht festgelegt sind:

- Read
- Write
- Execute
- Reference

- c) Fügen Sie weitere Benutzerprofile hinzu, die Sie Berechtigungen für /QIBM/UserData/mqm/qmgrs erteilen möchten.

Sie können z. B. die Standardberechtigungen für Benutzerprofil (Public) Read und Execute für /QIBM/UserData/mqm/qmgrs angeben.

- Überprüfen Sie, ob alle Benutzerprofile, denen der Zugriff auf /QIBM/UserData/mqm/qmgrs auf GAMMA gewährt wird, dasselbe Kennwort wie auf den Servern haben, die auf GAMMA zugreifen.

Stellen Sie insbesondere sicher, dass die QMQM -Benutzerprofile auf anderen Servern, die auf die Freigabe zugreifen werden, dasselbe Kennwort wie das QMQM -Benutzerprofil auf GAMMA haben.

Tipp: Klicken Sie auf den Ordner My Connections/GAMMA/Users and Groups in der System i Navigator , um die Kennwörter festzulegen. Alternativ können Sie die Befehle **CHFUSRPRF** und **CHGPWD** verwenden.

Ergebnisse

Überprüfen Sie, ob Sie mit der Freigabe auf GAMMA von anderen Servern zugreifen können. Wenn Sie die anderen Tasks ausführen, überprüfen Sie, dass Sie mit dem Pfad /QNTC/GAMMA/WMQ auf GAMMA von ALPHA und BETA zugreifen können. Wenn das Verzeichnis /QNTC/GAMMA auf ALPHA oder BETA nicht vorhanden ist, müssen Sie das Verzeichnis erstellen. Abhängig von der NetServer-Domäne müssen Sie möglicherweise ALPHA oder BETA erneut starten, bevor Sie das Verzeichnis erstellen.

```
CRTDIR DIR('/QNTC/GAMMA')
```

Wenn Sie geprüft haben, ob Sie über ALPHA oder BETA Zugriff auf /QNTC/GAMMA/WMQ haben, CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ') erstellt /QIBM/UserData/mqm/qmgrs/QM1 auf GAMMA.

Nächste Schritte

Erstellen Sie anhand der Anweisungen in Abschnitt „[Multi-Instanz-Warteschlangenmanager unter Verwendung von Journalspiegelung und NetServer unter IBM i erstellen](#)“ auf Seite 314 oder „[Einzel-Instanz-Warteschlangenmanager mithilfe von NetServer und Journalspiegelung unter IBM i in einen Multi-Instanz-Warteschlangenmanager konvertieren](#)“ auf Seite 318 einen Warteschlangenmanager mit mehreren Instanzen.

IBM i *Failover-Leistung unter IBM i*

Die Zeit, die zum Erkennen einer Warteschlangenmanagerinstanz benötigt wird, ist fehlgeschlagen, und die Wiederaufnahme der Verarbeitung auf einem Standby-Server kann in Abhängigkeit von der Konfiguration zwischen zehn Sekunden und fünfzehn Minuten oder mehr erfolgen. Bei der Planung und Erprobung einer Hochverfügbarkeitslösung muss die Leistung eine wichtige Überlegung sein.

Es gibt Vor- und Nachteile, um zu entscheiden, ob ein Warteschlangenmanager mit mehreren Instanzen für die Verwendung der Journalreplikation konfiguriert werden soll oder ob ein IASP verwendet werden soll. Für die Spiegelung ist es erforderlich, dass der Warteschlangenmanager synchron in ein fernes Journal schreibt. Aus Sicht der Hardware muss sich diese Leistung nicht auf die Leistung auswirken, aber aus einer Softwareperspektive besteht eine größere Pfadlänge beim Schreiben in ein fernes Journal als nur in einem lokalen Journal, und es wird möglicherweise erwartet, dass die Leistung eines aktiven Warteschlangenmanagers in einem gewissen Umfang reduziert wird. Wenn der Standby-Warteschlangenmanager jedoch übernimmt, ist die Verzögerung bei der Synchronisierung des lokalen Journals vom fernen Journal, das von der aktiven Instanz verwaltet wird, bevor sie fehlgeschlagen ist, in der Regel klein im Vergleich zu der Zeit, die IBM i benötigt, um den IASP zu erkennen und auf den Server zu übertragen, auf dem die Standby-Instanz des Warteschlangenmanagers ausgeführt wird. Die IASP-Übertragungszeiten können bis zu zehn bis fünfzehn Minuten betragen, anstatt sie in Sekunden zu vervollständigen. Die IASP-Übertragungszeit hängt von der Anzahl der Objekte ab, die *angehängt* werden müssen, wenn der IASP in das Bereitschaftssystem übertragen wird, und die Größe der Zugriffspfade oder Indizes, die zusammengeführt werden müssen, ist.

Wenn der Standby-Warteschlangenmanager übernimmt, ist die Verzögerung beim Synchronisieren des lokalen Journals vom fernen Journal, das von der aktiven Instanz verwaltet wird, bevor sie fehlgeschlagen ist, in der Regel klein im Vergleich zu der Zeit, die IBM i benötigt, um den unabhängigen ASP zu erkennen

und auf den Server zu übertragen, auf dem die Standby-Instanz des Warteschlangenmanagers ausgeführt wird. Die Übertragungszeiten des unabhängigen ASP können bis zu zehn bis fünfzehn Minuten betragen, anstatt sie in Sekunden zu vervollständigen. Die Zeit der unabhängigen ASP-Übertragung hängt von der Anzahl der Objekte ab, die *angehängt* werden müssen, wenn der unabhängige ASP in das Bereitschaftssystem übertragen wird und die Größe der Zugriffspfade oder Indizes, die zusammengeführt werden müssen, ist.

Die Übertragung des Journals ist jedoch nicht der einzige Faktor, der Einfluss auf die Zeit hat, die für die vollständige Wiederaufnahme der Standby-Instanz benötigt wird. Sie müssen außerdem die Zeit in Betracht ziehen, die es für das Netzdateisystem benötigt, um die Sperre für WS-Manager-Daten freizugeben, die Signale an die Standby-Instanz signalisiert, um mit ihrem Start fortzufahren, sowie die Zeit, die benötigt wird, um die Warteschlangen aus dem Journal wiederherzustellen, so dass die Instanz in der Lage ist, die Verarbeitung von Nachrichten erneut zu starten. Diese anderen Verzögerungsquellen addieren alle zu der Zeit, die zum Starten einer Standby-Instanz benötigt wird. Die Gesamtzeit zum Umschalten besteht aus den folgenden Komponenten:

Fehlererkennungszeit

Die Zeit, die für NFS benötigt wird, um die Sperre für die WS-Manager-Daten freizugeben, und die Standby-Instanz, um den Startprozess fortzusetzen.

Übertragungszeit

Bei einem HA-Cluster die Zeit, die IBM i benötigt, um den IASP vom System, auf dem sich die aktive Instanz befindet, in die Standby-Instanz zu übertragen, und im Fall der Journalreplikation die Zeit, die benötigt wird, um das lokale Journal in der Bereitschaftsdatenbank mit den Daten des fernen Replikats zu aktualisieren.

Zeit für Neustart

Die Zeit, die für die neu aktive WS-Manager-Instanz benötigt wird, um ihre Warteschlangen vom letzten Prüfpunkt in ihrem wiederhergestellten Journal wiederherzustellen und die Verarbeitung von Nachrichten wieder aufzunehmen.

Anmerkung:

Wenn die Standby-Instanz, die übernommen wurde, so konfiguriert ist, dass sie sich synchron auf die zuvor aktive Instanz repliziert, könnte der Start verzögert werden. Die neue aktivierte Instanz kann möglicherweise nicht in ihr fernes Journal replizieren, wenn sich das ferne Journal auf dem Server befindet, auf dem sich die zuvor aktive Instanz befindet, und der Server ist ausgefallen.

Die Standardzeit für das Warten auf eine synchrone Antwort ist eine Minute. Sie können die maximale Verzögerung vor dem Replikationszeitout konfigurieren. Alternativ können Sie Standby-Instanzen so konfigurieren, dass sie mit der asynchronen Replikation in der fehlgeschlagenen aktiven Instanz beginnen. Später wechseln Sie in die synchrone Replikation, wenn die fehlgeschlagene Instanz wieder im Bereitschaftsmodus ausgeführt wird. Dieselbe Überlegung gilt auch für die Verwendung von synchronen unabhängigen ASP-Spiegeln

Sie können separate Referenzkonfigurationsmessungen für diese Komponenten vornehmen, um Sie bei der Beurteilung der Gesamtzeit für die Funktionsübernahme zu unterstützen und um die Entscheidung zu berücksichtigen, die von der Konfiguration verwendet werden soll. Wenn Sie die beste Konfigurationsentscheidung treffen, müssen Sie auch überlegen, wie andere Anwendungen auf demselben Server eine Funktionsübernahme haben und ob es Sicherungs- oder Disaster-Recovery-Prozesse gibt, die bereits IASP verwenden.

Die IASP-Übertragungszeiten können durch Optimierung der Clusterkonfiguration verkürzt werden:

1. Benutzerprofile auf allen Systemen im Cluster sollten über die gleiche GID und UID verfügen, um die Notwendigkeit des Änderungsprozesses für die Änderung von UIDs und GIDs zu eliminieren.
2. Minimieren Sie die Anzahl der Datenbankobjekte im System- und Basisbenutzerplattenpool, da diese zusammengeführt werden müssen, um die Querverweistabelle für die Plattenpoolgruppe zu erstellen.
3. Weitere Tipps zur Leistung finden Sie im IBM Redbook, *Implementing PowerHA for IBM i*, SG24-7405.

Eine Konfiguration mit Basis-ASPs, Journalspiegelung und eine kleine Konfiguration sollte in der Größenordnung von 10 Sekunden umschalten.

Übersicht über die Kombination von IBM i-Clustering-Funktionen mit IBM MQ-Clustering

Wenn Sie IBM MQ unter IBM i ausführen und die IBM i-Clustering-Funktionen nutzen, können Sie eine umfassendere Hochverfügbarkeitslösung bereitstellen als nur mit dem IBM MQ-Clustering.

Um diese Funktion zu haben, müssen Sie Folgendes konfigurieren:

1. Cluster auf Ihrer IBM i-Maschine (siehe „[IBM i-Cluster](#)“ auf Seite 304)
2. Einen unabhängigen Zusatzspeicherpool, in den der Warteschlangenmanager verschoben wird (siehe „[Unabhängige Zusatzspeicherpools \(IASPs\)](#)“ auf Seite 304)
3. Eine Clusterressourcengruppe (siehe „[Einheiten-Clusterressourcengruppen](#)“ auf Seite 304), in der Sie Folgendes definieren:
 - Wiederherstellungsdomäne
 - IASP
 - Exitprogramm (siehe „[Exitprogramm für Einheiten-CRG](#)“ auf Seite 305)

IBM i-Cluster

Ein IBM i-Cluster ist eine Zusammenstellung von Instanzen, d. h. IBM i-Computer oder -Partitionen, die logisch miteinander verbunden sind.

Der Zweck dieser Gruppierung besteht darin, dass jede Instanz gesichert werden kann, wodurch ein Single Point of Failure eliminiert wird und die Anwendungs- und Datenausfallsicherheit erhöht werden kann. Wenn ein Cluster erstellt wird, können die verschiedenen Typen der Clusterressourcengruppe (CRG) konfiguriert werden, um Anwendungen, Daten und Einheiten im Cluster zu verwalten.

Weitere Informationen finden Sie im Abschnitt [Cluster erstellen](#) und den Befehl [Cluster erstellen \(CRTCLU\)](#).

Unabhängige Zusatzspeicherpools (IASPs)

Ein IASP ist ein Typ von Benutzer-ASP, der als Erweiterung des Speichers auf der Einzelebene dient. Es ist ein Teil des Speichers, der aufgrund seiner Unabhängigkeit vom Systemspeicher leicht manipuliert werden kann, ohne dass ein IPL für das System durchgeführt werden muss.

Ein IASP kann ohne großen Aufwand in eine andere Betriebssysteminstanz umgestellt oder in einen Ziel-IASP in einer anderen Betriebssysteminstanz repliziert werden. Es können zwei Methoden zum Umschalten eines IASP zwischen Instanzen verwendet werden:

- Die erste Methode erfordert, dass alle Computer im Cluster und der umschaltbare Plattenturm, der den IASP enthält, mit einer HSL-Schleife (High Speed Link) verbunden werden.
- Bei der zweiten Methode müssen die Betriebssysteminstanzen Partitionen auf demselben IBM i-Computer sein, auf dem Ein-/Ausgabeprozessoren (IOPs) zwischen den Partitionen umgeschaltet werden können. Es ist keine spezielle Hardware erforderlich, um einen IASP replizieren zu können. Die Replikation wird unter Verwendung von TCP/IP über das Netz ausgeführt.

Weitere Informationen enthält der Befehl [Configure Device ASP \(CFGDEVASP\)](#).

Einheiten-Clusterressourcengruppen

Es gibt mehrere Typen von Clusterressourcengruppen (CRGs). Weitere Informationen zu den verschiedenen Typen von CRGs finden Sie im Abschnitt [Clusterressourcengruppe](#).

Dieses Thema konzentriert sich auf eine Einheiten-CRG. Eine Einheiten-CRG:

- Beschreibt und verwaltet Einheitenressourcen, wie z. B. unabhängige Zusatzspeicherpools (IASPs).
- Definiert die Wiederherstellungsdomäne der Clusterknoten.
- Ordnet eine Einheit zu und

- Ordzeichnet das Exitprogramm zu, das Clusterereignisse verarbeitet.

Die Wiederherstellungsdomäne gibt an, welcher Clusterknoten als Primärknoten betrachtet wird. Der Rest der Knoten wird als Sicherungen betrachtet. Die Ausweichknoten werden auch in der Wiederherstellungsdomäne angeordnet. Geben Sie dabei an, welcher Knoten die erste Sicherung, die zweite Sicherung usw. ist, je nachdem, wie viele Knoten in der Wiederherstellungsdomäne vorhanden sind.

Wenn ein Primärknotenfehler aufgetreten ist, wird das Exitprogramm auf allen Knoten in der Wiederherstellungsdomäne ausgeführt. Das Exitprogramm, das in der ersten Sicherung ausgeführt wird, kann dann die erforderlichen Initialisierungen vornehmen, um diesen Knoten zu einem neuen Primärknoten zu machen.

Weitere Informationen finden Sie unter [Einheiten-CRGs erstellen](#) und den Befehl [CRG erstellen \(CRG erstellen\)](#).

Exitprogramm für Einheiten-CRG

Der Betriebssystem-Cluster-Ressourcenservice ruft ein Exitprogramm für Einheiten-CRG auf, wenn ein Ereignis in einem der Knoten auftritt, die die Wiederherstellungsdomäne definiert, z. B. ein Failover oder Switchover.

Ein Failover-Ereignis tritt auf, wenn der Primärknoten des Clusters ausfällt und die CRGs mit allen Ressourcen, die sie verwalten, umgeschaltet werden, und ein Switchover-Ereignis auftritt, wenn eine bestimmte CRG manuell vom Primärknoten auf den Ausweichknoten umgeschaltet wird.

In jedem Fall ist das Exitprogramm für die Initialisierung und das Starten aller Programme zuständig, die auf dem vorherigen Primärknoten ausgeführt wurden, wodurch der erste Ausweichknoten in den neuen Primärknoten konvertiert wird.

Beispiel: Mit IBM MQ sollte das Exitprogramm für das Starten des IBM MQ-Subsystems (QMQM) und von Warteschlangenmanagern verantwortlich sein. Warteschlangenmanager sollten so konfiguriert werden, dass Empfangsprogramme und Services, wie z. B. Auslösemonitore, automatisch gestartet werden.

Umschaltbare IASP-Konfiguration

IBM MQ kann so konfiguriert werden, dass es die Clustering-Funktionen von IBM i nutzt. Gehen Sie dazu wie folgt vor:

1. Erstellen Sie einen IBM i-Cluster zwischen den Rechenzentrumssystemen.
2. Den WS-Manager in einen IASP verschieben.

„Warteschlangenmanager in oder aus einem unabhängigen Zusatzspeicherpool versetzen oder aus einem unabhängigen Zusatzspeicherpool entfernen“ auf Seite 306 enthält Beispielcode, der Sie bei der Ausführung dieser Operation unterstützt.

3. Es muss eine CRG erstellt werden, die die Wiederherstellungsdomäne, den IASP und das Exitprogramm definiert.

„Einheiten-Clusterressourcengruppe konfigurieren“ auf Seite 305 enthält Beispielcode, der Sie bei der Ausführung dieser Operation unterstützt.

Zugehörige Konzepte

„Unabhängige ASPs und hohe Verfügbarkeit“ auf Seite 327

Unabhängige ASPs aktivieren Anwendungen und Daten, die zwischen den Servern verschoben werden. Die Flexibilität unabhängiger ASPs bedeutet, dass sie die Basis für einige IBM i-Hochverfügbarkeitslösungen bilden. Wenn Sie überlegen, ob ein ASP oder ein unabhängiger ASP für das WS-Manager-Journal verwendet werden soll, sollten Sie eine andere Konfiguration mit hoher Verfügbarkeit auf der Basis unabhängiger ASPs berücksichtigen.

 [Einheiten-Clusterressourcengruppe konfigurieren](#)

Ein Beispielprogramm zum Festlegen einer Einheiten-Clusterressourcengruppe (CRG).

Informationen zu diesem Vorgang

Beachten Sie im folgenden Beispiel Folgendes:

- [PRIMARY SITE NAME] und [BACKUP SITE NAME] können beliebige zwei unterschiedliche Zeichenfolgen mit maximal acht Zeichen sein.
- [PRIMARY IP] und [BACKUP IP] sind die IPs, die für die Spiegelung verwendet werden sollen.

Vorgehensweise

1. Geben Sie den Namen des Clusters an.
2. Geben Sie den Namen und die Bibliothek des CRG-Exitprogramms an.
3. Bestimmen Sie den Namen des Primärknotens und der Ausweichknoten, die von dieser CRG definiert werden sollen.
4. Geben Sie den IASP an, der von dieser CRG verwaltet werden soll, und stellen Sie sicher, dass er unter dem Primärknoten erstellt wurde.
5. Erstellen Sie mit dem Befehl eine Einheitenbeschreibung in den Ausweichknoten:

```
CRTDEVASP DEVD([IASP NAME]) RSRNAME([IASP NAME])
```

6. Fügen Sie die Übernahme-IP-Adresse allen Knoten hinzu, indem Sie den folgenden Befehl verwenden:

```
ADDCPIFC INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. Starten Sie die IP-Übernahmeadresse nur im Primärknoten, indem Sie den folgenden Befehl verwenden:

```
STRTCPIFC INTNETADR(' [TAKEOVER IP]')
```

8. Optional: Wenn Ihr IASP umschaltbar ist, rufen Sie den folgenden Befehl auf:

```
CRTCRCG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT  
NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY) ([BACKUP NAME] *BACKUP))  
EXITPGMFMT(EXTP0200) CFGOBJ(( [IASP NAME] *DEVD *ONLINE ' [TAKEOVER IP]')
```

9. Optional: Wenn der IASP gespiegelt werden soll, den folgenden Befehl aufrufen:

```
CRTCRCG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY *LAST [PRIMARY SITE NAME] (' [PRIMARY  
IP]'))  
[BACKUP NAME] *BACKUP *LAST [BACKUP SITE NAME] (' [BACKUP IP]')) EXITPGMFMT(EXTP0200)  
CFGOBJ(( [IASP NAME] *DEVD *ONLINE ' [TAKEOVER IP]'))
```



Warteschlangenmanager in oder aus einem unabhängigen Zusatzspeicherpool versetzen oder aus einem unabhängigen Zusatzspeicherpool entfernen

Ein Beispielprogramm zum Verschieben eines Warteschlangenmanagers in einen unabhängigen Zusatzspeicherpool (IASP) und Befehle zum Entfernen eines Warteschlangenmanagers aus einem IASP.

Informationen zu diesem Vorgang

Beachten Sie im folgenden Beispiel Folgendes:

- [MANAGER NAME] ist der Name Ihres Warteschlangenmanagers.
- [IASP NAME] ist der Name Ihres IASP.
- [MANAGER LIBRARY] ist der Name Ihrer WS-Manager-Bibliothek.
- [MANAGER DIRECTORY] ist der Name Ihres WS-Manager-Verzeichnisses.

Vorgehensweise

1. Identifizieren Sie den Primärknoten und die Ausweichknoten.
2. Gehen Sie wie folgt vor, um die folgende Prozedur auf Ihrem Primärknoten zu verwenden:

- a) Stellen Sie sicher, dass der WS-Manager beendet wurde.
- b) Stellen Sie sicher, dass Ihr IASP vary on mit dem Befehl

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

- c) Erstellen Sie das WS-Manager-Verzeichnis unter dem IASP.
Es wird ein Verzeichnis unter dem Stammverzeichnis mit dem Namen des IASP vorhanden sein.
Dies ist:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- d) Versetzen Sie die IFS-Objekte Ihres Managers in das Verzeichnis mit den WS-Managern, das Sie gerade unter dem IASP erstellt haben. Verwenden Sie dazu den folgenden Befehl:

```
QSH CMD('mv /QIBM/UserData/mqm/qmgrs/[MANAGER NAME]  
/[IASP NAME]/QIBM/UserData/mqm/qmgrs/')
```

- e) Erstellen Sie eine temporäre Sicherungsdatei mit dem Namen MGRLIB mit dem folgenden Befehl:

```
CRTSAVF QGPL/MGRLIB
```

- f) Speichern Sie die WS-Manager-Bibliothek in der MGRLIB -Sicherungsdatei, indem Sie den folgenden Befehl verwenden:

```
SAVLIB LIB([MANGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
```

- g) Löschen Sie die WS-Manager-Bibliothek, indem Sie den folgenden Befehl verwenden, und ignorieren Sie alle Anfragenachrichten:

```
DLTLIB [MANAGER LIBRARY]
```

- h) Stellen Sie die WS-Manager-Bibliothek in den IASP zurück, indem Sie den folgenden Befehl verwenden:

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)  
RSTASPDEV([IASP NAME])
```

- i) Löschen Sie die temporäre Sicherungsdatei mit dem folgenden Befehl:

```
DLTF FILE(QGPL/MGRLIB)
```

- j) Erstellen Sie eine symbolische Verbindung zu den IFS-Objekten des Warteschlangenmanagers unter dem IASP, indem Sie den folgenden Befehl verwenden:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

- k) Hängen Sie den IASP mit dem folgenden Befehl an:

```
SETASPGRP [IASP NAME]
```

- l) Starten Sie den WS-Manager mit dem folgenden Befehl:

```
STRMQM [MANAGER NAME]
```

3. Gehen Sie wie folgt vor, um die folgende Prozedur auf Ihrem Ausweichknoten oder Knoten zu verwenden:

a) Erstellen Sie mit dem folgenden Befehl ein temporäres WS-Manager-Verzeichnis:

```
QSH CMD('mkdir -p /[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

b) Erstellen Sie mit dem folgenden Befehl eine symbolische Verbindung zum temporären Verzeichnis des Warteschlangenmanagers:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

c) Löschen Sie das temporäre Verzeichnis mit dem folgenden Befehl:

```
QSH CMD('rm -r /[IASP NAME]')
```

d) Fügen Sie am Ende der Datei /QIBM/UserData/mqm/mqs.ini Folgendes hinzu:

```
QueueManager:  
Name=[MANAGER NAME]  
Prefix=/QIBM/UserData/mqm  
Library=[MANAGER LIBRARY]  
Directory=[MANAGER DIRECTORY]
```

4. Geben Sie die folgenden Befehle aus, um einen WS-Manager aus einem IASP zu entfernen:

- a) VRYCFG CFGOBJ ([IASP NAME]) CFGTYPE (*DEV) STATUS (*ON)
- b) SETASPGRP [IASP-NAME]
- c) ENDMQM [MANAGER NAME]
- d) DLTMQM [MANAGER NAME]

IBM i Konfiguration mit gespiegeltem Journal für ASP unter IBM i

Konfigurieren Sie einen stabilen Multi-Instanz-WS-Manager mit synchroner Replikation zwischen gespiegelten Journalen.

In einer Konfiguration mit gespiegelten WS-Managern werden Journale verwendet, die in Basis- oder unabhängigen Zusatzspeicherpools (ASP) erstellt werden.

Unter IBM i werden Warteschlangenmanagerdaten in Journale und in ein Dateisystem geschrieben. Journale enthalten die Masterkopie der WS-Manager-Daten. Journale werden von Systemen gemeinsam genutzt, die entweder synchrone oder asynchrone Journalreplikation verwenden. Es ist eine Mischung aus lokalen und fernen Journalen erforderlich, um eine Warteschlangenmanagerinstanz erneut zu starten. Der Neustart des Warteschlangenmanagers liest Journalsätze aus der Mischung aus lokalen und fernen Journalen auf dem Server und den WS-Manager-Daten im gemeinsam genutzten Netzdateisystem. Die Daten in dem Dateisystem beschleunigen den Neustart des Warteschlangenmanagers. Prüfpunkte werden im Dateisystem gespeichert, wobei die Synchronisationspunkte zwischen dem Dateisystem und den Journalen markiert werden. Journalsätze, die vor dem Prüfpunkt gespeichert werden, sind für einen typischen WS-Manager-Neustarts nicht erforderlich. Die Daten im Dateisystem sind jedoch möglicherweise nicht auf dem neuesten Stand, und Journalsätze nach dem Prüfpunkt werden verwendet, um den Neustart des Warteschlangenmanagers abzuschließen. Die Daten in den Journalen, die an die Instanz angehängt sind, werden auf dem neuesten Stand gehalten, so dass der Neustart erfolgreich abgeschlossen werden kann.

Aber selbst die Journalsätze sind möglicherweise nicht auf dem neuesten Stand, wenn das ferne Journal auf dem Standby-Server asynchron repliziert wurde und der Fehler aufgetreten ist, bevor er synchronisiert wurde. Wenn Sie sich entscheiden, einen Warteschlangenmanager unter Verwendung eines fernen Journals erneut zu starten, das nicht synchronisiert ist, kann die Standby-WS-Manager-Instanz entweder Nachrichten erneut verarbeiten, die gelöscht wurden, bevor die aktive Instanz fehlgeschlagen ist, oder keine Prozessnachrichten verarbeiten, die empfangen wurden, bevor die aktive Instanz fehlgeschlagen ist.

Eine weitere, seltene Möglichkeit besteht darin, dass das Dateisystem den letzten Prüfpunktsatz enthält und ein nicht synchronisiertes fernes Journal in der Bereitschaftsdatenbank nicht. In diesem Fall wird der WS-Manager nicht automatisch erneut gestartet. Sie können warten, bis das ferne Journal synchronisiert

ist, oder einen Kaltstart des Standby-WS-Managers aus dem Dateisystem. Auch wenn in diesem Fall das Dateisystem einen neueren Prüfpunkt der WS-Manager-Daten enthält als das ferne Journal, enthält es möglicherweise nicht alle Nachrichten, die verarbeitet wurden, bevor die aktive Instanz ausgefallen ist. Einige Nachrichten werden möglicherweise erneut verarbeitet, und einige werden nach einem Kaltstart, der nicht mit den Journalen synchronisiert ist, nicht verarbeitet.

Mit einem Warteschlangenmanager mit mehreren Instanzen wird auch das Dateisystem verwendet, um zu steuern, welche Instanz eines Warteschlangenmanagers aktiv ist und welche Standby-Instanz vorhanden ist. Die aktive Instanz fordert eine Sperre für die WS-Manager-Daten an. Die Bereitschaftsdatenbank wartet darauf, die Sperre anzufordern, und wenn sie ausgeführt wird, wird sie zur aktiven Instanz. Die Sperre wird von der aktiven Instanz freigegeben, wenn sie normal beendet wird. Die Sperre wird vom Dateisystem freigegeben, wenn das Dateisystem feststellt, dass die aktive Instanz ausgefallen ist, oder kann nicht auf das Dateisystem zugreifen. Das Dateisystem muss die Voraussetzungen für die Erkennung von Fehlern erfüllen. Weitere Informationen finden Sie im Abschnitt Anforderungen für gemeinsam genutzte Dateisysteme.

Die Architektur von Multi-Instanz-Warteschlangenmanagern unter IBM i ermöglicht einen automatischen Neustart nach einem Server- oder Warteschlangenmanagerfehler. Außerdem unterstützt es die Wiederherstellung von WS-Manager-Daten nach einem Ausfall des Dateisystems, in dem die WS-Manager-Daten gespeichert werden.

Wenn ALPHA in Abbildung 36 auf Seite 310 ausfällt, können Sie QM1 auf BETA auf Basis des gespiegelten Journals manuell erneut starten. Durch das Hinzufügen der Multi-Instanz-WS-Manager-Funktionalität zu QM1 wird die Standby-Instanz von QM1 automatisch auf BETA aufgenommen, wenn die aktive Instanz auf ALPHA fehlschlägt. QM1 kann auch automatisch wieder aufgenommen werden, wenn es sich um den Server ALPHA handelt, der fehlschlägt, und nicht nur die aktive Instanz von QM1. Sobald BETA zum Host der aktiven WS-Manager-Instanz wird, kann die Standby-Instanz auf ALPHA gestartet werden.

In Abbildung 36 auf Seite 310 ist eine Konfiguration zu sehen, bei der Journale zwischen zwei Instanzen eines Warteschlangenmanagers gespiegelt werden; dabei wird NetServer zum Speichern der WS-Manager-Daten verwendet. Sie können das Muster erweitern, um weitere Journale und damit mehr Instanzen einzuschließen. Dabei müssen Sie sich an die im Abschnitt „Warteschlangenmanagerjournale unter IBM i“ auf Seite 287 dargelegten Namensregeln für Journale halten. Derzeit ist die Anzahl der aktiven Instanzen eines Warteschlangenmanagers auf zwei begrenzt, einer ist aktiv und einer befindet sich im Standby-Modus.

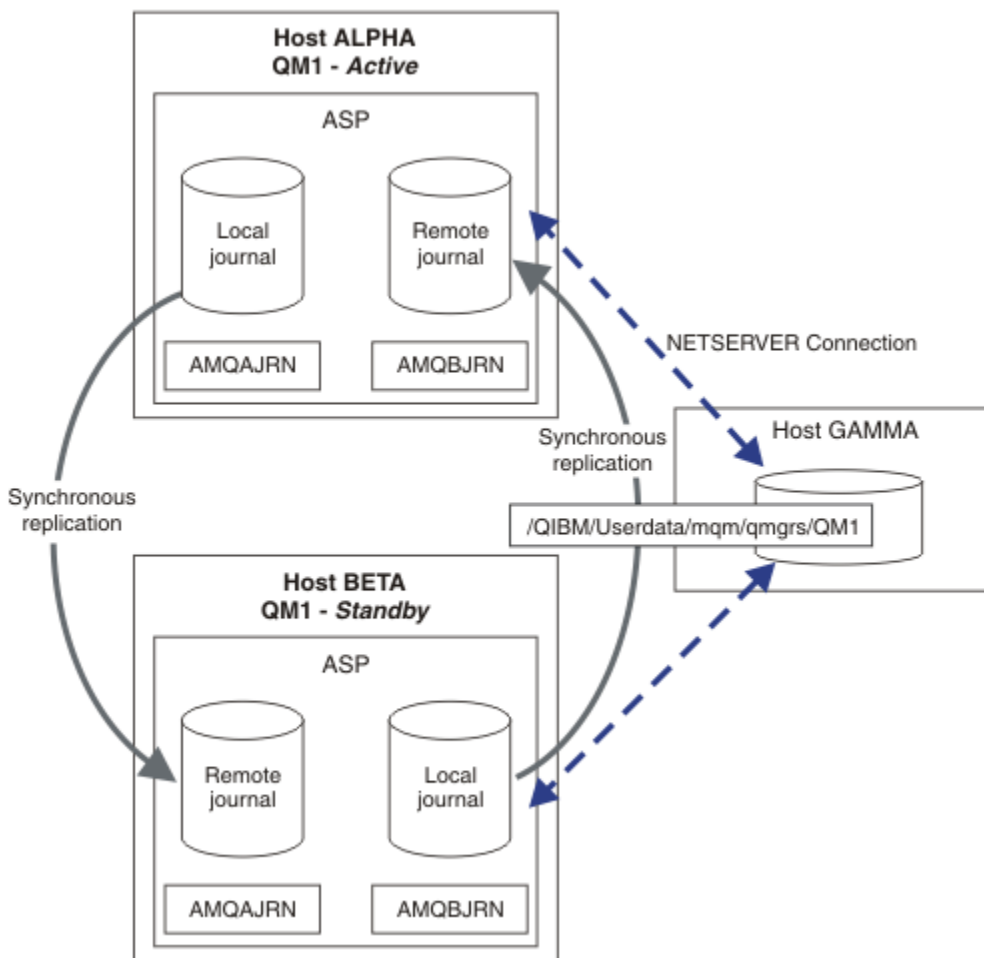


Abbildung 36. WS-Manager-Journal spiegeln

Das lokale Journal für QM1 auf dem Host ALPHA heißt AMQAJRN (oder vollständiger, QMQM1/AMQAJRN) und auf BETA das Journal ist QMQM1/AMQBJRN. Jedes lokale Journal repliziert in ferne Journale auf allen anderen Instanzen des Warteschlangenmanagers. Wenn der WS-Manager mit zwei Instanzen konfiguriert ist, wird ein lokales Journal in ein fernes Journal repliziert.

Replikation des fernen Journals *SYNC oder *ASYN

IBM i-Journale werden entweder durch synchrone (*SYNC) oder asynchrone (*ASYN) Journalführung gespiegelt (siehe [Verwaltung ferner Journale](#)).

Der Replikationsmodus in [Abbildung 36 auf Seite 310](#) ist *SYNC, nicht *ASYN. *ASYN ist schneller, aber wenn ein Fehler auftritt, wenn der ferne Journalstatus *ASYNCPENDist, sind das lokale und das ferne Journal nicht konsistent. Das ferne Journal muss sich mit dem lokalen Journal abfangen. Wenn Sie *SYNC auswählen, wartet das lokale System auf das ferne Journal, bevor es von einem Aufruf zurückkehrt, der einen vollständigen Schreibzugriff erfordert. Die lokalen und fernen Journale bleiben in der Regel konsistent miteinander. Nur wenn die *SYNC -Operation länger als eine bestimmte Zeit dauert¹ und die ferne Journalführung ist inaktiviert. Führen Sie die Journale nicht mehr synchron aus. Ein Fehler wird an der Journalnachrichtenwarteschlange und an QSYSOPR protokolliert. Der WS-Manager erkennt diese Nachricht, schreibt einen Fehler in das Fehlerprotokoll des Warteschlangenmanagers und inaktiviert die ferne Replikation des WS-Manager-Journals. Die aktive WS-Manager-Instanz wird ohne ferne Journalführung in diesem Journal wieder aufgenommen. Wenn der ferne Server wieder verfügbar ist, müssen Sie die synchrone ferne Journalreplikation manuell reaktivieren. Die Journale werden dann resynchronisiert.

¹ Die angegebene Zeit beträgt 60 Sekunden auf IBM i 5 und im Bereich von 1-3600 Sekunden auf IBM i 6.1 und weiter.

Ein Problem der *SYNC/*SYNC-Konfiguration in [Abbildung 36](#) auf [Seite 310](#) ist die Übernahme der Steuerung durch die Standby-Instanz des Warteschlangenmanagers auf BETA. Sobald die WS-Manager-Instanz auf BETA ihre erste persistente Nachricht schreibt, versucht sie, das ferne Journal auf ALPHA zu aktualisieren. Wenn die Ursache für die Steuerung von ALPHA in BETA der Ausfall von ALPHA war und ALPHA noch nicht ausgeführt wird, schlägt die ferne Journalführung für ALPHA fehl. BETA wartet darauf, dass ALPHA antwortet, und inaktiviert dann die ferne Journalführung und setzt die Verarbeitung von Nachrichten mit nur der lokalen Journalführung wieder ein. BETA muss eine Weile warten, um zu erkennen, dass ALPHA inaktiv ist, was zu einer Inaktivität führt.

Die Auswahl zwischen dem Festlegen der fernen Journalaufzeichnung auf *SYNC oder *ASYNCR ist ein Trade-off. In [Tabelle 22](#) auf [Seite 311](#) sind die Unterschiede der Journalführung im *SYNC- und im *ASYNCR-Modus zwischen zwei Warteschlangenmanagern aufgeführt.

Aktiv	Standby	*SYNC	*ASYNCR
*SYNC		<ol style="list-style-type: none"> 1. Konsistentes Switchover und Failover 2. Die Standby-Instanz wird nach dem Failover nicht sofort wieder aufgenommen. 3. Die ferne Journalführung muss die ganze Zeit verfügbar sein. 4. Die Leistung des WS-Managers hängt von der fernen Journalführung ab. 	<ol style="list-style-type: none"> 1. Konsistentes Switchover und Failover 2. Das ferne Journaling muss auf *SYNC umgeschaltet werden, wenn der Standby-Server verfügbar ist. 3. Die ferne Journalfunktion muss verfügbar bleiben, nachdem sie erneut gestartet wurde. 4. Die Leistung des WS-Managers hängt von der fernen Journalführung ab.
*ASYNCR		<ol style="list-style-type: none"> 1. Keine sinnvolle Kombination 	<ol style="list-style-type: none"> 1. Einige Nachrichten können nach einem Failover oder Switchover verloren gehen oder dupliziert werden. 2. Die Standby-Instanz muss nicht immer verfügbar sein, damit die aktive Instanz ohne Verzögerung fortgesetzt werden kann. 3. Die Leistung hängt nicht von der fernen Journalführung ab.

***SYNC / *SYNC**

Die aktive WS-Manager-Instanz verwendet die *SYNC-Journalfunktion. Wenn die Standby-WS-Manager-Instanz gestartet wird, versucht sie sofort, die *SYNC-Journalfunktion zu verwenden.

1. Das ferne Journal ist transaktionskonsistent mit dem lokalen Journal des aktiven Warteschlangenmanagers. Wenn der WS-Manager auf die Standby-Instanz umgeschaltet wird, kann er sofort wieder aufgenommen werden. Die Standby-Instanz wird normalerweise ohne Verlust oder Duplizierung von Nachrichten wieder aufgenommen. Nachrichten gehen nur verloren oder werden dupliziert, wenn die ferne Journalführung seit dem letzten Prüfpunkt ausgefallen ist und der zuvor aktive Warteschlangenmanager nicht erneut gestartet werden kann.
2. Wenn der WS-Manager in der Standby-Instanz ausfällt, kann er möglicherweise nicht sofort gestartet werden. Die Standby-Warteschlangenmanagerinstanz wird mit dem Journaling *SYNC aktiviert. Die Ursache für die Funktionsübernahme kann die ferne Journalführung für den Server, auf dem sich die Standby-Instanz befindet, verhindern. Der Warteschlangenmanager wartet, bis das Problem erkannt wird, bevor persistente Nachrichten verarbeitet werden. Ein Fehler wird an der Journalnachrichtwarteschlange und an QSYSOPR protokolliert. Der WS-Manager erkennt diese Nachricht, schreibt einen Fehler in das Fehlerprotokoll des Warteschlangenmanagers und

inaktiviert die ferne Replikation des WS-Manager-Journals. Die aktive WS-Manager-Instanz wird ohne ferne Journalführung in diesem Journal wieder aufgenommen. Wenn der ferne Server wieder verfügbar ist, müssen Sie die synchrone ferne Journalreplikation manuell reaktivieren. Die Journale werden dann resynchronisiert.

3. Der Server, auf dem das ferne Journal repliziert wird, muss immer verfügbar sein, um das ferne Journal zu verwalten. Das ferne Journal wird in der Regel auf denselben Server repliziert, auf dem sich der Standby-Warteschlangenmanager befindet. Der Server ist möglicherweise nicht verfügbar. Ein Fehler wird an der Journalnachrichtenwarteschlange und an QSYOPR protokolliert. Der WS-Manager erkennt diese Nachricht, schreibt einen Fehler in das Fehlerprotokoll des Warteschlangenmanagers und inaktiviert die ferne Replikation des WS-Manager-Journals. Die aktive WS-Manager-Instanz wird ohne ferne Journalführung in diesem Journal wieder aufgenommen. Wenn der ferne Server wieder verfügbar ist, müssen Sie die synchrone ferne Journalreplikation manuell reaktivieren. Die Journale werden dann resynchronisiert.
4. Die ferne Journalführung ist langsamer als die lokale Journalführung und ist wesentlich langsamer, wenn die Server durch einen großen Abstand voneinander getrennt sind. Der WS-Manager muss auf die ferne Journalführung warten, wodurch die Leistung des Warteschlangenmanagers reduziert wird.

Die *SYNC/*SYNC-Konfiguration zwischen einem Serverpaar hat den Nachteil einer Verzögerung beim Wiederaufnehmen der Standby-Instanz nach dem Failover. Für die Konfiguration *SYNC / *ASYN ist dieses Problem nicht aufgetreten.

*SYNC / *SYNC garantiert keinen Nachrichtenverlust nach Switchover oder Failover, sofern ein fernes Journal verfügbar ist. Wenn Sie das Risiko eines Nachrichtenverlusts nach einem Failover oder Switchover reduzieren möchten, haben Sie zwei Möglichkeiten. Stoppen Sie die aktive Instanz, wenn das ferne Journal inaktiv wird, oder erstellen Sie ferne Journale auf mehr als einem Server.

***SYNC / *ASYN**

Die aktive WS-Manager-Instanz verwendet die *SYNC -Journalfunktion, und wenn die Standby-Warteschlangenmanagerinstanz gestartet wird, wird die *ASYN -Journalfunktion verwendet. Kurz nachdem der Server, auf dem die neue Standby-Instanz installiert ist, verfügbar ist, muss der Systembediener das ferne Journal auf der aktiven Instanz in *SYNC umschalten. Wenn der Bediener die ferne Journalführung von *ASYN in *SYNC umschaltet, wird die aktive Instanz angehalten, wenn der Status des fernen Journals *ASYNPEND ist. Die aktive Warteschlangenmanagerinstanz wartet, bis die verbleibenden Journaleinträge in das ferne Journal übertragen werden. Wenn das ferne Journal mit dem lokalen Journal synchronisiert wurde, wird die neue Standby-Instanz mit der neuen aktiven Instanz transaktionskonsistent. Aus der Perspektive der Verwaltung von Multi-Instanz-Warteschlangenmanagern hat der IBM i-Systembediener in einer *SYNC / *ASYN-Konfiguration eine zusätzliche Aufgabe. Der Bediener muss die ferne Journalfunktion zusätzlich zu *SYNC umschalten, um die fehlgeschlagene WS-Manager-Instanz erneut zu starten.

1. Das ferne Journal ist transaktionskonsistent mit dem lokalen Journal des aktiven Warteschlangenmanagers. Wenn die Instanz des aktiven Warteschlangenmanagers umgeschaltet wird oder die Instanz der Standby-Instanz ausfällt, kann die Standby-Instanz sofort wieder aufgenommen werden. Die Standby-Instanz wird normalerweise ohne Verlust oder Duplizierung von Nachrichten wieder aufgenommen. Nachrichten gehen nur verloren oder werden dupliziert, wenn die ferne Journalführung seit dem letzten Prüfpunkt ausgefallen ist und der zuvor aktive Warteschlangenmanager nicht erneut gestartet werden kann.
2. Der Systembediener muss das ferne Journal von *ASYN in *SYNC umschalten, sobald das System, auf dem die aktive Instanz ausgeführt wird, wieder verfügbar wird. Der Bediener kann warten, bis das ferne Journal aufsteht, bevor das ferne Journal auf *SYNC umgeschaltet wird. Alternativ kann der Bediener die ferne Instanz sofort auf *SYNC umschalten und die aktive Instanz zwingen, zu warten, bis das Journal der Standby-Instanz aufgeholt wurde. Wenn das ferne Journaling auf *SYNGesetzt ist, ist die Standby-Instanz im Allgemeinen transaktionsorientiert mit der aktiven Instanz konsistent. Nachrichten gehen nur verloren oder werden dupliziert, wenn die ferne Journalführung seit dem letzten Prüfpunkt ausgefallen ist und der zuvor aktive Warteschlangenmanager nicht erneut gestartet werden kann.

3. Wenn die Konfiguration von einem Switchover oder Failover wiederhergestellt wurde, muss der Server, auf dem das ferne Journal gehostet wird, die ganze Zeit verfügbar sein.

Wählen Sie `*SYNC / *ASYNC` aus, wenn der Standby-WS-Manager nach einem Failover schnell wieder aufgenommen werden soll. Sie müssen die Einstellung für das ferne Journal in der neuen aktiven Instanz manuell auf `*SYNC` zurückschreiben. Die Konfiguration `*SYNC / *ASYNC` stimmt mit dem normalen Muster der Verwaltung eines Paares von WS-Managern mit mehreren Instanzen überein. Nachdem eine Instanz ausgefallen ist, gibt es eine Zeit, bevor die Standby-Instanz erneut gestartet wird, während der die aktive Instanz nicht umschlagen kann.

***ASYNC / *ASYNC**

Sowohl die Server, auf die sich die aktiven Warteschlangenmanager als auch die Standby-Warteschlangenmanager befinden, sind für die Verwendung der fernen Journaling `*ASYNC` konfiguriert.

1. Wenn die Switchover-oder Funktionsübernahme erfolgt, wird der Warteschlangenmanager mit dem Journal auf dem neuen Server fortgesetzt. Das Journal wird möglicherweise nicht synchronisiert, wenn das Switchover oder Failover stattfindet. Daher können Nachrichten verloren gehen oder dupliziert werden.
2. Die aktive Instanz wird ausgeführt, auch wenn der Server, auf dem sich der Standby-Warteschlangenmanager befindet, nicht verfügbar ist. Das lokale Journal wird asynchron mit dem Standby-Server repliziert, wenn es verfügbar ist.
3. Die Leistung des lokalen Warteschlangenmanagers wird durch die ferne Journalführung nicht beeinflusst.

Wählen Sie `*ASYNC / *ASYNC` aus, wenn die Leistung die Hauptvoraussetzung ist, und Sie sind bereit, einige Nachrichten nach einem Failover oder Switchover zu verlieren oder zu duplizieren.

***ASYNC / *SYNC**

Es gibt keinen Grund, diese Kombination von Optionen zu verwenden.

Aktivierung des Warteschlangenmanagers aus einem fernen Journal

Journale werden entweder synchron oder asynchron repliziert. Das ferne Journal ist möglicherweise nicht aktiv, oder es wird möglicherweise mit dem lokalen Journal abgeholt. Das ferne Journal kann aufgeholt werden, auch wenn es synchron repliziert wird, da es möglicherweise erst kürzlich aktiviert wurde. Die Regeln, die der Warteschlangenmanager auf den Status des fernen Journals bezieht, das er beim Start verwendet, ist wie folgt.

1. Der Standby-Start scheitert, wenn er aus dem fernen Journal in der Bereitschaftsdatenbank erneut ausgeführt werden muss und der Journalstatus `*FAILED` oder `*INACTPEND` ist.
2. Wenn die Aktivierung der Bereitschaftsdatenbank beginnt, muss der Status des fernen Journals in der Bereitschaftsdatenbank entweder `*ACTIVE` oder `*INACTIVE` sein. Wenn der Status `*INACTIVE` ist, kann die Aktivierung fehlschlagen, wenn nicht alle Journaldaten repliziert wurden.

Der Fehler tritt auf, wenn die WS-Manager-Daten im Netzdateisystem über einen aktuelleren Prüfpunktsatz als im fernen Journal vorhanden sind. Der Fehler wird wahrscheinlich nicht auftreten, solange das ferne Journal innerhalb des Standardintervalls von 30 Minuten zwischen den Prüfpunkten aktiviert ist. Wenn der Standby-WS-Manager einen aktuelleren Prüfpunktsatz aus dem Dateisystem liest, wird er nicht gestartet.

Sie haben die Wahl: Warten Sie, bis das lokale Journal auf dem aktiven Server wiederhergestellt werden kann, oder starten Sie den Standby-WS-Manager. Wenn Sie einen Kaltstart auswählen, wird der Warteschlangenmanager ohne Journaldaten gestartet und stützt sich auf die Konsistenz und Vollständigkeit der WS-Manager-Daten im Dateisystem.

Anmerkung: Wenn Sie einen WS-Manager kalt starten, laufen Sie Gefahr, Nachrichten nach dem letzten Prüfpunkt zu verlieren oder zu duplizieren. Die Nachrichtentransaktionen wurden in das Journal geschrieben, aber einige der Transaktionen wurden möglicherweise nicht in die WS-Manager-Daten im Dateisystem geschrieben. Beim Kaltstart eines Warteschlangenmanagers wird ein neues Journal gestartet, und die Transaktionen, die nicht in die WS-Manager-Daten im Dateisystem geschrieben werden, gehen verloren.

3. Die Aktivierung des Standby-Warteschlangenmanagers wartet auf den Status des fernen Journals in der Bereitschaftsdatenbank, um von *ASYNCPEND oder *SYNCPEND in *ASYNCR oder *SYNCR zu ändern. Nachrichten werden in regelmäßigen Abständen in das Jobprotokoll des Ausführungscontrollers geschrieben.

Anmerkung: In diesem Fall wartet die Aktivierung auf das ferne Journal auf dem lokalen Bereitschaftswarteschlangenmanager, das gerade aktiviert wird. Der WS-Manager wartet ebenfalls auf eine Zeit, bevor er ohne ein fernes Journal fortsetzt. Er wartet, wenn er versucht, synchron in sein fernes Journal (oder Journale) zu schreiben, und das Journal ist nicht verfügbar.

4. Die Aktivierung wird gestoppt, wenn sich der Journalstatus in *FAILED oder *INACTPEND ändert.

Die Namen und Status der lokalen und fernen Journale, die in der Aktivierung verwendet werden sollen, werden in das Fehlerprotokoll des Warteschlangenmanagers geschrieben.

Multi-Instanz-Warteschlangenmanager unter Verwendung von Journalspiegelung und NetServer unter IBM i erstellen

Erstellen Sie einen Multi-Instanz-Warteschlangenmanager, der auf zwei IBM i-Servern ausgeführt werden soll. Die Warteschlangenmanagerdaten werden unter Verwendung von NetServer auf einem dritten IBM i-Server gespeichert. Das WS-Manager-Journal wird unter Verwendung der fernen Journalfunktion zwischen den beiden Servern gespiegelt. Mit dem Befehl **ADDMQMJRN** können Sie die Erstellung der fernen Journale vereinfachen.

Vorbereitende Schritte

1. Für die Aufgabe sind drei IBM i-Server erforderlich. Installieren Sie IBM MQ auf zwei dieser Server; im Beispiel sind dies ALPHA und BETA. IBM MQ muss mindestens Version 7.0.1.1 sein.
2. Der dritte Server ist ein IBM i-Server, der über NetServer mit ALPHA und BETA verbunden ist. Er wird verwendet, um die WS-Manager-Daten gemeinsam zu nutzen. IBM MQ muss dort nicht installiert sein. Es ist jedoch hilfreich, IBM MQ zumindest temporär zu installieren, um die Warteschlangenmanagerverzeichnisse und berechtigungen einzurichten.
3. Stellen Sie sicher, dass das QMQM -Benutzerprofil auf allen drei Servern das gleiche Kennwort hat.
4. Installieren Sie IBM i NetServer (siehe [i5/OS NetServer](#)).

Informationen zu diesem Vorgang

Die in [Abbildung 37](#) auf Seite 317 veranschaulichte Konfiguration wird wie folgt erstellt. Die Warteschlangenmanagerdaten werden unter Verwendung von IBM i NetServer verbunden.

- Erstellen Sie Verbindungen von ALPHA und BETA zum Verzeichnisfreigabe auf GAMMA, in dem die WS-Manager-Daten gespeichert werden sollen. Die Task richtet außerdem die erforderlichen Berechtigungen, Benutzerprofile und Kennwörter ein.
- Fügen Sie RDBE-Einträge (Einträge für relationale Datenbank) zu den IBM i-Systemen hinzu, die Warteschlangenmanagerinstanzen ausführen werden. Über die RDBE-Einträge werden Verbindungen zu den IBM i-Systemen für ferne Journalführung hergestellt.
- Erstellen Sie den Warteschlangenmanager QM1 auf dem IBM i -Server ALPHA.
- Fügen Sie die Steuerinformationen für Warteschlangenmanager QM1 auf dem anderen IBM i-Server (BETA) hinzu.
- Erstellen Sie ferne Journale auf beiden IBM i-Servern für beide Warteschlangenmanagerinstanzen. Jeder WS-Manager schreibt in das lokale Journal. Das lokale Journal wird in das ferne Journal repliziert. Der Befehl **ADDMQMJRN** vereinfacht das Hinzufügen der Journale und der Verbindungen.
- Starten Sie den Warteschlangenmanager, um eine Standby-Instanz zu ermöglichen.

Vorgehensweise

1. Führen Sie den im Abschnitt „[Gemeinsam Nutzung über ein Netz für Warteschlangenmanagerdaten mithilfe von NetServer unter IBM i einrichten](#)“ auf Seite 300 beschriebenen Schritt aus.

Dies hat zur Folge, dass ALPHA und BETA eine gemeinsame Nutzung (/QNTC/GAMMA/WMQ) aufweisen, die auf /QIBM/UserData/mqm/qmgrs auf GAMMA verweist. Die Benutzerprofile QMQM und QMQMADM verfügen über die erforderlichen Berechtigungen, und QMQM verfügt über übereinstimmende Kennwörter auf allen drei Systemen.

2. Fügen Sie RDBE-Einträge (Einträge für relationale Datenbank) zu den IBM i-Systemen hinzu, die Warteschlangenmanagerinstanzen hosten werden.

a) Erstellen Sie auf ALPHA die Verbindung zu BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

b) Erstellen Sie auf BETA die Verbindungen zu ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Erstellen Sie den WS-Manager QM1 auf ALPHA, und speichern Sie die WS-Manager-Daten auf GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ')
```

Der Pfad verwendet NetServer, um die Warteschlangenmanagerdaten in zu erstellen.

4. Auf ALPHA ausführen. Der Befehl fügt ein fernes Journal auf BETA für hinzu.

```
ADDMQJRN MQMNAME(QM1) RMTJNRDB(BETA)
```

erstellt Journaleinträge im lokalen Journal auf ALPHA, wenn sich die aktive Instanz von auf ALPHA befindet. Das lokale Journal auf ALPHA wird in das ferne Journal auf BETA repliziert.

5. Verwenden Sie den Befehl, um die IBM MQ -Konfigurationsdaten zu überprüfen, die von für auf ALPHA erstellt wurden.

Die Informationen werden im nächsten Schritt benötigt.

In diesem Beispiel wird die folgende Konfiguration in ALPHA für erstellt:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. Erstellen Sie mit dem Befehl eine Warteschlangenmanagerinstanz von QM1 auf BETA. Führen Sie den folgenden Befehl auf BETA aus, um die Steuerinformationen des Warteschlangenmanagers auf BETA zu ändern.

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

Tipp: Kopieren Sie die Konfigurationsinformationen und fügen Sie sie ein. Die Zeilengruppe des Warteschlangenmanagers ist auf ALPHA und BETA identisch.

7. Auf BETA ausführen. Der Befehl fügt ein lokales Journal auf BETA und ein fernes Journal auf ALPHA für hinzu.

```
ADDMQJRN MQMNAME(QM1) RMTJNRDB(ALPHA)
```

erstellt Journaleinträge im lokalen Journal auf BETA, wenn sich die aktive Instanz von auf BETA befindet. Das lokale Journal auf BETA wird in das ferne Journal auf ALPHA repliziert.

Anmerkung: Alternativ können Sie die ferne Journalaufzeichnung von BETA auf ALPHA mit asynchronem Journaling einrichten.

Anstatt den Befehl in Schritt „7“ auf Seite 315 auszuführen, können Sie auch mit dem folgenden Befehl die Journalführung von BETA auf ALPHA im asynchronen Modus ausführen:

```
ADDQMQRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNCR)
```

Wenn der Server oder das Journaling auf ALPHA die Ursache des Fehlers ist, wird BETA gestartet, ohne zu warten, dass neue Journaleinträge in ALPHA repliziert werden.

Wechseln Sie mit dem Befehl in den Replikationsmodus *SYNC, wenn ALPHA wieder online ist.

Entscheiden Sie anhand der Informationen in [„Konfiguration mit gespiegeltem Journal für ASP unter IBM i“](#) auf Seite 308, ob die Journalspiegelung im synchronen und/oder asynchronen Modus erfolgen soll. Der Standardwert ist die synchrone Replikation mit einer 60-Sekunden-Wartezeit für eine Antwort vom fernen Journal.

8. Stellen Sie sicher, dass die Journale auf ALPHA und BETA aktiviert sind und der Status der fernen Journalreplikation lautet.

a) Auf ALPHA:

```
WRKMQMQRN MQMNAME(QM1)
```

b) Auf BETA:

```
WRKMQMQRN MQMNAME(QM1)
```

9. Starten Sie die WS-Manager-Instanzen auf ALPHA und BETA.

a) Starten Sie die erste Instanz auf ALPHA, so dass sie die aktive Instanz ist. Aktivieren der Umschaltung auf eine Standby-Instanz.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Starten Sie die zweite Instanz auf BETA, und machen Sie sie zur Standby-Instanz.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Ergebnisse

Verwenden Sie diesen Befehl, um den Status des Warteschlangenmanagers zu überprüfen:

1. Der Status der Warteschlangenmanagerinstanz auf ALPHA sollte lauten.
2. Der Status der Warteschlangenmanagerinstanz auf BETA sollte lauten.

Beispiel

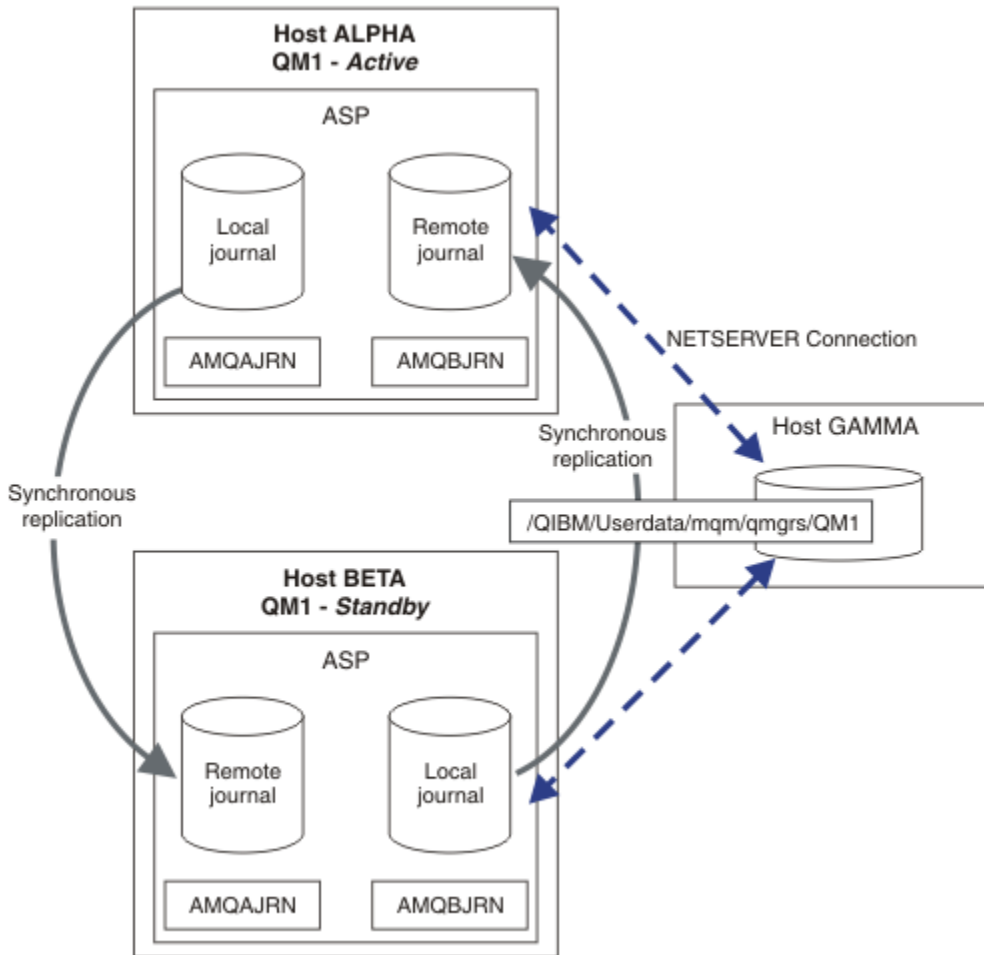


Abbildung 37. Gespiegelte Journalkonfiguration

Nächste Schritte

- Stellen Sie sicher, dass die aktiven Instanzen und die Standby-Instanzen automatisch umgeschaltet werden. Sie können die Musterprogramme mit hoher Verfügbarkeit ausführen, um den Switch zu testen. Weitere Informationen finden Sie im Abschnitt [Musterprogramme für Hochverfügbarkeit](#). Die Beispielprogramme sind 'C'-Clients. Sie können sie von einer Windows- oder UNIX-Plattform aus ausführen.

1. Starten Sie die Beispielprogramme mit hoher Verfügbarkeit.
2. Beenden Sie auf ALPHA den Warteschlangenmanager, der die Umschaltung angefordert hat:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Überprüfen Sie, ob die Instanz von auf BETA aktiv ist.
4. Neustart auf ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Sehen Sie sich alternative Hochverfügbarkeitskonfigurationen an:

1. Verwenden Sie NetServer, um die Warteschlangenmanagerdaten auf einem Windows-Server abzulegen.

2. Statt die ferne Journalfunktion zum Spiegeln des WS-Manager-Journals zu verwenden, speichern Sie das Journal in einem unabhängigen ASP. Verwenden Sie das IBM i-Clustering, um den unabhängigen ASP von ALPHA nach BETA zu übertragen.

IBM i Einzel-Instanz-Warteschlangenmanager mithilfe von NetServer und Journalspiegelung unter IBM i in einen Multi-Instanz-Warteschlangenmanager konvertieren

Konvertieren Sie einen einzelnen Instanzwarteschlangenmanager in einen Multi-Instanz-Warteschlangenmanager. Versetzen Sie die WS-Manager-Daten in eine Netzverbindung, die von NetServer verbunden ist. Spiegeln Sie das Warteschlangenmanagerjournal mithilfe der fernen Journalführung auf einem zweiten IBM i-Server.

Vorbereitende Schritte

1. Für die Aufgabe sind drei IBM i-Server erforderlich. Die vorhandene IBM MQ-Installation, im Beispiel auf dem Server ALPHA, muss mindestens IBM WebSphere MQ 7.0.1 Fix Pack 1 sein. ALPHA führt einen WS-Manager mit dem Namen QM1 in dem Beispiel aus.
2. Installieren Sie IBM MQ auf dem zweiten IBM i-Server, im Beispiel ist das BETA.
3. Der dritte Server ist ein IBM i-Server, der über NetServer mit ALPHA und BETA verbunden ist. Er wird verwendet, um die WS-Manager-Daten gemeinsam zu nutzen. IBM MQ muss dort nicht installiert sein. Es ist jedoch hilfreich, IBM MQ zumindest temporär zu installieren, um die Warteschlangenmanagerverzeichnisse und berechtigungen einzurichten.
4. Stellen Sie sicher, dass das QMQM -Benutzerprofil auf allen drei Servern das gleiche Kennwort hat.
5. Installieren Sie IBM i NetServer (siehe [i5/OS NetServer](#)).

Informationen zu diesem Vorgang

Konvertieren Sie wie folgt einen Warteschlangenmanager mit einer Instanz in den in [Abbildung 38 auf Seite 322](#) dargestellten Warteschlangenmanager mit mehreren Instanzen. Der einzelne Instanz-WS-Manager wird in der Task gelöscht und anschließend erneut erstellt, wobei die WS-Manager-Daten auf dem gemeinsam mit NetServer verbundenen Netzfreigabe gespeichert werden. Diese Prozedur ist zuverlässiger, als die WS-Manager-Verzeichnisse und Dateien mit dem Befehl **CPY** in die gemeinsam genutzte Netzfreigabe zu versetzen.

- Erstellen Sie Verbindungen von ALPHA und BETA zum Verzeichnisfreigabe auf GAMMA, in dem die WS-Manager-Daten gespeichert werden sollen. Die Task richtet außerdem die erforderlichen Berechtigungen, Benutzerprofile und Kennwörter ein.
- Fügen Sie RDBE-Einträge (Einträge für relationale Datenbank) zu den IBM i-Systemen hinzu, die Warteschlangenmanagerinstanzen ausführen werden. Über die RDBE-Einträge werden Verbindungen zu den IBM i-Systemen für ferne Journalführung hergestellt.
- Speichern Sie die WS-Manager-Protokolle und -Definitionen, stoppen Sie den Warteschlangenmanager, und löschen Sie ihn.
- Erstellen Sie den Warteschlangenmanager erneut, und speichern Sie die WS-Manager-Daten in der Netzfreigabe auf GAMMA.
- Fügen Sie die zweite Instanz des Warteschlangenmanagers dem anderen Server hinzu.
- Erstellen Sie ferne Journale auf beiden IBM i-Servern für beide Warteschlangenmanagerinstanzen. Jeder WS-Manager schreibt in das lokale Journal. Das lokale Journal wird in das ferne Journal repliziert. Der Befehl **ADDQMJRN** vereinfacht das Hinzufügen der Journale und der Verbindungen.
- Starten Sie den Warteschlangenmanager, um eine Standby-Instanz zu ermöglichen.

Anmerkung:

In Schritt „4“ auf Seite 319 dieses Vorgangs wird der Warteschlangenmanager QM1 mit einer einzigen Instanz gelöscht. Beim Löschen des Warteschlangenmanagers werden alle persistenten Nachrichten in den Warteschlangen gelöscht. Aus diesem Grund die Verarbeitung aller Nachrichten, die vom Warteschlangenmanager gespeichert werden, vor der Konvertierung des Warteschlangenmanagers vollständig verarbeiten. Können nicht alle Nachrichten verarbeitet werden, müssen Sie vor dem Schritt „4“ auf Seite

319 zunächst die WS-Manager-Bibliothek sichern. Im Anschluss an Schritt „5“ auf Seite 319 wird die WS-Manager-Bibliothek wiederhergestellt.

Anmerkung:

In Schritt „5“ auf Seite 319 der Task erstellen Sie QM1 erneut. Obwohl der WS-Manager denselben Namen hat, hat er eine andere WS-Manager-ID. Das WS-Manager-Clustering verwendet die WS-Manager-ID. Um einen Warteschlangenmanager in einem Cluster zu löschen und erneut zu erstellen, müssen Sie zunächst den Warteschlangenmanager aus dem Cluster entfernen (siehe [Entfernen eines Warteschlangenmanagers aus einem Cluster: Alternative Methode](#) oder [Warteschlangenmanager aus einem Cluster entfernen](#)). Wenn Sie den WS-Manager erneut erstellt haben, fügen Sie ihn dem Cluster hinzu. Obwohl er denselben Namen wie zuvor hat, scheint er ein neuer Warteschlangenmanager zu den anderen WS-Managern im Cluster zu sein.

Vorgehensweise

1. Führen Sie den im Abschnitt „Gemeinsam Nutzung über ein Netz für Warteschlangenmanagerdaten mithilfe von NetServer unter IBM i einrichten“ auf Seite 300 beschriebenen Schritt aus.

Dies hat zur Folge, dass ALPHA und BETA eine gemeinsame Nutzung (/QNTC/GAMMA/WMQ) aufweisen, die auf /QIBM/UserData/mqm/qmgrs auf GAMMA verweist. Die Benutzerprofile QMQM und QMQMADM verfügen über die erforderlichen Berechtigungen, und QMQM verfügt über übereinstimmende Kennwörter auf allen drei Systemen.

2. Fügen Sie RDBE-Einträge (Einträge für relationale Datenbank) zu den IBM i-Systemen hinzu, die Warteschlangenmanagerinstanzen hosten werden.
 - a) Erstellen Sie auf ALPHA die Verbindung zu BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) Erstellen Sie auf BETA die Verbindungen zu ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Erstellen Sie die Scripts, die die WS-Manager-Objekte erneut erstellen.

```
QSAVEQMGR LCLQMGRNAM(QM1) FILENAME('*CURLIB/QMQSC(QM1)')  
OUTPUT(*REPLACE) MAKEAUTH(*YES) AUTHFN('*CURLIB/QMAUT(QM1)')
```

4. Stoppen Sie den WS-Manager und löschen Sie ihn.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*YES) TIMEOUT(15)  
DLTMQM MQMNAME(QM1)
```

5. Erstellen Sie den WS-Manager QM1 auf ALPHA, und speichern Sie die WS-Manager-Daten auf GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP('/QNTC/GAMMA/WMQ')
```

Der Pfad verwendet NetServer, um die Warteschlangenmanagerdaten in zu erstellen.

6. Erstellen Sie die WS-Manager-Objekte für QM1 aus den gespeicherten Definitionen erneut.

```
STRMQMMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. Wenden Sie die Berechtigungen aus den gespeicherten Informationen an.

- a) Kompilieren Sie das gespeicherte Berechtigungsprogramm.

```
CRTCLPGM PGM(*CURLIB/QM1) SRCFILE(*CURLIB/QMAUT)
SRCMBR(QM1) REPLACE(*YES)
```

b) Führen Sie das Programm aus, um die Berechtigungen anzuwenden.

```
CALL PGM(*CURLIB/QM1)
```

c) Aktualisieren Sie die Sicherheitsinformationen für QM1.

```
RFRMQMAUT MQMNAME(QM1)
```

8. Auf ALPHA ausführen. Der Befehl fügt ein fernes Journal auf BETA für hinzu.

```
ADDQMJRNM MQMNAME(QM1) RMTJRNRDB(BETA)
```

erstellt Journaleinträge im lokalen Journal auf ALPHA, wenn sich die aktive Instanz von auf ALPHA befindet. Das lokale Journal auf ALPHA wird in das ferne Journal auf BETA repliziert.

9. Verwenden Sie den Befehl, um die IBM MQ -Konfigurationsdaten zu überprüfen, die von für auf ALPHA erstellt wurden.

Die Informationen werden im nächsten Schritt benötigt.

In diesem Beispiel wird die folgende Konfiguration in ALPHA für erstellt:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. Erstellen Sie mit dem Befehl eine Warteschlangenmanagerinstanz von QM1 auf BETA. Führen Sie den folgenden Befehl auf BETA aus, um die Steuerinformationen des Warteschlangenmanagers auf BETA zu ändern.

```
ADDQMIMF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

Tipp: Kopieren Sie die Konfigurationsinformationen und fügen Sie sie ein. Die Zeilengruppe des Warteschlangenmanagers ist auf ALPHA und BETA identisch.

11. Auf BETA ausführen. Der Befehl fügt ein lokales Journal auf BETA und ein fernes Journal auf ALPHA für hinzu.

```
ADDQMJRNM MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

erstellt Journaleinträge im lokalen Journal auf BETA, wenn sich die aktive Instanz von auf BETA befindet. Das lokale Journal auf BETA wird in das ferne Journal auf ALPHA repliziert.

Anmerkung: Alternativ können Sie die ferne Journalaufzeichnung von BETA auf ALPHA mit asynchronem Journaling einrichten.

Anstatt den Befehl in Schritt „7“ auf Seite 315 auszuführen, können Sie auch mit dem folgenden Befehl die Journalführung von BETA auf ALPHA im asynchronen Modus ausführen:

```
ADDQMJRNM MQMNAME(QM1) RMTJRNRDB(ALPHA) RMTJRNDLV(*ASYNC)
```


Wenn der Server oder das Journaling auf ALPHA die Ursache des Fehlers ist, wird BETA gestartet, ohne zu warten, dass neue Journaleinträge in ALPHA repliziert werden.

Wechseln Sie mit dem Befehl in den Replikationsmodus *SYNC, wenn ALPHA wieder online ist.

Entscheiden Sie anhand der Informationen in „Konfiguration mit gespiegeltem Journal für ASP unter IBM i“ auf Seite 308, ob die Journalspiegelung im synchronen und/oder asynchronen Modus erfolgen soll. Der Standardwert ist die synchrone Replikation mit einer 60-Sekunden-Wartezeit für eine Antwort vom fernen Journal.

12. Stellen Sie sicher, dass die Journale auf ALPHA und BETA aktiviert sind und der Status der fernen Journalreplikation lautet.

a) Auf ALPHA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) Auf BETA:

```
WRKMQMJRN MQMNAME(QM1)
```

13. Starten Sie die WS-Manager-Instanzen auf ALPHA und BETA.

a) Starten Sie die erste Instanz auf ALPHA, so dass sie die aktive Instanz ist. Aktivieren der Umschaltung auf eine Standby-Instanz.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Starten Sie die zweite Instanz auf BETA, und machen Sie sie zur Standby-Instanz.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Ergebnisse

Verwenden Sie diesen Befehl, um den Status des Warteschlangenmanagers zu überprüfen:

1. Der Status der Warteschlangenmanagerinstanz auf ALPHA sollte lauten.
2. Der Status der Warteschlangenmanagerinstanz auf BETA sollte lauten.

Beispiel

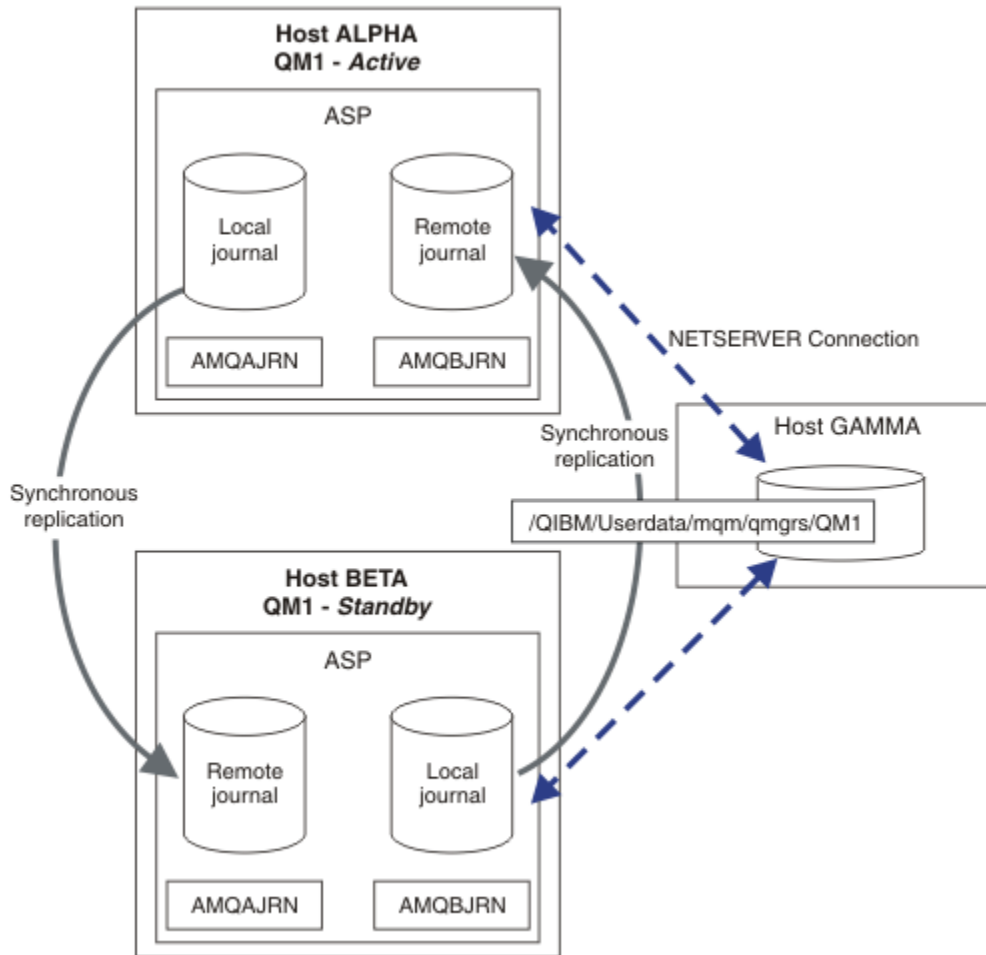


Abbildung 38. Gespiegelte Journalkonfiguration

Nächste Schritte

- Stellen Sie sicher, dass die aktiven Instanzen und die Standby-Instanzen automatisch umgeschaltet werden. Sie können die Musterprogramme mit hoher Verfügbarkeit ausführen, um den Switch zu testen. Weitere Informationen finden Sie im Abschnitt [Musterprogramme für Hochverfügbarkeit](#). Die Beispielprogramme sind 'C' -Clients. Sie können sie von einer Windows- oder UNIX-Plattform aus ausführen.

1. Starten Sie die Beispielprogramme mit hoher Verfügbarkeit.
2. Beenden Sie auf ALPHA den Warteschlangenmanager, der die Umschaltung angefordert hat:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Überprüfen Sie, ob die Instanz von auf BETA aktiv ist.
4. Neustart auf ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Sehen Sie sich alternative Hochverfügbarkeitskonfigurationen an:

1. Verwenden Sie NetServer, um die Warteschlangenmanagerdaten auf einem Windows-Server abzulegen.

2. Statt die ferne Journalfunktion zum Spiegeln des WS-Manager-Journals zu verwenden, speichern Sie das Journal in einem unabhängigen ASP. Verwenden Sie das IBM i-Clustering, um den unabhängigen ASP von ALPHA nach BETA zu übertragen.

IBM i Konfiguration für Wechsel des unabhängigen ASP-Journals unter IBM i

Es ist nicht erforderlich, ein unabhängiges ASP-Journal zu replizieren, um eine Konfiguration mit mehreren Instanzen des Warteschlangenmanagers zu erstellen. Sie müssen ein Mittel automatisieren, um den unabhängigen ASP vom aktiven WS-Manager in den Standby-Warteschlangenmanager zu übertragen. Es gibt alternative Hochverfügbarkeitslösungen, die mit einem unabhängigen ASP möglich sind, und nicht alle, die einen Multi-Instanz-Warteschlangenmanager benötigen.

Wenn Sie einen unabhängigen ASP verwenden, müssen Sie das WS-Manager-Journal nicht spiegeln. Wenn Sie die Clusterverwaltung installiert haben und sich die Server, auf denen sich die Warteschlangenmanagerinstanzen befinden, in derselben Clusterressourcengruppe befinden, kann das WS-Manager-Journal automatisch auf einen anderen Server innerhalb eines kurzen Abstands des aktiven Servers übertragen werden, wenn der Host, auf dem die aktive Instanz ausgeführt wird, ausfällt. Sie können das Journal auch manuell als Teil eines geplanten Schalters übertragen, oder Sie können eine Befehlsprozedur schreiben, um den unabhängigen ASP programmgestützt zu übertragen.

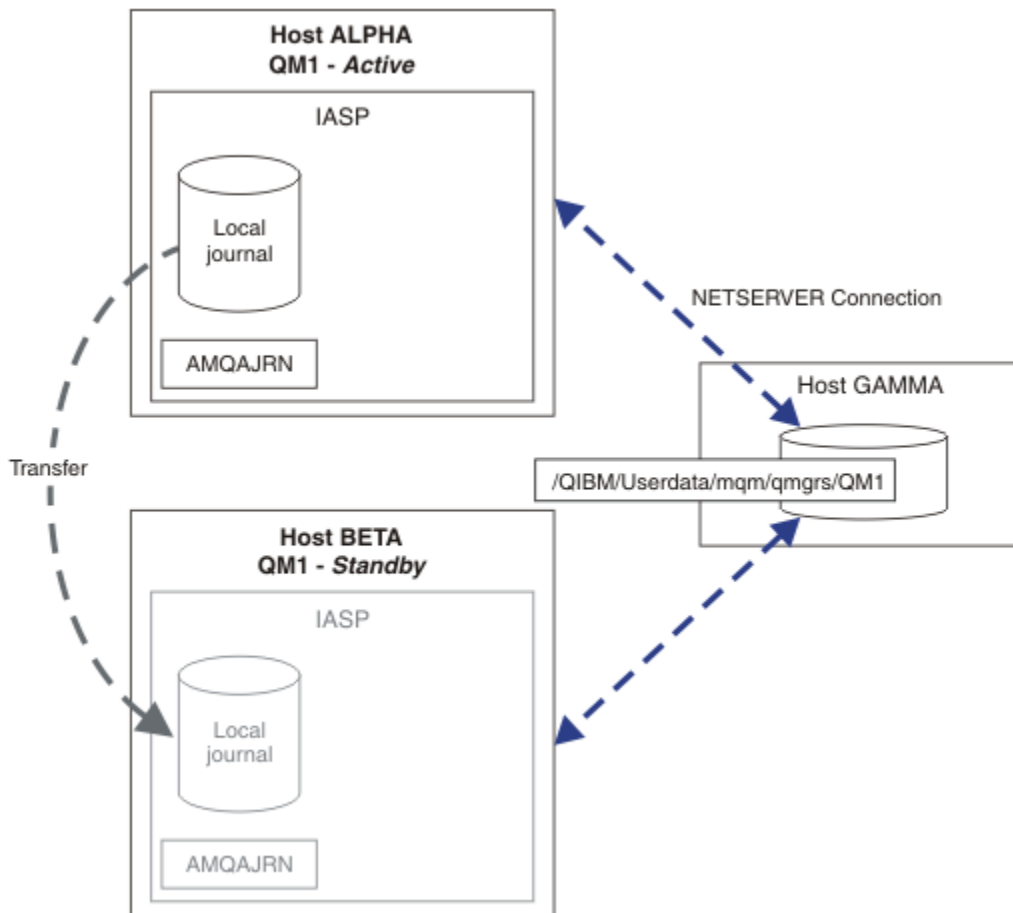


Abbildung 39. WS-Manager-Journal mit einem unabhängigen ASP übertragen

Bei einer WS-Manager-Operation mit mehreren Instanzen müssen die WS-Manager-Daten in einem gemeinsam genutzten Dateisystem gespeichert werden. Das Dateisystem kann auf einer Vielzahl unterschiedlicher Plattformen gehostet werden. Es können keine Warteschlangenmanagerdaten für mehrere Instanzen in einem ASP oder einem unabhängigen ASP gespeichert werden.

Das gemeinsam genutzte Dateisystem führt zwei Rollen in der Konfiguration aus: Dieselben WS-Manager-Daten werden von allen Instanzen des Warteschlangenmanagers gemeinsam genutzt. Das Dateisys-

tem muss über ein robustes Sperrprotokoll verfügen, das sicherstellt, dass nur eine Instanz des Warteschlangenmanagers Zugriff auf die WS-Manager-Daten hat, nachdem sie gestartet wurde. Wenn der Warteschlangenmanager ausfällt oder die Kommunikation mit dem Dateiserver unterbrochen wird, muss das Dateisystem die Sperre für die WS-Manager-Daten freigeben, die von der aktiven Instanz gehalten werden, die nicht mehr mit dem Dateisystem kommunizieren kann. Die Standby-WS-Manager-Instanz kann dann Schreib-/Lesezugriff auf die WS-Manager-Daten erhalten. Für das Dateisystemprotokoll sind eine Reihe von Regeln vorgegeben, damit eine problemlose Verwendung von Warteschlangenmanagern mit mehreren Instanzen möglich ist (siehe [„Komponenten einer Hochverfügbarkeitslösung unter IBM i“](#) auf Seite 299).

Der Sperrmechanismus serialisiert den Befehl zum Starten des Warteschlangenmanagers und steuert, welche Instanz des Warteschlangenmanagers aktiv ist. Sobald ein Warteschlangenmanager aktiv wird, erstellt er seine Warteschlangen aus dem lokalen Journal, das Sie oder den HA-Cluster an den Standby-Server übertragen haben. Wiederinverbindbare Clients, die auf die erneute Verbindung zum selben Warteschlangenmanager warten, werden erneut verbunden, und alle unzustellbaren Transaktionen werden zurückgesetzt. Anwendungen, die so konfiguriert sind, dass sie als WS-Manager-Services gestartet werden, werden gestartet

Sie müssen sicherstellen, dass das lokale Journal aus der fehlgeschlagenen aktiven WS-Manager-Instanz im unabhängigen ASP an den Server übertragen wird, der die neu aktivierte Standby-Warteschlangenmanagerinstanz hostet, entweder durch Konfigurieren des Clusterressourcenmanagers oder durch manuelle Übertragung des unabhängigen ASP. Die Verwendung unabhängiger ASPs schließt die Konfiguration von fernen Journalen und Spiegelungen nicht aus, wenn Sie sich für die Verwendung des unabhängigen ASP für die Sicherung und Wiederherstellung nach einem Katastrophenfall entscheiden, und die Konfiguration des fernen Journals für die Konfiguration mehrerer Instanzen des Warteschlangenmanagers verwenden.

Wenn Sie sich für die Verwendung eines unabhängigen ASP entschieden haben, gibt es alternative hoch verfügbare Konfigurationen, die Sie möglicherweise in Betracht ziehen. Hintergrundinformationen zu diesen Lösungen finden Sie im Abschnitt [„Unabhängige ASPs und hohe Verfügbarkeit“](#) auf Seite 327.

1. Verwenden Sie keine Multi-Instanz-Warteschlangenmanager, sondern installieren und konfigurieren Sie einen Einzel-Instanz-Warteschlangenmanager vollständig in einem unabhängigen ASP, und verwenden Sie IBM i-Hochverfügbarkeitsservices zur Übernahme des Warteschlangenmanagers. Wahrscheinlich müssen Sie die Lösung mit einem Warteschlangenmanager-Monitor erweitern, um festzustellen, ob der Warteschlangenmanager unabhängig vom Server ausgefallen ist. Dies ist die Basis der in *Supportpac MC41: Configuring IBM MQ for iSeries for High Availability* bereitgestellten Lösung.
2. Unabhängige ASPs und standortübergreifende Spiegelung (XSM) verwenden, um den unabhängigen ASP zu spiegeln, anstatt den unabhängigen ASP im lokalen Bus zu wechseln. Dadurch wird der geographische Bereich der unabhängigen ASP-Lösung bis zu der Zeit erweitert, die zum Schreiben von Protokollsätzen über einen langen Abstand möglich ist.

IBM i *Multi-Instanz-Warteschlangenmanager unter Verwendung von unabhängigem ASP und Net-Server unter IBM i erstellen*

Erstellen Sie einen Multi-Instanz-Warteschlangenmanager, der auf zwei IBM i-Servern ausgeführt werden soll. Die Warteschlangenmanagerdaten werden unter Verwendung von NetServer auf einem IBM i-Server gespeichert. Das WS-Manager-Journal wird in einem unabhängigen ASP gespeichert. Verwenden Sie das IBM i-Clustering oder eine manuelle Prozedur, um den unabhängigen ASP, der das Warteschlangenmanagerjournal enthält, auf den anderen IBM i-Server zu übertragen.

Vorbereitende Schritte

1. Für die Aufgabe sind drei IBM i-Server erforderlich. Installieren Sie IBM MQ auf zwei dieser Server; im Beispiel sind dies ALPHA und BETA. IBM MQ muss mindestens Version 7.0.1.1 sein.
2. Der dritte Server ist ein IBM i-Server, der über NetServer mit ALPHA und BETA verbunden ist. Er wird verwendet, um die WS-Manager-Daten gemeinsam zu nutzen. IBM MQ muss dort nicht installiert sein. Es ist jedoch hilfreich, IBM MQ zumindest temporär zu installieren, um die Warteschlangenmanagerverzeichnisse und berechtigungen einzurichten.
3. Stellen Sie sicher, dass das QMQM -Benutzerprofil auf allen drei Servern das gleiche Kennwort hat.

4. Installieren Sie IBM i NetServer (siehe [i5/OS NetServer](#)).
5. Erstellen Sie Prozeduren, um den unabhängigen ASP vom fehlgeschlagenen Warteschlangenmanager in die Bereitschaftsdatenbank zu übertragen, die die Übernahme übernimmt. Hilfreiche Hinweise zur Erstellung dieser Prozeduren finden Sie möglicherweise in *SupportPac MC41: Configuring IBM MQ for iSeries for High Availability*.

Informationen zu diesem Vorgang

Die in [Abbildung 40 auf Seite 326](#) veranschaulichte Konfiguration wird wie folgt erstellt. Die Warteschlangenmanagerdaten werden unter Verwendung von IBM i NetServer verbunden.

- Erstellen Sie Verbindungen von ALPHA und BETA zum Verzeichnisfreigabe auf GAMMA, in dem die WS-Manager-Daten gespeichert werden sollen. Die Task richtet außerdem die erforderlichen Berechtigungen, Benutzerprofile und Kennwörter ein.
- Erstellen Sie den Warteschlangenmanager QM1 auf dem IBM i -Server ALPHA.
- Fügen Sie die Steuerinformationen für Warteschlangenmanager QM1 auf dem anderen IBM i-Server (BETA) hinzu.
- Starten Sie den Warteschlangenmanager, um eine Standby-Instanz zu ermöglichen.

Vorgehensweise

1. Führen Sie den im Abschnitt „Gemeinsam Nutzung über ein Netz für Warteschlangenmanagerdaten mithilfe von NetServer unter IBM i einrichten“ auf Seite 300 beschriebenen Schritt aus.

Dies hat zur Folge, dass ALPHA und BETA eine gemeinsame Nutzung (/QNTC/GAMMA/WMQ) aufweisen, die auf /QIBM/UserData/mqm/qmgrs auf GAMMA verweist. Die Benutzerprofile QMQM und QMQMADM verfügen über die erforderlichen Berechtigungen, und QMQM verfügt über übereinstimmende Kennwörter auf allen drei Systemen.

2. Erstellen Sie den WS-Manager QM1 auf ALPHA, und speichern Sie die WS-Manager-Daten auf GAMMA.

```
CRTMQM MQMNAME(QM1) UDLSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ')
```

Der Pfad verwendet NetServer, um die Warteschlangenmanagerdaten in zu erstellen.

3. Verwenden Sie den Befehl, um die IBM MQ -Konfigurationsdaten zu überprüfen, die von für auf ALPHA erstellt wurden.

Die Informationen werden im nächsten Schritt benötigt.

In diesem Beispiel wird die folgende Konfiguration in ALPHA für erstellt:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMQM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. Erstellen Sie mit dem Befehl eine Warteschlangenmanagerinstanz von QM1 auf BETA. Führen Sie den folgenden Befehl auf BETA aus, um die Steuerinformationen des Warteschlangenmanagers auf BETA zu ändern.

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMQM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

Tipp: Kopieren Sie die Konfigurationsinformationen und fügen Sie sie ein. Die Zeilengruppe des Warteschlangenmanagers ist auf ALPHA und BETA identisch.

5. Starten Sie die WS-Manager-Instanzen auf ALPHA und BETA.

- a) Starten Sie die erste Instanz auf ALPHA, so dass sie die aktive Instanz ist. Aktivieren der Umschaltung auf eine Standby-Instanz.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) Starten Sie die zweite Instanz auf BETA, und machen Sie sie zur Standby-Instanz.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Ergebnisse

Verwenden Sie diesen Befehl, um den Status des Warteschlangenmanagers zu überprüfen:

1. Der Status der Warteschlangenmanagerinstanz auf ALPHA sollte lauten.
2. Der Status der Warteschlangenmanagerinstanz auf BETA sollte lauten.

Beispiel

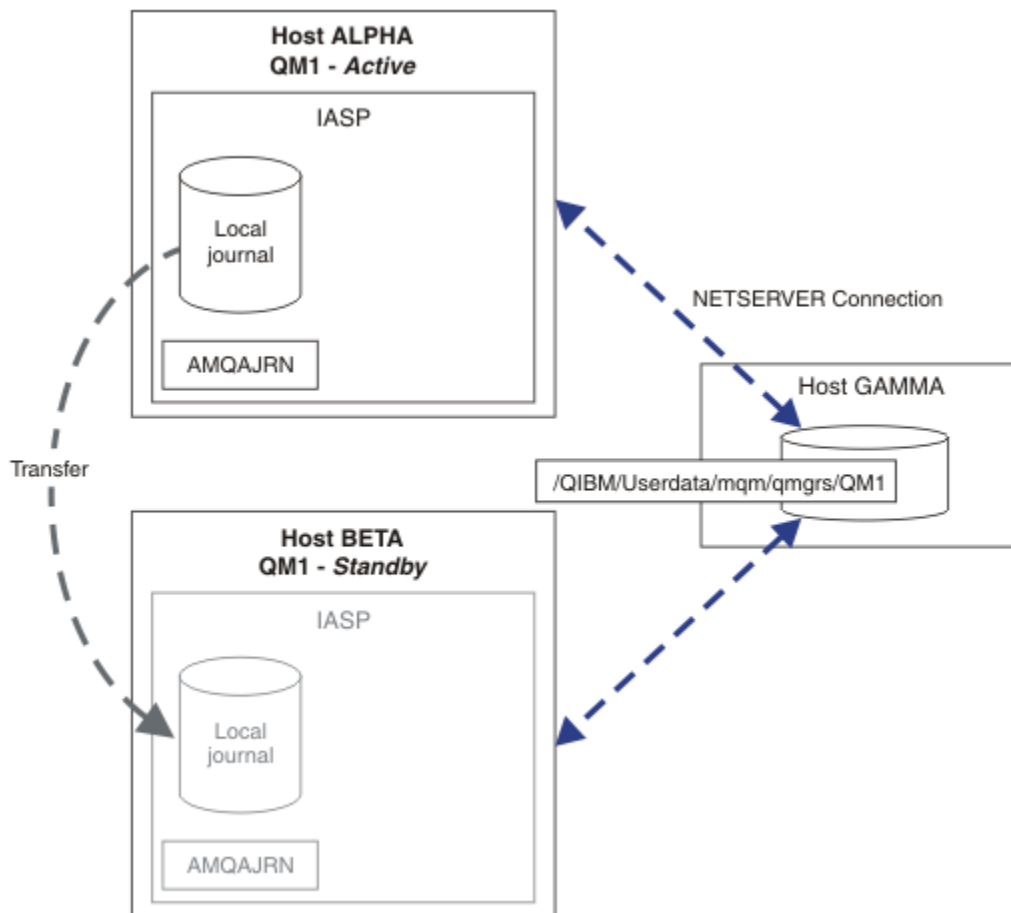


Abbildung 40. WS-Manager-Journal mit einem unabhängigen ASP übertragen

Nächste Schritte

- Stellen Sie sicher, dass die aktiven Instanzen und die Standby-Instanzen automatisch umgeschaltet werden. Sie können die Musterprogramme mit hoher Verfügbarkeit ausführen, um den Switch zu testen.

Weitere Informationen finden Sie im Abschnitt [Musterprogramme für Hochverfügbarkeit](#). Die Beispielprogramme sind 'C'-Clients. Sie können sie von einer Windows- oder UNIX-Plattform aus ausführen.

1. Starten Sie die Beispielprogramme mit hoher Verfügbarkeit.
2. Beenden Sie auf ALPHA den Warteschlangenmanager, der die Umschaltung angefordert hat:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Überprüfen Sie, ob die Instanz von auf BETA aktiv ist.
4. Neustart auf ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Sehen Sie sich alternative Hochverfügbarkeitskonfigurationen an:
 1. Verwenden Sie NetServer, um die Warteschlangenmanagerdaten auf einem IBM i-Server abzulegen.
 2. Anstatt einen unabhängigen ASP zu verwenden, um das WS-Manager-Journal auf den Standby-Server zu übertragen, verwenden Sie das ferne Journaling, um das Journal auf dem Standby-Server zu spiegeln.

IBM i Unabhängige ASPs und hohe Verfügbarkeit

Unabhängige ASPs aktivieren Anwendungen und Daten, die zwischen den Servern verschoben werden. Die Flexibilität unabhängiger ASPs bedeutet, dass sie die Basis für einige IBM i-Hochverfügbarkeitslösungen bilden. Wenn Sie überlegen, ob ein ASP oder ein unabhängiger ASP für das WS-Manager-Journal verwendet werden soll, sollten Sie eine andere Konfiguration mit hoher Verfügbarkeit auf der Basis unabhängiger ASPs berücksichtigen.

Zusatzspeicherpools (Auxiliary Storage Pools; ASPs) sind ein Baustein der IBM i-Architektur. Platteneinheiten werden zu einem einzigen ASP gruppiert. Indem Sie Objekte in verschiedene ASPs stellen, können Sie die Daten in einem ASP vor Plattenausfällen in einem anderen ASP schützen.

Jeder IBM i-Server verfügt über mindestens einen *Basis*-ASP, der auch als System-ASP bezeichnet wird. Sie wird als ASP1 bezeichnet und wird manchmal auch als *SYSBAS bezeichnet. Sie können bis zu 31 zusätzliche *Benutzer*-ASPs konfigurieren, die nicht vom System-ASP aus der Sicht der Anwendung unterschieden werden können, da sie denselben Namensbereich gemeinsam nutzen. Wenn Sie mehrere Basis-ASPs verwenden, um Anwendungen auf viele Platten zu verteilen, können Sie die Leistung verbessern und die Wiederherstellungszeit verkürzen. Die Verwendung mehrerer Basis-ASPs kann auch einen gewissen Grad der Isolation gegen Plattenfehler bieten, aber die Zuverlässigkeit insgesamt wird dadurch nicht verbessert.

Unabhängige ASPs sind eine spezielle Art von ASP. Sie werden häufig als unabhängige Plattenpools bezeichnet. Unabhängige Plattenpools sind eine Schlüsselkomponente der Hochverfügbarkeit von IBM i. Sie können Daten und Anwendungen speichern, die sich unabhängig von dem aktuellen System, mit dem sie verbunden sind, auf unabhängige Plattenspeicher-Speichereinheiten betrachten. Sie können umschaltbare oder nicht umschaltbare unabhängige ASPs konfigurieren. Aus der Verfügbarkeitsperspektive geht es in der Regel nur um umschaltbare unabhängige ASPs, die automatisch vom Server auf den Server übertragen werden können. Als Ergebnis können Sie die Anwendungen und Daten auf dem unabhängigen ASP vom Server auf den Server verschieben.

Im Gegensatz zu Basis-Benutzer-ASPs verwenden unabhängige ASPs nicht denselben Namensbereich wie der System-ASP. Anwendungen, die mit Benutzer-ASPs arbeiten, erfordern Änderungen an der Arbeit mit einem unabhängigen ASP. Sie müssen Ihre Software und die Software anderer Anbieter, die Sie verwenden, in einer unabhängigen ASP-Umgebung überprüfen.

Wenn der unabhängige ASP an einen anderen Server angeschlossen ist, muss der Namensbereich des unabhängigen ASP mit dem Namensbereich des System-ASP kombiniert werden. Dieser Prozess wird als *Variierender-on* des unabhängigen ASP bezeichnet. Sie können einen unabhängigen ASP ohne IPLing des Servers anhängen. Clustering-Unterstützung ist erforderlich, um unabhängige ASPs automatisch von einem Server auf einen anderen zu übertragen.

Zuverlässige Lösungen mit unabhängigen ASPs erstellen

Das Journaling in einem unabhängigen ASP und nicht die Aufzeichnung in einem ASP und die Verwendung der Journalreplikation bietet eine alternative Möglichkeit, den Standby-WS-Manager mit einer Kopie des lokalen Journals aus der fehlgeschlagenen WS-Manager-Instanz zu versorgen. Um den unabhängigen ASP automatisch auf einen anderen Server zu übertragen, müssen Sie die Clustering-Unterstützung installiert und konfiguriert haben. Es gibt eine Reihe von Hochverfügbarkeitslösungen für unabhängige ASPs auf der Basis der Clusterunterstützung und der Plattenspiegelung auf niedriger Ebene, die Sie mit Multi-Instanz-WS-Managern kombinieren können oder diese ersetzen können.

In der folgenden Liste werden die Komponenten beschrieben, die für die Erstellung einer zuverlässigen Lösung auf der Basis unabhängiger ASPs erforderlich sind.

Journaling

Warteschlangenmanager und andere Anwendungen verwenden lokale Journale, um persistente Daten sicher auf die Platte zu schreiben, um vor dem Verlust von Daten im Speicher aufgrund eines Serverfehlers zu schützen. Dies wird manchmal als punktzeitbasierte Konsistenz bezeichnet. Sie gewährleistet nicht die Konsistenz mehrerer Aktualisierungen, die über einen bestimmten Zeitraum stattfinden.

COMMIT-Steuerung

Durch die Verwendung globaler Transaktionen können Sie Aktualisierungen an Nachrichten und Datenbanken koordinieren, sodass die Daten, die in das Journal geschrieben werden, konsistent sind. Sie ermöglicht die Konsistenz über einen bestimmten Zeitraum, indem ein Protokoll für zweiphasiges Commit verwendet wird.

Wähllte Platte

Switched-Disks werden von der Einheiten-Clusterressourcengruppe (CRG) in einem HA-Cluster verwaltet. CRG schaltet bei ungeplanter Betriebsunterbrechung automatisch unabhängige ASPs zu einem neuen Server um. CRGs sind geographisch auf den Umfang des lokalen E/A-Busses beschränkt.

Wenn Sie Ihr lokales Journal in einem umschaltbaren unabhängigen ASP konfigurieren, können Sie das Journal an einen anderen Server übertragen und die Verarbeitung von Nachrichten fortsetzen. Es gehen keine Änderungen an persistenten Nachrichten ohne Synchronisationspunktsteuerung oder bei Synchronisationspunktsteuerung verloren, es sei denn, der unabhängige ASP ist fehlgeschlagen.

Wenn Sie sowohl Journaling als auch COMMIT-Steuerung für umschaltbare unabhängige ASPs verwenden, können Sie Datenbankjournale und WS-Manager-Journale an einen anderen Server übertragen und die Verarbeitung von Transaktionen ohne Verlust von Konsistenztransaktionen oder festgeschriebenen Transaktionen fortsetzen.

Standortübergreifende Spiegelung (XSM-Cross-Site Mirroring)

XSM spiegelt den primären unabhängigen ASP in einem geographisch fernen sekundären unabhängigen ASP über ein TCP/IP-Netz und überträgt die Steuerung bei einem Ausfall automatisch. Sie haben die Wahl, einen synchronen oder asynchronen Spiegel zu konfigurieren. Die synchrone Spiegelung verringert die Leistung des Warteschlangenmanagers, da die Daten gespiegelt werden, bevor die Schreiboperationen auf dem Produktionssystem abgeschlossen sind, aber es garantiert, dass der sekundäre unabhängige ASP auf dem neuesten Stand ist. Wenn Sie die asynchrone Spiegelung verwenden, können Sie nicht garantieren, dass der sekundäre unabhängige ASP auf dem neuesten Stand ist. Bei der asynchronen Spiegelung wird die Konsistenz des sekundären unabhängigen ASP beibehalten.

Es gibt drei XSM-Technologien.

Geographische Spiegelung

Die geographische Spiegelung ist eine Erweiterung des Clustering, die es Ihnen ermöglicht, unabhängige ASPs in einem weiten Bereich zu wechseln. Es hat sowohl synchrone als auch asynchrone Modi. Sie können eine hohe Verfügbarkeit nur im synchronen Modus garantieren, aber die Trennung von unabhängigen ASPs kann die Leistung zu sehr beeinträchtigen. Sie können die geographische Spiegelung mit einer umschaltbaren Platte kombinieren, um eine hohe Verfügbarkeit lokal zu ermöglichen und die Wiederherstellung über Remotezugriff über Remote

Metro-Spiegelung

Bei der Metro-Spiegelung handelt es sich um einen Service auf Einheitenebene, der eine schnelle lokale synchrone Spiegelung über größere Entfernungen ermöglicht als der lokale Bus. Sie können sie mit einem WS-Manager mit mehreren Instanzen kombinieren, um Ihnen eine hohe Verfügbarkeit des Warteschlangenmanagers zu ermöglichen, und zwei Kopien des unabhängigen ASP mit hoher Verfügbarkeit des WS-Manager-Journals zu haben.

Globale Spiegelung

Global Mirroring ist ein Service auf Einheitenebene, der die asynchrone Spiegelung bereitstellt und für die Sicherung und Wiederherstellung über längere Entfernungen geeignet ist, aber keine normale Wahl für hohe Verfügbarkeit ist, da er nur Punkte in der Zeitkonsistenz und nicht in der Währung aufrechterhält.

Die wichtigsten Entscheidungspunkte, die Sie in Betracht ziehen sollten, sind:

ASP oder unabhängigen ASP?

Für die Verwendung von Multi-Instanz-Warteschlangenmanager ist kein IBM i-HA-Cluster erforderlich. Sie können unabhängige ASPs auswählen, wenn Sie bereits unabhängige ASPs verwenden oder wenn Sie Verfügbarkeitsanforderungen für andere Anwendungen haben, die unabhängige ASPs benötigen. Es kann sinnvoll sein, unabhängige ASPs mit Warteschlangenmanagern mit mehreren Instanzen zu kombinieren, um die Warteschlangenmanagerüberwachung als eine Möglichkeit zum Erkennen des Warteschlangenmanagerfehlers zu ersetzen.

Verfügbarkeit?

Was ist das Ziel der Wiederherstellungszeit (RTO)? Wenn Sie die Darstellung von nahezu ununterbrochenem Verhalten benötigen, welche Lösung hat dann die schnellste Wiederanlaufzeit?

Journalverfügbarkeit?

Wie eliminieren Sie das Journal als Single Point of Failure. Sie können eine Hardwarelösung verwenden, indem Sie RAID-1-Einheiten oder besser verwenden, oder Sie können eine Softwarelösung mit Replikatjournalen oder Plattenspiegelung kombinieren oder verwenden.

Abstand?

Wie weit auseinander liegen die aktiven und die Standby-WS-Manager-Instanzen. Können Ihre Benutzer die Leistungseinbußen bei der Replikation synchron über Entfernungen von mehr als etwa 250 Metern tolerieren?

Fachgebiete?

Es ist zu tun, um die Verwaltungstasks zu automatisieren, die die regelmäßige Wartung und die regelmäßige Durchführung der Lösung gewährleisten. Die für die Automatisierung erforderlichen Kenntnisse sind für die Lösungen, die auf ASPs und unabhängigen ASPs basieren, unterschiedlich.



Multi-Instanz-Warteschlangenmanager unter IBM i löschen

Bevor Sie einen WS-Manager mit mehreren Instanzen löschen, stoppen Sie die ferne Journalführung und entfernen Sie die WS-Manager-Instanzen.

Vorbereitende Schritte

1. In diesem Beispiel werden zwei Instanzen des QM1 -Warteschlangenmanagers auf den Servern ALPHA und BETA definiert. ALPHA ist die aktive Instanz, und BETA ist die Standby-Instanz. Die Warteschlangenmanagerdaten, die dem Warteschlangenmanager QM1 zugeordnet sind, werden unter Verwendung von NetServer auf dem IBM i-Server GAMMA gespeichert. Weitere Informationen finden Sie im Abschnitt „Multi-Instanz-Warteschlangenmanager unter Verwendung von Journalspiegelung und NetServer unter IBM i erstellen“ auf Seite 314.
2. ALPHA und BETA müssen verbunden sein, damit alle definierten fernen Journale von IBM MQ gelöscht werden können.
3. Stellen Sie mit dem Systembefehl **EDTF** oder **WRKLNK** sicher, dass auf das Verzeichnis /QNTC und das freigegebene Serververzeichnis zugegriffen werden kann.

Informationen zu diesem Vorgang

Bevor Sie mit dem Befehl **DLTMQM** einen Warteschlangenmanager mit mehreren Instanzen von einem Server löschen, entfernen Sie alle Warteschlangenmanagerinstanzen auf anderen Servern mit dem Befehl **RMVMQMINF**.

Wenn Sie eine Warteschlangenmanagerinstanz mit dem Befehl **RMVMQMINF** entfernen, werden lokale und ferne Journale mit dem Präfix AMQ, die der Instanz zugeordnet sind, gelöscht. Es werden auch Konfigurationsinformationen zur Warteschlangenmanagerinstanz, die auf dem Server lokal sind, gelöscht.

Führen Sie den Befehl **RMVMQMINF** nicht auf dem Server aus, der die verbleibende Instanz des Warteschlangenmanagers enthält. Dadurch wird verhindert, dass **DLTMQM** ordnungsgemäß funktioniert.

Löschen Sie den WS-Manager mit dem Befehl **DLTMQM**. WS-Manager-Daten werden aus dem freigesetzten Netzfreigabe entfernt. Lokale und ferne Journale, die mit AMQ vorfixiert und der Instanz zugeordnet sind, werden gelöscht. **DLTMQM** löscht auch Konfigurationsinformationen zur Warteschlangenmanagerinstanz, die lokal auf dem Server vorhanden sind.

Im Beispiel sind nur zwei WS-Manager-Instanzen vorhanden. IBM MQ unterstützt eine aktive Mehrinstanzkonfiguration, die über eine aktive Warteschlangenmanagerinstanz und eine Standby-Instanz verfügt. Wenn Sie zusätzliche Warteschlangenmanagerinstanzen erstellt haben, die in aktiven Konfigurationen verwendet werden sollen, entfernen Sie sie mit dem Befehl **RMVMQMINF**, bevor Sie die verbleibende Instanz löschen.

Vorgehensweise

1. Führen Sie auf jedem Server den Befehl **CHGMQMJRN RMTJRNSTS (*INACTIVE)** aus, um die ferne Journalführung zwischen den Warteschlangenmanagerinstanzen zu inaktivieren.

a) Auf ALPHA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('BETA') RMTJRNSTS(*INACTIVE)
```

b) Auf BETA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. Führen Sie den Befehl **ENDMQM** auf der aktiven Warteschlangenmanagerinstanz ALPHA aus, um beide Instanzen von QM1 zu stoppen.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

3. Führen Sie den Befehl **RMVMQMINF** auf ALPHA aus, um die Warteschlangenmanagerressourcen für die Instanz von ALPHA und BETA zu entfernen.

```
RMVMQMINF MQMNAME(QM1)
```

RMVMQMINF entfernt die Konfigurationsdaten des Warteschlangenmanagers für QM1 aus ALPHA. Wenn der Journalname von AMQ als Präfix verwendet wird, wird das lokale Journal, das QM1 zugeordnet ist, aus ALPHA gelöscht. Wenn der Journalname von AMQ als Präfix verwendet wird und ein fernes Journal erstellt wurde, entfernt es auch das ferne Journal von BETA.

4. Führen Sie den Befehl **DLTMQM** auf BETA aus, um QM1 zu löschen.

```
DLTMQM MQMNAME(QM1)
```

DLTMQM löscht die Warteschlangenmanagerdaten aus dem gemeinsam genutzten Netzbereich auf GAMMA. Sie entfernt die Konfigurationsdaten des Warteschlangenmanagers für QM1 von BETA. Wenn der Journalname von AMQ als Präfix verwendet wird, wird das lokale Journal, das QM1 von BETA

zugeordnet ist, gelöscht. Wenn der Journalname von AMQ als Präfix verwendet wird und ein fernes Journal erstellt wurde, entfernt es auch das ferne Journal von ALPHA.

Ergebnisse

DLTMQM und **RMVMQMINF** löschen die von **CRTMQM** und **ADDMQJRN** erstellten lokalen und fernen Journale. Mit den Befehlen werden auch die Journalempfänger gelöscht. Die Journale und Journalempfänger müssen der Namenskonvention entsprechen, deren Namen mit AMQ beginnen. **DLTMQM** und **RMVMQMINF** entfernen die Warteschlangenmanagerobjekte, Warteschlangenmanagerdaten und Warteschlangenmanagerkonfigurationsinformationen aus `mq.s.ini`.

Nächste Schritte

Ebenso können Sie nach dem Inaktivieren der Journalführung in Schritt „1“ auf Seite 330 und vor dem Beenden der WS-Manager-Instanzen auch die folgenden Befehle ausführen. Wenn Sie die Namenskonvention nicht eingehalten haben, müssen Sie die Journale und Journalempfänger nach Namen löschen.

1. Auf ALPHA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. Auf BETA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Nachdem Sie die Journale gelöscht haben, fahren Sie mit den restlichen Schritten fort.

Multi-Instanz-Warteschlangenmanager unter IBM i sichern

In der Prozedur wird gezeigt, wie Warteschlangenmanagerobjekte auf dem lokalen Server und die WS-Manager-Daten auf dem Netzdateiserver gesichert werden. Passen Sie das Beispiel an, um Daten für andere WS-Manager zu sichern.

Vorbereitende Schritte

In diesem Beispiel werden die Warteschlangenmanagerdaten, die dem Warteschlangenmanager QM1 zugeordnet sind, mithilfe von NetServer auf dem IBM i-Server mit dem Namen GAMMA gespeichert. Weitere Informationen finden Sie in „Multi-Instanz-Warteschlangenmanager unter Verwendung von Journalspiegelung und NetServer unter IBM i erstellen“ auf Seite 314. IBM MQ ist auf den Servern, ALPHA und BETA, installiert. Der WS-Manager QM1 ist auf ALPHA und BETA konfiguriert.

Informationen zu diesem Vorgang

IBM i unterstützt keine Speicherung von Daten aus einem fernen Verzeichnis. Speichern Sie die WS-Manager-Daten in einem fernen Dateisystem unter Verwendung der Sicherungsprozeduren, die für den Dateisystemserver lokal sind. In diesem Beispiel befindet sich das Netzdateisystem auf dem IBM i-Server GAMMA. Die WS-Manager-Daten werden in einer Sicherungsdatei auf GAMMA gesichert.

Wenn sich das Netzdateisystem auf einem System unter Windows oder Linux befand, können die Warteschlangenmanagerdaten in einer komprimierten Datei gespeichert und dann gesichert werden. Wenn Sie über ein Sicherungssystem verfügen, z. B. Tivoli Storage Manager, verwenden Sie es zum Sichern der WS-Manager-Daten.

Vorgehensweise

1. Erstellen Sie eine Sicherungsdatei auf ALPHA für die WS-Manager-Bibliothek, die QM1 zugeordnet ist. Verwenden Sie den Namen der WS-Manager-Bibliothek, um die Sicherungsdatei zu benennen.

```
CRTSAVF FILE(QGPL/QMQM1)
```

- Speichern Sie die WS-Manager-Bibliothek in der Sicherungsdatei auf ALPHA.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

- Erstellen Sie eine Sicherungsdatei für das WS-Manager-Datenverzeichnis auf GAMMA.

Verwenden Sie den Namen des Warteschlangenmanagers, um die Sicherungsdatei zu benennen.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

- Speichern Sie die Kopie der WS-Manager-Daten aus dem lokalen Verzeichnis auf GAMMA.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

IBM i

Befehle zum Festlegen von Multi-Instanz-Warteschlangenmanagern

IBM MQ stellt Befehle bereit, die die Journalreplikation, das Hinzufügen neuer Warteschlangenmanagerinstanzen und die Konfiguration von Warteschlangenmanagern für die Verwendung unabhängiger ASPs vereinfachen.

Die Journalbefehle zum Erstellen und Verwalten lokaler und ferner Journale sind:

ADDMQMJRN

Mit diesem Befehl können Sie benannte lokale und ferne Journale für eine Warteschlangenmanagerinstanz erstellen und konfigurieren, ob die Replikation synchron oder asynchron ist, was das synchrone Zeitlimit ist, und ob das ferne Journal sofort aktiviert werden soll.

CHGMQMJRN

Der Befehl ändert das Zeitlimit, den Status und die Zustellungsparameter, die sich auf Replikatjournale auswirken.

RMVMQMJRN

Entfernt benannte *ferne* Journale aus einer WS-Manager-Instanz.

WRKMQMJRN

Listet den Status von lokalen und fernen Journalen für eine lokale WS-Manager-Instanz auf.

Fügen Sie zusätzliche WS-Manager-Instanzen hinzu und verwalten Sie sie mit den folgenden Befehlen, die die Datei `mq.s.ini` ändern.

ADDMQMINF

Der Befehl verwendet Informationen, die Sie mit dem Befehl `DSPMQMINF` aus der Datei `mq.s.ini` extrahiert haben, um eine neue Warteschlangenmanagerinstanz auf einem anderen IBM i-Server hinzuzufügen.

RMVMQMINF

Entfernen Sie eine WS-Manager-Instanz. Verwenden Sie diesen Befehl, um eine Instanz eines vorhandenen Warteschlangenmanagers zu entfernen oder um die Konfigurationsdaten für einen Warteschlangenmanager zu entfernen, der aus einem anderen Server gelöscht wurde.

Der Befehl **CRTMQM** verfügt über drei Parameter zur Unterstützung der Konfiguration eines Warteschlangenmanagers mit mehreren Instanzen.

MQMDIRP (*DFT | *directory-prefix*)

Verwenden Sie diesen Parameter, um einen Mountpunkt auszuwählen, der WS-Manager-Daten im Netzspeicher zugeordnet wird.

ASP (*SYSTEM | *ASPDEV | *auxiliary-storage-pool-number*)

Geben Sie `*SYSTEM` oder ein *auxiliary-storage-pool-number* an, um das WS-Manager-Journal auf dem System oder einem Basisbenutzer-ASP zu platzieren. Wählen Sie die Option `*ASPDEV` aus und

legen Sie mit dem Parameter **ASPDEV** einen Einheitenamen fest, um das WS-Manager-Journal in einen unabhängigen ASP zu stellen.

ASPDEV (*ASP | device-name)

Geben Sie eine *device-name* einer primären oder sekundären unabhängigen ASP-Einheit an. Die Auswahl von *ASP hat das gleiche Ergebnis wie die Angabe von **ASP** (*SYSTEM) .

IBM i Hinweise zu Leistung und Plattenübernahme unter IBM i

Verwenden Sie verschiedene Zusatzspeicherpools, um die Leistung und Zuverlässigkeit zu verbessern.

Wenn Sie eine große Anzahl persistenter Nachrichten oder große Nachrichten in Ihren Anwendungen verwenden, wird die Zeit, die Sie beim Schreiben dieser Nachricht auf die Platte verbracht haben, zu einem wichtigen Faktor bei der Leistung des Systems.

Stellen Sie sicher, dass Sie über eine ausreichende Plattenaktivierung verfügen, um diese Möglichkeit zu bewältigen, oder Sie sollten einen separaten Zusatzspeicherpool (ASP) in Betracht ziehen, in dem die Journalempfänger Ihres Warteschlangenmanagers gehalten werden sollen.

Sie können angeben, in welchem ASP Ihre Warteschlangenmanagerbibliothek und Journale gespeichert werden, wenn Sie Ihren Warteschlangenmanager mit dem Parameter ASP von **CRTMQM** erstellen. Standardmäßig werden die WS-Manager-Bibliothek und Journale und IFS-Daten im System-ASP gespeichert.

ASPs ermöglichen die Isolation von Objekten auf einer oder mehreren bestimmten Platteneinheiten. Dadurch kann auch der Datenverlust aufgrund eines Plattenmedienfehlers reduziert werden. In den meisten Fällen gehen nur die Daten verloren, die auf Platteneinheiten in dem betroffenen ASP gespeichert sind.

Es wird empfohlen, die WS-Manager-Bibliothek und die Journaldaten in separaten Benutzer-ASPs auf die des IFS-Stammdatensystems zu speichern, um eine Funktionsübernahme zu ermöglichen und die Plattenkonflikte zu reduzieren.

Weitere Informationen finden Sie im Abschnitt [Sicherheit und Wiederherstellung](#) .

IBM i SAVLIB zum Speichern von IBM MQ-Bibliotheken unter IBM i verwenden

Sie können SAVLIB LIB (*ALLUSR) nicht zum Speichern der IBM MQ -Bibliotheken verwenden, da diese Bibliotheken Namen haben, die mit Q beginnen.

Sie können SAVLIB LIB (QM*) verwenden, um alle WS-Manager-Bibliotheken zu sichern, aber nur dann, wenn Sie eine andere Sicherungseinheit als *SAVF verwenden. Für DEV (*SAVF) müssen Sie für jede WS-Manager-Bibliothek auf Ihrem System einen Befehl SAVLIB verwenden.

IBM i IBM MQ for IBM i in den Quiescemodus versetzen

In diesem Abschnitt wird beschrieben, wie IBM MQ for IBM i in den Quiescemodus (ordnungsgemäße Beendigung) versetzt wird.

So wird IBM MQ for IBM i in den Quiescemodus versetzt:

1. Melden Sie sich bei einer neuen interaktiven IBM MQ for IBM i-Sitzung an und stellen Sie sicher, dass Sie nicht auf Objekte zugreifen.
2. Stellen Sie sicher, dass Folgendes gilt:
 - Berechtigung *ALLOBJ oder Objektverwaltungsberechtigung für die Bibliothek QMQM
 - Genügend die Berechtigung zur Verwendung des Befehls ENDSBS
3. Weisen Sie alle Benutzer darauf hin, dass IBM MQ for IBM i gestoppt wird.
4. Die weiteren Schritte hängen davon ab, ob ein einzelner Warteschlangenmanager heruntergefahren werden soll (wobei noch weitere Warteschlangenmanager vorhanden sein können; siehe [„Einzelnen Warteschlangenmanager für IBM MQ for IBM i herunterfahren“](#) auf Seite 334) oder ob alle Warte-

schlangenmanager heruntergefahren werden sollen (siehe „[Alle Warteschlangenmanager für IBM MQ for IBM i herunterfahren](#)“ auf Seite 335).

ENDMQM, Parameter ENDCCTJOB (*YES)

Der ENDMQM-Parameter ENDCCTJOB (*YES) funktioniert in IBM MQ for IBM i 6.0 anders als in früheren Versionen.

In früheren Versionen, wenn Sie ENDCCTJOB (*YES) angeben, beendet MQ Ihre Anwendungen zwangsweise für Sie.

Wenn Sie unter IBM MQ for IBM i 6.0 oder höher ENDCCTJOB (*YES) angeben, werden Ihre Anwendungen nicht beendet, sondern vom Warteschlangenmanager getrennt.

Wenn Sie ENDCCTJOB (*YES) angeben und Anwendungen, die nicht geschrieben sind, um festzustellen, dass ein Warteschlangenmanager beendet wird, beim nächsten Aufruf eines neuen MQI-Aufrufs ausgegeben wird, kehrt der Aufruf mit einem Fehler MQRC_CONNECTION_BROKEN (2009) zurück.

Als Alternative zur Verwendung von ENDCCTJOB (*YES) den Parameter ENDCCTJOB (*NO) verwenden und die WRKMQM-Option 22 (Mit Jobs arbeiten) verwenden, um alle Anwendungsjobs, die einen Neustart des Warteschlangenmanagers verhindern, manuell zu beenden.

Einzelnen Warteschlangenmanager für IBM MQ for IBM i herunterfahren

Verwenden Sie diese Informationen, um die drei Typen des Systemabschlusses zu verstehen.

In den folgenden Prozeduren verwenden wir einen Beispiel-WS-Manager-Namen QMgr1 und einen Beispiel-Subsystemnamen von SUBX. Ersetzen Sie diese Namen bei Bedarf durch eigene Werte.

Geplanter Systemabschluss

Geplantes Herunterfahren eines Warteschlangenmanagers unter IBM i

1. Führen Sie vor dem Herunterfahren Folgendes aus

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

2. Führen Sie die folgenden Schritte aus, um den Warteschlangenmanager zu beenden:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Wenn QMgr1 nicht beendet wird, sind der Kanal oder die Anwendungen wahrscheinlich ausgelastet.

3. Wenn Sie QMgr1 sofort beenden müssen, führen Sie die folgenden Schritte aus:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Ungeplante Beendigung

1. Führen Sie die folgenden Schritte aus, um den Warteschlangenmanager zu beenden:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Wenn QMgr1 nicht beendet wird, sind der Kanal oder die Anwendungen wahrscheinlich ausgelastet.

2. Führen Sie die folgenden Schritte aus, wenn Sie QMgr1 sofort beenden müssen:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Herunterfahren unter abnormalen Bedingungen

1. Führen Sie die folgenden Schritte aus, um den Warteschlangenmanager zu beenden:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Wenn QMgr1 nicht beendet wird, fahren Sie mit Schritt 3 fort, und stellen Sie Folgendes bereit:

- QMgr1 befindet sich in einem eigenen Subsystem oder
 - Sie können alle WS-Manager beenden, die dasselbe Subsystem wie QMgr1 gemeinsam nutzen. Verwenden Sie die ungeplante Systemabschlussprozedur für alle diese Warteschlangenmanager.
2. Wenn Sie alle Schritte in der Prozedur für alle Warteschlangenmanager ausgeführt haben, die das Subsystem gemeinsam nutzen (SUBX in unseren Beispielen), führen Sie die folgenden Schritte aus:

```
ENDSBS SUBX *IMMED
```

Wenn dieser Befehl nicht ausgeführt werden kann, beenden Sie alle Warteschlangenmanager, indem Sie die ungeplante Prozedur zum Herunterfahren verwenden und ein IPL auf Ihrer Maschine durchführen.

Warnung: Verwenden Sie ENDJOBABN nicht für IBM MQ -Jobs, die aufgrund von ENDJOB oder ENDSBS nicht beendet werden können, es sei denn, Sie sind sofort darauf vorbereitet, ein einleitendes Programmladen auf Ihrer Maschine durchzuführen.

3. Starten Sie das Subsystem, indem Sie Folgendes ausführen:

```
STRSBS SUBX
```

4. Fahren Sie den WS-Manager sofort herunter, indem Sie Folgendes ausführen:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(10)
```

5. Starten Sie den WS-Manager erneut, indem Sie Folgendes ausführen

```
STRMQM MQMNAME(QMgr1)
```

Wenn dies fehlschlägt und Sie:

- Sie haben Ihre Maschine neu gestartet, indem Sie ein IPL durchführen oder
- Sie haben nur einen einzigen Warteschlangenmanager.

Bereinigen Sie den gemeinsam genutzten IBM MQ-Speicher, indem Sie folgende Befehle ausführen:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

bevor Sie Schritt 5 wiederholen.

Wenn der Neustart des Warteschlangenmanagers länger als ein paar Sekunden dauert, fügt IBM MQ dem Jobprotokoll automatisch Statusnachrichten hinzu, in denen der Startfortschritt detailliert beschrieben wird.

Wenn Sie weiterhin Probleme beim Neustart des Warteschlangenmanagers haben, fordern Sie IBM Unterstützung an. Jede weitere Maßnahme, die Sie möglicherweise ergreifen, könnte den Warteschlangenmanager so beschädigen, dass IBM MQ nicht mehr wiederhergestellt werden kann.



Alle Warteschlangenmanager für IBM MQ for IBM i herunterfahren

Verwenden Sie diese Informationen, um die drei Typen des Systemabschlusses zu verstehen.

Die Prozeduren sind fast dieselben wie bei einem einzelnen Warteschlangenmanager, verwenden jedoch anstelle des Namens des Warteschlangenmanagers *ALL anstelle des Namens des Warteschlangenmanagers, und verwenden Sie andernfalls einen Befehl, der wiederholt jeden Warteschlangenmanagernamen verwendet. Während der gesamten Prozeduren verwenden wir einen Beispiel-WS-Manager-Namen QMgr1 und einen Beispiel-Subsystemnamen SUBX. Ersetzen Sie diese durch Ihre eigenen.

Geplanter Systemabschluss

1. Führen Sie eine Stunde vor dem Herunterfahren aus:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

Wiederholen Sie diesen Schritt für jeden Warteschlangenmanager, den Sie herunterfahren möchten.

2. Führen Sie die folgenden Schritte aus, um den Warteschlangenmanager zu beenden:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Wiederholen Sie diese Schritte für jeden Warteschlangenmanager, den Sie herunterfahren möchten. Es können separate Befehle parallel ausgeführt werden.

Wenn ein WS-Manager nicht innerhalb einer angemessenen Zeit (z. B. 10 Minuten) endet, fahren Sie mit Schritt 3 fort.

3. Führen Sie die folgenden Schritte aus, um alle WS-Manager sofort zu beenden:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Ungeplante Beendigung

1. Führen Sie die folgenden Schritte aus, um einen WS-Manager zu beenden:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Wiederholen Sie diese Schritte für jeden Warteschlangenmanager, den Sie herunterfahren möchten. Es können separate Befehle parallel ausgeführt werden.

Wenn Warteschlangenmanager nicht beendet werden, sind der Kanal oder die Anwendungen wahrscheinlich ausgelastet.

2. Führen Sie die folgenden Schritte aus, wenn Sie die Warteschlangenmanager sofort herunterfahren müssen:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Herunterfahren unter abnormalen Bedingungen

1. Führen Sie die folgenden Schritte aus, um die WS-Manager herunterzufahren

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Wiederholen Sie diese Schritte für jeden Warteschlangenmanager, den Sie herunterfahren möchten. Es können separate Befehle parallel ausgeführt werden.

2. Beenden Sie die Subsysteme (SUBX in unseren Beispielen), indem Sie die folgenden Schritte ausführen:

```
ENDSBS SUBX *IMMED
```

Wiederholen Sie diese Schritte für jedes Subsystem, das heruntergefahren werden soll. Es können separate Befehle parallel ausgeführt werden.

Wenn dieser Befehl nicht ausgeführt werden kann, führen Sie ein IPL auf Ihrem System aus.

Warnung: Verwenden Sie ENDJOBABN nicht für Jobs, die als Ergebnis von ENDJOB oder ENDSBS nicht beendet werden, es sei denn, Sie sind bereit, sofort ein IPL auf Ihrem System auszuführen.

3. Starten Sie die Subsysteme, indem Sie Folgendes ausführen

```
STRSBS SUBX
```

Wiederholen Sie diese Schritte für jedes Subsystem, das Sie starten möchten.

4. Fahren Sie die WS-Manager sofort herunter, indem Sie die folgenden Schritte ausführen:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. Starten Sie die WS-Manager erneut, indem Sie die folgenden

```
STRMQM MQMNAME(QMgr1)
```

Wiederholen Sie diesen Schritt für jeden Warteschlangenmanager, den Sie starten möchten.

Wenn der Neustart eines Warteschlangenmanagers länger als ein paar Sekunden dauert, zeigt IBM MQ immer wieder Statusnachrichten zum Fortschritt des Startvorgangs an.

Wenn Sie weiterhin Probleme beim Neustart eines Warteschlangenmanagers haben, fordern Sie IBM Unterstützung an. Jede weitere Maßnahme, die Sie möglicherweise ergreifen, könnte die Warteschlangenmanager so beschädigen, dass MQSeries oder IBM MQ nicht wiederhergestellt werden kann.

IBM MQ for z/OS verwalten

Die Verwaltung von Warteschlangenmanagern und zugeordneten Ressourcen umfasst die Tasks, die Sie häufig ausführen, um diese Ressourcen zu aktivieren und zu verwalten. Wählen Sie die Methode aus, die Sie bevorzugen, um die Warteschlangenmanager und die zugehörigen Ressourcen zu verwalten.

IBM MQ for z/OS kann mithilfe einer Gruppe von Dienstprogrammen und Programmen gesteuert und verwaltet werden, die mit dem Produkt bereitgestellt werden. Sie können IBM MQ for z/OS mit den IBM MQ-Scriptbefehlen (MQSC) oder mit PCF-Befehlen (Programmable Command Format) verwalten. Informationen zur Verwendung von Befehlen für IBM MQ for z/OS finden Sie im Abschnitt [„Befehle für IBM MQ for z/OS ausgeben“](#) auf Seite 338.

IBM MQ for z/OS stellt auch eine Gruppe von Dienstprogrammen bereit, die Sie bei der Systemverwaltung unterstützen können. Informationen zu den verschiedenen Dienstprogrammen und zur Verwendung dieser Programme finden Sie in [„Dienstprogramme von IBM MQ for z/OS“](#) auf Seite 346.

Informationen zur Verwaltung von IBM MQ for z/OS und zu den verschiedenen Verwaltungstasks, die Sie möglicherweise ausführen müssen, finden Sie unter den folgenden Links:

Zugehörige Konzepte

[„IBM MQ verwalten“](#) auf Seite 5

Wenn Sie Ihre IBM MQ-Warteschlangenmanager und die zugehörigen Ressourcen verwalten möchten, wählen Sie Ihre bevorzugte Methode aus einer Reihe von Tasks aus, die Sie zum Aktivieren und Verwalten dieser Ressourcen verwenden können.

Zugehörige Tasks

„Lokale IBM MQ-Objekte verwalten“ auf Seite 145

Sie können lokale IBM MQ-Objekte verwalten, um Anwendungsprogramme zu unterstützen, die die Schnittstelle für Nachrichtenwarteschlangen (Message Queue Interface, MQI) verwenden.

„Ferne IBM MQ-Objekte verwalten“ auf Seite 200

In diesem Abschnitt erfahren Sie, wie Sie IBM MQ-Objekte mithilfe von MQSC-Befehlen auf einem fernen Warteschlangenmanager verwalten und wie ferne Warteschlangenobjekte zum Steuern der Zieladresse von Nachrichten und Antwortnachrichten verwendet werden.

Zugehörige Informationen

[IBM MQ for z/OS - Konzepte](#)

[Planung](#)

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

[Konfiguration](#)

[z/OS konfigurieren](#)

[Programmierbare Befehlsformate-Referenz](#)

[Referenzinformationen zu MQSC](#)

[IBM MQ for z/OS-Dienstprogramme verwenden](#)

z/OS

Befehle für IBM MQ for z/OS ausgeben

Sie können Warteschlangenmanager mit IBM MQ-Scriptbefehlen (MQSC) im Stapel- oder Dialogmodus steuern.

IBM MQ for z/OS unterstützt MQSC-Befehle, die aus folgenden Quellen ausgegeben werden können:

- Der z/OS-Konsole oder einem Äquivalent (z. B. SDSF/TSO).
- Die Initialisierungseingabedatensätze.
- Das angegebene Stapeldienstprogramm CSQUTIL verarbeitet eine Liste der Befehle in einer sequenziellen Datei.
- Eine entsprechend berechtigte Anwendung, indem ein Befehl als Nachricht an die Befehlseingabewarteschlange gesendet wird. Die Anwendung kann eine der folgenden Anwendungen sein:
 - Ein Stapelregionsprogramm
 - Eine CICS-Anwendung
 - Eine IMS-Anwendung
 - Eine TSO-Anwendung
 - Ein Anwendungs- oder Dienstprogramm auf einem anderen IBM MQ-System

In Tabelle 24 auf Seite 342 sind die MQSC-Befehle und die Quellen, aus denen sie ausgegeben werden können, zusammengefasst.

Ein großer Teil der Funktionalität dieser Befehle ist auf einfache Weise in den Operations- und Steueranzeigen von IBM MQ for z/OS verfügbar.

Änderungen, die mit den Befehlen (direkt oder indirekt) an den Ressourcendefinitionen eines Warteschlangenmanagers vorgenommen werden, bleiben bei Neustarts des IBM MQ-Subsystems erhalten.

IBM MQ for z/OS unterstützt auch PCF-Befehle (Programmable Command Format). Diese vereinfachen die Erstellung von Anwendungen für die Verwaltung von IBM MQ. Die MQSC-Befehle befinden sich in lesbarer Textform, während PCF Anwendungen ermöglicht, Anforderungen zu erstellen und die Antworten zu lesen, ohne dass Textzeichenfolgen syntaktisch analysiert werden müssen. Wie die MQSC-Befehle geben Anwendungen PCF-Befehle aus, indem sie sie als Nachrichten an die Befehlseingabewarteschlange senden. Weitere Informationen zur Verwendung von PCF-Befehlen sowie Details zu den Befehlen finden Sie in der Dokumentation zu [Programmierbare Befehlsformate-Referenz](#).

Wenn Sie ein Objekt in IBM MQ for z/OS definieren, können Sie auswählen, ob diese Definition mit anderen Warteschlangenmanagern gemeinsam genutzt werden soll (*globale* Definition) oder ob die Objektdefinition nur von einem einzigen Warteschlangenmanager verwendet werden soll (*private* Definition). Dies wird als Objekt *disposition* bezeichnet.

Globale Definition

Wenn Ihr Warteschlangenmanager zu einer Gruppe mit gemeinsamer Warteschlange gehört, können Sie alle Objektdefinitionen gemeinsam nutzen, die mit den anderen Mitgliedern der Gruppe erstellt wurden. Dies bedeutet, dass Sie ein Objekt nur einmal definieren müssen, wodurch die Gesamtzahl der für das gesamte System erforderlichen Definitionen reduziert wird.

Globale Objektdefinitionen werden in einem *gemeinsam genutzten Repository* (einer gemeinsam genutzten Db2-Datenbank) gespeichert und sind für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verfügbar. Diese Objekte haben die Disposition GROUP.

Private Definition

Wenn Sie eine Objektdefinition erstellen möchten, die nur für einen Warteschlangenmanager erforderlich ist, oder wenn Ihr Warteschlangenmanager kein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, können Sie Objektdefinitionen erstellen, die nicht gemeinsam mit anderen Mitglieder einer Gruppe mit gemeinsamer Warteschlange genutzt werden.

Private Objektdefinitionen werden auf der Seitengruppe null des definierenden Warteschlangenmanagers gehalten. Diese Objekte haben die Disposition QMGR.

Sie können private Definitionen für alle Typen von IBM MQ-Objekten erstellen, außer für CF-Strukturen (d. h. Kanäle, Namenslisten, Prozessdefinitionen, Warteschlangen, Warteschlangenmanager, Speicherklassendefinitionen und Authentifizierungsdatenobjekte), und globale Definitionen für alle Typen von Objekten, außer Warteschlangenmanagern.

IBM MQ kopiert die Definition eines Gruppenobjekts automatisch in die Seitengruppe null jedes Warteschlangenmanagers, der sie verwendet. Sie können die Kopie der Definition temporär ändern, wenn Sie möchten, und IBM MQ ermöglicht es Ihnen, die Kopien der Seitengruppe bei Bedarf aus der Repository-Kopie zu aktualisieren.

IBM MQ versucht immer, die Kopien der Seitengruppe beim Start aus der Repository-Kopie zu aktualisieren (bei Kanalbefehlen erfolgt dies beim Neustart des Kanalinitiators) oder wenn das Gruppenobjekt geändert wird.

Anmerkung: Die Kopie der Definition wird aus der Definition der Gruppe aktualisiert, nur wenn die Definition der Gruppe geändert wurde, nachdem Sie die Kopie der Definition erstellt haben.

Dadurch wird sichergestellt, dass die Kopien der Seitengruppe die Version im Repository widerspiegeln, einschließlich aller Änderungen, die beim Inaktiven des Warteschlangenmanagers vorgenommen wurden. Die Kopien werden aktualisiert, indem die Befehle DEFINE REPLACE generiert werden. Daher gibt es Situationen, unter denen die Aktualisierung nicht ausgeführt wird, z. B.:

- Wenn eine Kopie der Warteschlange geöffnet ist, schlägt eine Aktualisierung fehl, die die Verwendung der Warteschlange ändert.
- Wenn eine Kopie einer Warteschlange Nachrichten enthält, schlägt eine Aktualisierung fehl, die diese Warteschlange löscht.
- Wenn eine Kopie einer Warteschlange ALTER mit FORCE erforderlich macht, um sie zu ändern.

Unter diesen Umständen wird die Aktualisierung nicht für diese Kopie ausgeführt, sondern wird in den Kopien auf allen anderen Warteschlangenmanagern ausgeführt.

Wenn der WS-Manager heruntergefahren und anschließend wieder neu gestartet wird, werden alle lokalen Kopien von Objekten gelöscht, es sei denn, die Warteschlange enthält beispielsweise Nachrichten.

Es gibt eine Disposition eines dritten Objekts, die nur für lokale Warteschlangen gilt. Auf diese Weise können Sie gemeinsam genutzte Warteschlangen erstellen. Die Definition für eine gemeinsam genutzte

Warteschlange wird im gemeinsam genutzten Repository gespeichert und ist für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verfügbar. Außerdem sind die Nachrichten in einer gemeinsam genutzten Warteschlange ebenfalls für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verfügbar. Dies wird im Abschnitt Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange beschrieben. Gemeinsam genutzte Warteschlangen haben die Objektdisposition SHARED.

In der folgenden Tabelle sind die Auswirkungen der Objektdispositionsoptionen für Warteschlangenmanager, die eigenständig gestartet wurden, und für Warteschlangenmanager als Mitglied einer Gruppe mit gemeinsamer Warteschlange zusammengefasst.

Disposition	Eigenständiger Warteschlangenmanager	Mitglied einer Gruppe mit gemeinsamer Warteschlange
QMGR	Objektdefinition auf Seitengruppe 0 gehalten.	Objektdefinition auf Seitengruppe 0 gehalten.
GRUPPE	Nicht zulässig.	Objektdefinition, die im gemeinsam genutzten Repository enthalten ist. Lokale Kopie auf Seite gesetzt null jedes Warteschlangenmanagers in der Gruppe.
SHARED	Nicht zulässig.	Warteschlangendefinition im gemeinsam genutzten Repository. Nachrichten, die jedem WS-Manager in der Gruppe zur Verfügung stehen.

Globale Definitionen bearbeiten

Wenn Sie die Definition eines Objekts ändern möchten, das im gemeinsam genutzten Repository enthalten ist, müssen Sie angeben, ob die Version im Repository geändert werden soll, oder ob die lokale Kopie auf Seite null gesetzt werden soll. Verwenden Sie die Objektdisposition als Teil des Befehls, um dies zu tun.

Befehle an verschiedene Warteschlangenmanager unter z/OS richten

Sie können den *Befehlsumfang* verwenden, um zu steuern, auf welchem Warteschlangenmanager der Befehl ausgeführt wird.

Sie können einen Befehl auf dem Warteschlangenmanager ausführen, auf dem er eingegeben wurde, oder auf einem anderen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange. Sie können auch einen bestimmten Befehl parallel auf allen Warteschlangenmanagern in einer Gruppe mit gemeinsamer Warteschlange ausgeben. Dies ist sowohl für MQSC-Befehle als auch für PCF-Befehle möglich.

Dies wird durch den *Befehlsumfang* bestimmt. Der Befehlsbereich wird mit der Objektdisposition verwendet, um festzustellen, mit welcher Version eines Objekts gearbeitet werden soll.

Sie können z. B. einige der Attribute eines Objekts ändern, dessen Definition im gemeinsamen Repository enthalten ist.

- Sie können die Version nur auf einem WS-Manager ändern und keine Änderungen an der Version im Repository vornehmen oder die Version von anderen Warteschlangenmanagern verwenden.
- Möglicherweise möchten Sie die Version im gemeinsam genutzten Repository für zukünftige Benutzer ändern, aber vorhandene Kopien unverändert lassen.
- Sie können die Version im gemeinsam genutzten Repository ändern und Ihre Änderungen sofort in allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange wiedergeben, in denen eine Kopie des Objekts auf der Seitengruppe null enthalten ist.

Geben Sie im Befehlsbereich an, ob der Befehl auf diesem Warteschlangenmanager, einem anderen Warteschlangenmanager oder allen Warteschlangenmanagern ausgeführt wird. Mit der Disposition des Objekts können Sie angeben, ob sich das Objekt, das Sie bearbeiten, im gemeinsamen Repository (einem Gruppenobjekt) befindet oder eine lokale Kopie auf der Seitengruppe Null ist (ein WS-Manager-Objekt).

Für die Arbeit mit einer gemeinsam genutzten Warteschlange müssen Sie den Befehlsbereich und die Objektdisposition nicht angeben, da jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange die gemeinsam genutzte Warteschlange als einzelne Warteschlange verarbeitet.

Befehlszusammenfassung für IBM MQ for z/OS

Verwenden Sie dieses Thema als Referenz auf die Haupt-MQSC- und PCF-Befehle.

Tabelle 23 auf Seite 341 enthält eine Zusammenfassung der MQSC- und PCF-Befehle, die in IBM MQ for z/OS zum Ändern, Definieren, Löschen und Anzeigen von IBM MQ-Objekten verfügbar sind.

Tabelle 23. Zusammenfassung der wichtigsten MQSC- und PCF-Befehle nach Objekttyp

MQSC-Befehl	ALTER	DEFINIER	ANZEIGEN	DELETE
PCF-Befehl	Ändern	Erstellen/Kopieren	Inquire	Löschen
AUTHINFO	X	X	X	X
CFSTATUS			X	
CFSTRUCT	X	X	X	X
CHANNEL	X	X	X	X
CHSTATUS			X	
NAMELIST	X	X	X	X
PROCESS	X	X	X	X
QALIAS	M	M	M	M
QCLUSTER			M	
QLOCAL	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
WARTESCHLANGENGE	P	P	X	P
QSTATUS			X	
STGKLASSE	X	X	X	X

Schlüssel für Tabellensymbole:

- M = nur MQSC
- P = nur PCF
- X = beide

Es gibt viele weitere MQSC- und PCF-Befehle, mit denen Sie andere IBM MQ-Ressourcen verwalten und andere Aktionen zusätzlich zu den in [Tabelle 23 auf Seite 341](#) zusammengefassten Aktionen ausführen können.

Tabelle 24 auf Seite 342 zeigt jeden einzelnen MQSC-Befehl und wo er ausgegeben werden kann:

- CSQINP1 Initialisierungseingabedatei
- CSQINP2 Initialisierungseingabedatei
- z/OS-Konsole (oder Äquivalent)
- SYSTEM.COMMAND.INPUT-Warteschlange und Befehlsserver (aus Anwendungen, CSQUTIL oder Initialisierungseingabedatei CSQINPX)

<i>Tabelle 24. Quellen, von denen MQSC-Befehle ausgeführt werden sollen</i>				
Befehl	CSQINP1	CSQINP2	z/OS-Konsole	Befehlseingabewarteschlange und Server
ALTER AUTHINFO		X	X	X
ALTER BUFFPOOL		X	X	X
ALTER CFSTRUCT		X	X	X
ALTER CHANNEL		X	X	X
ALTER NAMELIST		X	X	X
ALTER PSID			X	X
ALTER PROCESS		X	X	X
ALTER QALIAS		X	X	X
ALTER QLOCAL		X	X	X
ALTER QMGR		X	X	X
ALTER QMODEL		X	X	X
ALTER QREMOTE		X	X	X
ALTER SECURITY	X	X	X	X
ALTER STGCLASS		X	X	X
ALTER SUB		X	X	X
ALTER TOPIC		X	X	X
ALTER TRACE	X	X	X	X
ARCHIVE LOG	X	X	X	X
BACKUP CFSTRUCT			X	X
CLEAR QLOCAL		X	X	X
AUTHINFO DEFINIER		X	X	X
DEFINE BUFFPOOL	X	X		
CFSTRUCT DEFINE CFSTRU		X	X	X
CHANNEL DEFINE CHANNEL		X	X	X
PROTOKOLL			X	X
NAMELIST DEFINI		X	X	X
PROZESS DEFI		X	X	X

Tabelle 24. Quellen, von denen MQSC-Befehle ausgeführt werden sollen (Forts.)

Befehl	CSQINP1	CSQINP2	z/OS-Konsole	Befehlseingabe- warteschlange und Server
DEFINE PSID	X		X	X
DEFINE QALIAS		X	X	X
QLOCAL DEFINIER		X	X	X
QMODEL DEFINIER		X	X	X
QREMOTE DEFINIER		X	X	X
STGKLASSE DEFINI		X	X	X
SUB DEFINI			X	X
TOPIC DEFINI		X	X	X
DELETE AUTHINFO		X	X	X
DELETE BUFFPOOL			X	X
DELETE CFSTRUCT		X	X	X
DELETE CHANNEL			X	X
DELETE NAMELIST		X	X	X
DELETE PROCESS		X	X	X
DELETE PSID			X	X
DELETE QALIAS		X	X	X
DELETE QLOCAL		X	X	X
DELETE QMODEL		X	X	X
DELETE QREMOTE		X	X	X
DELETE STGCLASS		X	X	X
DELETE SUB		X	X	X
DELETE TOPIC		X	X	X
ANZEIGEN ARCHIVE	X	X	X	X
DISPLAY AUTHINFO		X	X	X
DISPLAY CFSTATUS			X	X
ANZEIGEN CFSTRUCT		X	X	X
ANZEIGEN CHANNEL		X	X	X
ANZEIGEN CHSTATUS			X	X
DISPLAY CLUSQMGR			X	X
ANZEIGEN CMDSERV	X	X	X	X
DISPLAY CONN		X	X	X
ANZEIGEN CHINIT		X	X	X

Tabelle 24. Quellen, von denen MQSC-Befehle ausgeführt werden sollen (Forts.)

Befehl	CSQINP1	CSQINP2	z/OS-Konsole	Befehlseingabewarteschlange und Server
ANZEIGEN G		X	X	X
ANZEIGEN LOG	X	X	X	X
ANZEIGEN NAMELIST		X	X	X
ANZEIGEN PROZ		X	X	X
DISPLAY QALIAS		X	X	X
ANZEIGEN QCLUSTER		X	X	X
ANZEIGEN QLOCAL		X	X	X
ANZEIGEN QMGR		X	X	X
ANZEIGEN QMODEL		X	X	X
ANZEIGEN QREMOTE		X	X	X
ANZEIGEN QSTATUS		X	X	X
ANZEIGEN QUEUE		X	X	X
DISPLAY SECURITY			X	X
ANZEIGEN STGKLASSE		X	X	X
ANZEIGEN SUB		X	X	X
ANZEIGEN TOPIC		X	X	X
ANZEIGEN SYSTEM	X	X	X	X
ANZEIGEN THREAD		X	X	X
ANZEIGEN TRACE	X	X	X	X
ANZEIGEN SYNTA		X	X	X
MOVE QLOCAL		X	X	X
PING CHANNEL			X	X
RECOVER BSDS	X	X	X	X
RECOVER CFSTRUCT			X	X
REFRESH CLUSTER		X	X	X
REFRESH QMGR		X	X	X
REFRESH SECURITY		X	X	X
RESET CHANNEL			X	X
RESET CLUSTER		X	X	X
RESET QSTATS		X	X	X
RESET TPIPE			X	X
GELÖST-CHANNEL			X	X

Tabelle 24. Quellen, von denen MQSC-Befehle ausgeführt werden sollen (Forts.)

Befehl	CSQINP1	CSQINP2	z/OS-Konsole	Befehlseingabewarteschlange und Server
INDOUBT GELÖST		X	X	X
RESUME QMGR			X	X
SICHERHEIT ÜBERPRÜFEN		X	X	X
SET ARCHIVE	X	X	X	X
SET LOG	X	X	X	X
SET SYSTEM	X	X	X	X
START CHANNEL			X	X
START CHINIT		X	X	X
START CMDSERV	X	X	X	
START LISTENER			X	X
START QMGR			X	
START TRACE	X	X	X	X
STOP CHANNEL			X	X
STOP CHINIT			X	X
STOP CMDSERV	X	X	X	
STOP LISTENER			X	X
STOP QMGR			X	X
STOP-TRACE	X	X	X	X
SUSPEND QMGR			X	X

In MQSC-Befehlen gibt jede Befehlsbeschreibung die Quellen an, von denen dieser Befehl ausgeführt werden kann.

Initialisierungsbefehle für IBM MQ for z/OS

Initialisierungsbefehle können verwendet werden, um den Start des Warteschlangenmanagers zu steuern.

Befehle in den Initialisierungseingabedateien werden verarbeitet, wenn IBM MQ beim Start eines Warteschlangenmanagers initialisiert wird. Es können drei Befehlstypen aus den Initialisierungseingabedaten Gruppen ausgegeben werden:

- Befehle zum Definieren von IBM MQ-Entitäten, die nicht an anderer Stelle definiert werden können, z. B. DEFINE BUFFPOOL.

Diese Befehle müssen sich in der Datei befinden, die durch den DD-Namen CSQINP1 identifiziert wird. Sie werden vor der Neustartphase der Initialisierung verarbeitet. Sie können nicht über die Konsole, die Operations- und Steuerkonsolen oder ein Anwendungsprogramm ausgegeben werden. Die Antworten auf diese Befehle werden in die sequenzielle Datei geschrieben, auf die in der Anweisung CSQOUT1 der gestarteten Taskprozedur verwiesen wird.

- Befehle zum Definieren von IBM MQ-Objekten, die nach einem Neustart wiederherstellbar sind. Diese Definitionen müssen in der Datei angegeben werden, die durch den DD-Namen CSQINP2 angegeben wird. Sie werden in der Seitengruppe Null gespeichert. CSQINP2 wird nach der Neustartphase der Initialisierung verarbeitet. Die Antworten auf diese Befehle werden in die sequenzielle Datei geschrieben, auf die in der Anweisung CSQOUT2 der gestarteten Taskprozedur verwiesen wird.
- Befehle zum Bearbeiten von IBM MQ-Objekten. Diese Befehle müssen auch in der Datei angegeben werden, die durch den DD-Namen CSQINP2 angegeben wird. Das von IBM MQ bereitgestellte Beispiel enthält beispielsweise einen Befehl ALTER QMGR, der eine Warteschlange für nicht zustellbare Nachrichten für das Subsystem angibt. Die Antwort auf diese Befehle wird in die Ausgabedatei 'CSQOUT2' geschrieben.

Anmerkung: Wenn IBM MQ-Objekte in CSQINP2 definiert sind, versucht IBM MQ, sie bei jedem Start des Warteschlangenmanagers neu zu definieren. Wenn die Objekte bereits vorhanden sind, schlägt der Versuch fehl, die Objekte zu definieren. Wenn Sie Ihre Objekte in CSQINP2 definieren müssen, können Sie dieses Problem vermeiden, indem Sie den Parameter REPLACE der DEFINE-Befehle verwenden. Dies überschreibt jedoch alle Änderungen, die während der vorherigen Ausführung des Warteschlangenmanagers vorgenommen wurden.

Es werden Beispielinitialisierungsdateieinträge mit IBM MQ for z/OS bereitgestellt. Sie werden im Abschnitt [Mit IBM MQ gelieferte Beispieldefinitionen](#) beschrieben.

Initialisierungsbefehle für verteilte Steuerung von Warteschlangen

Sie können auch die Initialisierungsdatei CSQINP2 für den Befehl START CHINIT verwenden. Wenn Sie eine Reihe weiterer Befehle zum Definieren Ihrer Umgebung mit verteilter Steuerung von Warteschlangen benötigen (z. B. zum Starten von Listenern), stellt IBM MQ eine dritte Initialisierungseingabedatei mit dem Namen CSQINPX bereit, die als Teil der Prozedur der gestarteten Task für den Kanalinitiator verarbeitet wird.

Die in der Datei enthaltenen MQSC-Befehle werden am Ende der Kanalinitiatorinitialisierung ausgeführt, und die Ausgabe wird in die Datei geschrieben, die durch die DD-Anweisung CSQOUTX angegeben wird. Sie können beispielsweise die Initialisierungsdatei CSQINPX verwenden, um Empfangsprogramme zum Beispiel zu starten.

Es wird ein Beispielinitialisierungsdateieintrag für einen Kanalinitiator mit IBM MQ for z/OS bereitgestellt. Er wird im Abschnitt [Mit IBM MQ gelieferte Beispieldefinitionen](#) beschrieben.

Initialisierungsbefehle für Publish/Subscribe

Wenn Sie eine Reihe von Befehlen zum Definieren Ihrer Publish/Subscribe-Umgebung benötigen (z. B. bei der Definition von Subskriptionen), stellt IBM MQ eine vierte Initialisierungseingabedatei mit dem Namen CSQINPT bereit.

Die in der Datei enthaltenen MQSC-Befehle werden am Ende der Publish/Subscribe-Initialisierung ausgeführt, und die Ausgabe wird in die Datei geschrieben, die durch die DD-Anweisung CSQOUTT angegeben wird. Sie können z. B. die Initialisierungsdatei CSQINPT verwenden, um Subskriptionen zu definieren.

Es wird ein Beispielinitialisierungsdateieintrag für Publish/Subscribe mit IBM MQ for z/OS bereitgestellt. Er wird im Abschnitt [Mit IBM MQ gelieferte Beispieldefinitionen](#) beschrieben.

Dienstprogramme von IBM MQ for z/OS

IBM MQ for z/OS stellt eine Reihe von Dienstprogrammen bereit, die Sie zur Unterstützung bei der Systemverwaltung verwenden können.

IBM MQ for z/OS stellt eine Reihe von Dienstprogrammen bereit, die Sie bei der Ausführung verschiedener Verwaltungstasks unterstützen, z. B. bei folgenden Aufgaben:

- Sicherheitsrichtlinien für Nachrichten verwalten.
- Durchführen von Sicherungs-, Wiederherstellungs- und Reorganisationstasks.

- Geben Sie Befehle und Prozessobjektdefinitionen aus.
- Datenkonvertierungsexits generieren.
- Ändern Sie die Bootstrap-Datei.
- Listet Informationen zu den Protokollen auf.
- Drucken Sie die Protokolle.
- Einrichten von Db2-Tabellen und anderen Db2-Dienstprogrammen.
- Verarbeiten Sie Nachrichten in der Warteschlange für dead-letter.

Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie

Das Dienstprogramm für Nachrichtensicherheitsrichtlinien (CSQOUTIL) wird als eigenständiges Dienstprogramm zur Verwaltung von Nachrichtensicherheitsrichtlinien ausgeführt. Weitere Informationen finden Sie im Abschnitt [Das Nachrichtensicherheitsrichtliniendienstprogramm \(CSQOUTIL\)](#).

Das Dienstprogramm CSQUTIL

Hierbei handelt es sich um ein Dienstprogramm, das bereitgestellt wird, um Sie bei der Sicherung, Wiederherstellung und Reorganisation von Tasks zu unterstützen. Weitere Informationen finden Sie im Abschnitt [Dienstprogramm CSQUTIL](#).

Dienstprogramm für Datenkonvertierungsexit

Das IBM MQ for z/OS-Dienstprogramm für Datenkonvertierungsexits (**CSQUCVX**) wird als Standalone-Dienstprogramm ausgeführt, um Datenkonvertierungsexitrouninen zu erstellen.

Das Dienstprogramm zum Ändern des Protokollbestands

Das IBM MQ for z/OS-Dienstprogramm zum Ändern des Protokollbestands (**CSQJU003**) wird als Standalone-Dienstprogramm ausgeführt, um die Bootstrap-Datei zu ändern. Sie können das Dienstprogramm verwenden, um die folgenden Funktionen auszuführen:

- Aktive oder Archivierungsprotokolldatensätze hinzufügen oder löschen.
- Geben Sie Kennwörter für Archivprotokolle an.

Das Dienstprogramm für die Druckprotokollzuordnung

Das Dienstprogramm für die IBM MQ for z/OS -Druckprotokollzuordnung (**CSQJU004**) wird als eigenständiges Dienstprogramm ausgeführt, um die folgenden Informationen aufzulisten:

- Protokollname und Protokoll-RBA-Zuordnung für beide Kopien aller aktiven Archivierungs- und Archivierungsprotokolldateien. Wenn die doppelte Protokollierung nicht aktiv ist, gibt es nur eine Kopie der Datensätze.
- Aktive Protokolldateien sind für neue Protokolldaten verfügbar.
- Inhalt der Warteschlange von Prüfpunktdateisätzen in der Bootstrap-Datei (BSDS).
- Inhalt des archivierungsprotokolls des Befehlsprotokolls.
- System- und Dienstprogrammzeitmarken.

Das Protokolldruckdienstprogramm

Protokolldruckdienstprogramm (**CSQ1LOGP**) wird als eigenständiges Dienstprogramm ausgeführt. Sie können das Dienstprogramm ausführen, indem Sie Folgendes angeben:

- Bootstrap-Dateigruppe (BSDS)

- Aktive Protokolle (ohne BSDS)
- Archivprotokolle (ohne BSDS)

Dienstprogramm für die Gruppe mit gemeinsamer Warteschlange

Das Dienstprogramm für Gruppen mit gemeinsamer Warteschlange (**CSQ5PQSG**) wird als eigenständiges Dienstprogramm ausgeführt, um Db2-Tabellen einzurichten und weitere Db2-Tasks auszuführen, die für Gruppen mit gemeinsamer Warteschlange erforderlich sind.

Das Dienstprogramm zum Vorformatieren des aktiven Protokolls

Dienstprogramm zur Vorformatierung aktiver Protokolldateien (**CSQJUFMT**) formatiert aktive Protokolldateien, bevor sie von dem Warteschlangenmanager verwendet werden. Wenn die aktiven Protokolldateien vom Dienstprogramm vorformatiert sind, wird die Protokollschreibleistung auf dem ersten Durchlauf des Warteschlangenmanagers durch die aktiven Protokolle verbessert.

Dienstprogramm 'dead-letter queue handler'

Handlerdienstprogramm für nicht zustellbare Nachrichten (**CSQUDLQH**) wird als eigenständiges Dienstprogramm ausgeführt. Sie prüft Nachrichten, die sich in der Warteschlange für den dead-letter befinden, und verarbeitet sie gemäß einer Reihe von Regeln, die Sie dem Dienstprogramm zur Vorspeisungsfunktion zur Verwendung geben.

Dienstprogramm CSQUTIL für IBM MQ for z/OS

Das Dienstprogramm CSQUTIL wird mit IBM MQ for z/OS bereitgestellt, um Sie bei der Ausführung von Sicherheits-, Wiederherstellungs- und Reorganisationsaufgaben sowie bei der Ausgabe von Befehlen und Verarbeitung von Objektdefinitionen zu unterstützen.

Weitere Informationen zum Dienstprogramm CSQUTIL finden Sie im Abschnitt [IBM MQ-Dienstprogramm \(CSQUTIL\)](#). Wenn Sie dieses Dienstprogramm verwenden, können Sie die folgenden Funktionen aufrufen:

COMMAND

Zum Ausgeben von MQSC-Befehlen, zum Aufzeichnen von Objektdefinitionen und zum Ausführen von Clientkanaldefinitionsdateien.

COPY

Um den Inhalt einer benannten IBM MQ for z/OS-Nachrichtenwarteschlange oder den Inhalt aller Warteschlangen einer benannten Seitengruppe zu lesen, in eine sequenzielle Datei zu stellen und die ursprüngliche Warteschlange beizubehalten.

COPYPAGE

Gehen Sie wie folgt vor, um ganze Seitengruppen auf größere Seitengruppen zu kopieren

EMPTY

Um den Inhalt einer benannten IBM MQ for z/OS-Nachrichtenwarteschlange oder den Inhalt aller Warteschlangen einer benannten Seitengruppe zu löschen, unter Beibehaltung der Definitionen der Warteschlangen.

FORMAT

Um IBM MQ for z/OS-Seitengruppen zu formatieren.

LOAD

Zum Wiederherstellen des Inhalts einer benannten IBM MQ for z/OS-Nachrichtenwarteschlange oder des Inhalts aller Warteschlangen einer benannten Seitengruppe aus einer sequenziellen Datei, die von der Funktion COPY erstellt wurde.

PAGEINFO

Gehen Sie wie folgt vor, um Seitengruppen aus einer oder mehreren Seitengruppen zu extrahieren.

RESETPAGE

Wenn Sie ganze Seitengruppen auf andere Seitengruppen kopieren möchten, setzen Sie die Protokollinformationen in der Kopie zurück.

SCOPY

Zum Kopieren des Inhalts einer Warteschlange in eine Datei, während der WS-Manager offline ist.

SDEFS

Zum Erzeugen einer Gruppe von define-Befehlen für Objekte, während der Warteschlangenmanager offline ist.

SLOAD

Zum Zurückschreiben von Nachrichten aus der Zieldatei einer früheren COPY-oder SCOPY-Operation. SLOAD verarbeitet eine einzelne Warteschlange.

SWITCH

Sie können die Übertragungswarteschlange, die Clustersenderkanälen zugeordnet ist, abfragen oder abfragen.

XPARM

Gehen Sie wie folgt vor, um ein Kanalinitiatorparameterlademodul in WS-Manager-Attribute (für Migrationszwecke) zu konvertieren

z/OS IBM MQ for z/OS betreiben

Verwenden Sie diese grundlegenden Prozeduren, um IBM MQ for z/OS zu betreiben.

Sie können die in diesem Abschnitt beschriebenen Operationen auch mit dem IBM MQ Explorer ausführen, der mit IBM MQ for Windows, IBM MQ für Linux (x86- und x86-64-Plattformen) und SupportPac MS0T verteilt wird. Weitere Informationen finden Sie unter [„Verwaltung über die IBM MQ Explorer“](#) auf Seite 135 und [IBM Support & Downloads](#).

Dieser Abschnitt enthält Informationen zu den folgenden Themen:

z/OS Warteschlangenmanagerbefehle unter z/OS ausgeben

Sie können IBM MQ-Steuerbefehle über eine z/OS-Konsole oder mit dem Dienstprogramm CSQUTIL ausgeben. Befehle können eine Befehlspräfixzeichenfolge (CPF) verwenden, um anzugeben, welches IBM MQ-Subsystem den Befehl verarbeitet.

Sie können den Großteil der IBM MQ-Betriebsumgebung mithilfe der IBM MQ-Befehle steuern. IBM MQ for z/OS unterstützt sowohl die MQSC- als auch die PCF-Typen dieser Befehle. In diesem Abschnitt wird beschrieben, wie Attribute mit Hilfe von MQSC-Befehlen angegeben werden, und verweist daher nicht auf ihre PCF-Namen, sondern auf die Befehle und Attribute, die ihre MQSC-Befehlsnamen verwenden. Ausführliche Informationen zur Syntax der MQSC-Befehle finden Sie im Abschnitt [MQSC-Befehle](#). Ausführliche Informationen zur Syntax der PCF-Befehle finden Sie im Abschnitt [„IBM MQ Programmable Command Formats verwenden“](#) auf Seite 23. Wenn Sie ein entsprechend berechtigter Benutzer sind, können Sie IBM MQ-Befehle aus folgenden Quellen ausgeben:

- Initialisierungseingabedateien (siehe Beschreibung in [„Initialisierungsbefehle für IBM MQ for z/OS“](#) auf Seite 345).
- z/OS-Konsole oder Äquivalent, z. B. SDSF
- z/OS Master Get Command Routine, MGCRE (SVC 34)
- IBM MQ-Dienstprogramm CSQUTIL (siehe Beschreibung in [IBM MQ-Dienstprogramm](#))
- Eine Benutzeranwendung, die Folgendes sein kann:
 - Ein CICS -Programm
 - Ein TSO-Programm
 - Ein z/OS-Stapelverarbeitungsprogramm
 - Ein IMS-Programm

Informationen hierzu finden Sie in [„Programme zur Verwaltung von IBM MQ for z/OS schreiben“](#) auf Seite 371.

Ein großer Teil der Funktionalität dieser Befehle wird auf einfache Weise von den Operationen und Steuerkonsolen bereitgestellt, auf die über TSO und ISPF zugegriffen werden kann und die im Abschnitt [„Operationen und Steuerkonsolen für IBM MQ for z/OS“](#) auf Seite 355 beschrieben werden.

Weitere Informationen finden Sie unter

- [„Befehle über eine z/OS-Konsole oder deren Äquivalent ausgeben“](#) auf Seite 350
 - [Befehlspräfixzeichenfolgen](#)
 - [Befehle über die z/OS-Konsole ausgeben](#)
 - [Befehlsantworten](#)
- [Befehle aus dem Dienstprogramm CSQUTIL ausgeben](#)

Befehle über eine z/OS-Konsole oder deren Äquivalent ausgeben

Sie können alle IBM MQ-Befehle über eine z/OS-Konsole oder deren Äquivalent ausgeben. Sie können IBM MQ-Befehle auch überall dort ausgeben, wo Sie z/OS-Befehle ausgeben können, z. B. SDSF oder in einem Programm, das das Makro MGCRE verwendet.

Die maximale Datenmenge, die als Ergebnis eines in der Konsole eingegebenen Befehls angezeigt werden kann, beträgt 32 KB.

Anmerkung:

1. IBM MQ-Befehle können nicht mit dem Befehlsformat IMS/SSR von einem IMS-Terminal aus ausgegeben werden. Diese Funktion wird vom IMS-Adapter nicht unterstützt.
2. Das von SDSF bereitgestellte Eingabefeld ist für einige Befehle möglicherweise nicht lang genug, insbesondere die Befehle für Kanäle.

Befehlspräfixzeichenfolgen

Jedem IBM MQ-Befehl muss eine Befehlspräfixzeichenfolge (CPF) als Präfix vorangestellt werden, wie in [Abbildung 41](#) auf Seite 350 dargestellt.

Da mehr als ein IBM MQ-Subsystem unter z/OS ausgeführt werden kann, wird mit der CPF angegeben, welches IBM MQ-Subsystem den Befehl verarbeitet. Um beispielsweise den Warteschlangenmanager für ein Subsystem mit dem Namen CSQ1 zu starten, wobei CPF ' +CSQ1 ' ist, geben Sie den Befehl +CSQ1 START QMGR von der Bedienerkonsole aus. Dieser CPF muss in der Subsystemnamentabelle definiert sein (für das Subsystem CSQ1). Dies wird im Abschnitt [Befehlspräfixzeichenfolgen definieren](#) (CPFs) beschrieben. In den Beispielen wird die Zeichenfolge ' +CSQ1 ' als Befehlspräfix verwendet.

Befehle über die z/OS-Konsole ausgeben

Sie können einfache Befehle über die z/OS-Konsole eingeben, z. B. den Befehl DISPLAY in [Abbildung 41](#) auf Seite 350. Für komplexe Befehle oder für Gruppen von Befehlen, die häufig ausgegeben werden, sind die anderen Methoden zum Absetzen von Befehlen jedoch besser.

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLLOCAL)
```

Abbildung 41. DISPLAY-Befehl über die z/OS-Konsole ausgeben

Befehlsantworten

Direkte Antworten auf Befehle werden an die Konsole gesendet, die den Befehl ausgegeben hat. IBM MQ unterstützt die in z/OS verfügbare Funktion EMCS (*Extended Console Support*), weshalb Konsolen mit 4-Byte-IDs verwendet werden können. Darüber hinaus unterstützen alle Befehle außer START QMGR und STOP QMGR die Verwendung von Befehls- und Antworttoken (CARTs), wenn der Befehl von einem Programm unter Verwendung des MGCRC-Makros ausgegeben wird.

Befehle aus dem Dienstprogramm CSQUTIL absetzen

Sie können Befehle aus einer sequenziellen Datei mit der Funktion COMMAND des Dienstprogramms CSQUTIL ausgeben. Dieses Dienstprogramm überträgt die Befehle als Nachrichten an die *Eingabewarteschlange für Systembefehle* und wartet auf die Antwort, die zusammen mit den ursprünglichen Befehlen in SYSPRINT gedruckt wird. Weitere Einzelheiten dazu finden Sie im Abschnitt [IBM MQ-Dienstprogramm](#).

Warteschlangenmanager unter z/OS starten und stoppen

Verwenden Sie dieses Thema als Einführung zum Stoppen und Starten eines Warteschlangenmanagers.

In diesem Abschnitt wird die Vorgehensweise beim Starten und Stoppen eines Warteschlangenmanagers beschrieben. Sie enthält Informationen zu den folgenden Themen:

- „[Vor dem Start von IBM MQ](#)“ auf Seite 351
- „[WS-Manager starten](#)“ auf Seite 351
- „[Stoppen eines Warteschlangenmanagers](#)“ auf Seite 354

Das Starten und Stoppen eines Warteschlangenmanagers ist relativ einfach. Wenn ein Warteschlangenmanager unter normalen Bedingungen stoppt, wird die letzte Aktion ausgeführt, um einen Beendigungsprüfpunkt zu verwenden. Dieser Prüfpunkt und die Protokolle geben dem WS-Manager die Informationen, die er erneut starten muss.

Dieser Abschnitt enthält Informationen zu den Befehlen START und STOP und enthält eine kurze Übersicht über den Start, nachdem eine abnormale Beendigung aufgetreten ist.

Vor dem Start von IBM MQ

Nachdem Sie IBM MQ installiert haben, wird es als formales z/OS-Subsystem definiert. Bei jedem IPL (einleitendes Programmladen) von z/OS wird folgende Nachricht angezeigt:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

Dabei steht *ssnm* für den Namen des IBM MQ-Subsystems.

Ab jetzt können Sie den Warteschlangenmanager für dieses Subsystem *von jeder z/OS-Konsole aus starten, die zur Ausgabe von Systemsteuerbefehlen berechtigt ist*, d. h. eine z/OS-SYS-Befehlsgruppe. Sie müssen den Befehl START über die berechtigte Konsole absetzen. Sie können ihn nicht über JES oder TSO ausgeben.

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen Sie erst RRS und anschließend Db2 starten, bevor Sie den Warteschlangenmanager starten.

WS-Manager starten

Sie starten einen Warteschlangenmanager, indem Sie den Befehl START QMGR absetzen. Sie können den Befehl START jedoch nur dann erfolgreich verwenden, wenn Sie über die entsprechende Berechtigung verfügen. Informationen zur IBM MQ-Sicherheit finden Sie im Abschnitt [Sicherheit unter z/OS einrichten](#).

Abbildung 42 auf Seite 352 zeigt Beispiele für den Befehl START. (Denken Sie daran, dass Sie einem IBM MQ-Befehl eine Befehlspräfixzeichenfolge (CPF) voranstellen müssen.)

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

Abbildung 42. Warteschlangenmanager über eine z/OS-Konsole starten

Weitere Informationen zur Syntax des Befehls START QMGR finden Sie in [START QMGR](#).

Sie können den Warteschlangenmanager nicht als Stapeljob ausführen oder mit dem z/OS-Befehl START starten. Mit diesen Methoden wird wahrscheinlich ein Adressraum für IBM MQ gestartet, der dann abnormal beendet wird. Sie können auch keinen WS-Manager aus dem Dienstprogramm CSQUTIL oder einer ähnlichen Benutzeranwendung starten.

Sie können jedoch einen Warteschlangenmanager aus einem APF-autorisierten Programm starten, indem Sie den Befehl START QMGR an den z/OS-Service MGCRC (SVC 34) übergeben.

Wenn Sie Gruppen mit gemeinsamer Warteschlange verwenden, müssen die zugeordneten Db2-Systeme und RRS beim Starten des Warteschlangenmanagers aktiv sein.

Startoptionen

Wenn Sie einen WS-Manager starten, wird ein Systemparametermodul geladen. Sie können den Namen des Systemparametermoduls auf eine von zwei Arten angeben:

- Mit dem Parameter PARM des Befehls /cpf START QMGR zum Beispiel

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- Mit einem Parameter in der Startprozedur, z. B. Code der JCL-EXEC-Anweisung als

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARAM(CSQ1ZPRM)'
```

Ein Systemparametermodul stellt Informationen bereit, die bei der Anpassung des Warteschlangenmanagers angegeben wurden.

V 9.0.3 Ab IBM MQ 9.0.3 können Sie mit der Option **QMGRPROD** das Produkt angeben, für das die Verwendung des Warteschlangenmanagers aufgezeichnet werden soll, und mit der Option **AMSPROD** das Äquivalent für AMS angeben, wenn es verwendet wird. Weitere Informationen zu den zulässigen Werten finden Sie im MQSC-Befehl [START QMGR](#).

V 9.0.3 Es folgt eine Beispiel-JCL-EXEC-Anweisung:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

Weitere Informationen zur Produktnutzungserfassung finden Sie unter [z/OS MVS Product Management](#).

Sie können auch die Option ENVPARM verwenden, um einen oder mehrere Parameter in der JCL-Prozedur für den Warteschlangenmanager zu ersetzen.

Sie können z. B. die Startprozedur des Warteschlangenmanagers aktualisieren, sodass der DDname CSQINP2 eine Variable ist. Dies bedeutet, dass Sie den DDnamen CSQINP2 ändern können, ohne die Startprozedur zu ändern. Dies ist nützlich für die Implementierung von Änderungen, die Bereitstellung von Backouts für Operatoren und WS-Manageroperationen.

Nehmen wir an, dass Ihre Startprozedur für Warteschlangenmanager CSQ1 wie das Beispiel in [Abbildung 43 auf Seite 353](#) aussieht.


```

//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
// DD DISP=SHR,DSN=thlqual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BSDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BSDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqual.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*

```

Abbildung 43. Beispielstartprozedur

Wenn Sie den WS-Manager dann mit dem Befehl starten:

```
+CSQ1 START QMGR
```

Der verwendete CSQINP2 ist ein Member mit dem Namen CSQ1NORM.

Angenommen, Sie stellen eine neue Suite von Programmen in die Produktion ein, so dass die CSQINP2-Definitionen beim nächsten Start des WS-Managers CSQ1 aus dem Member CSQ1NEW übernommen werden. Dazu müssen Sie den WS-Manager mit diesem Befehl starten:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

und CSQ1NEW würde an Stelle von CSQ1NORM verwendet. Hinweis: z/OS begrenzt die Spezifikationen von KEYWORD=value für symbolische Parameter (wie in INP2=NEW) auf 255 Zeichen.

Starten nach einer abnormalen Beendigung

IBM MQ erkennt automatisch, ob ein Neustart auf einen normalen Systemabschluss oder eine abnormale Beendigung folgt.

Das Starten eines Warteschlangenmanagers nach dessen Beendigung unterscheidet sich vom Start nach dem Absetzen des Befehls STOP QMGR nicht. Nach dem Befehl STOP QMGR beendet das System seine Arbeit ordnungsgemäß und nimmt einen Beendigungsprüfpunkt vor, bevor er gestoppt wird. Wenn Sie den Warteschlangenmanager erneut starten, verwendet er Informationen aus dem Systemprüfpunkt und dem Wiederherstellungsprotokoll, um den Systemstatus bei Beendigung zu ermitteln.

Wenn der Warteschlangenmanager jedoch abnormal beendet wird, wird er beendet, ohne seine Arbeit beenden zu können oder einen Beendigungsprüfpunkt zu verwenden. Wenn Sie einen WS-Manager nach einer abnormalen Beendigung erneut starten, aktualisiert er seine Kenntnisse über seinen Status bei Beendigung mit Hilfe von Informationen im Protokoll und benachrichtigt Sie über den Status verschiedener Tasks. Normalerweise löst der Neustartprozess alle inkonsistenten Status auf. In einigen Fällen müssen Sie jedoch bestimmte Schritte unternehmen, um Inkonsistenzen zu beheben.

Benutzernachrichten beim Start

Wenn Sie einen WS-Manager erfolgreich starten, generiert der Warteschlangenmanager eine Gruppe von Startnachrichten.

Stoppen eines Warteschlangenmanagers

Bevor ein Warteschlangenmanager gestoppt wird, müssen alle IBM MQ-bezogenen WTOR-Nachrichten (Write-to-Operator-with-reply) Antworten empfangen, z. B. Abrufen von Protokollanforderungen. Jeder Befehl in [Abbildung 44 auf Seite 354](#) beendet einen aktiven Warteschlangenmanager.

```
+CSQ1 STOP QMGR
+CSQ1 STOP QMGR MODE(QUIESCE)
+CSQ1 STOP QMGR MODE(FORCE)
+CSQ1 STOP QMGR MODE(RESTART)
```

Abbildung 44. Stoppen eines Warteschlangenmanagers

Der Befehl STOP QMGR nimmt standardmäßig den Wert STOP QMGR MODE (QUIESCE) an.

Im Quiescemodus lässt IBM MQ nicht zu, dass neue Verbindungsthreads erstellt werden, ermöglicht aber die Fortsetzung vorhandener Threads; die Beendigung erfolgt nur, wenn alle Threads beendet sind. Anwendungen können beim Quiescemodus des Warteschlangenmanagers benachrichtigt werden, wenn sie benachrichtigt werden. Verwenden Sie deshalb den QUIESCE-Modus, wenn möglich, so dass Anwendungen, die eine Benachrichtigung angefordert haben, die Möglichkeit haben, die Verbindung zu trennen. Weitere Informationen finden Sie im Artikel [Was während der Kündigung passiert](#).

Wenn der Warteschlangenmanager nicht in einem angemessenen Zeitraum als Antwort auf den Befehl STOP QMGR MODE (QUIESCE) beendet wird, kann mit dem Befehl DISPLAY CONN festgestellt werden, ob Verbindungsthreads vorhanden sind, und die erforderlichen Schritte zum Beenden der zugeordneten Anwendungen ausführen. Wenn keine Threads vorhanden sind, geben Sie den Befehl STOP QMGR MODE (FORCE) aus.

Die Befehle STOP QMGR MODE (QUIESCE) und STOP QMGR MODE (FORCE) machen die Registrierung von IBM MQ beim MVS Automatic Restart Manager (ARM) rückgängig und verhindern, dass ARM den Warteschlangenmanager automatisch erneut startet. Der Befehl STOP QMGR MODE (RESTART) funktioniert genauso wie der Befehl STOP QMGR MODE (FORCE), außer dass die Registrierung von IBM MQ beim ARM nicht zurückgenommen wird. Dies bedeutet, dass der WS-Manager für einen sofortigen automatischen Neustart berechtigt ist.

Wenn das IBM MQ-Subsystem nicht beim ARM registriert ist, wird der Befehl STOP QMGR MODE (RESTART) zurückgewiesen und folgende Nachricht an die z/OS-Konsole gesendet:

```
CSQY205I ARM element arm-element is not registered
```

Wenn diese Nachricht nicht ausgegeben wird, wird der WS-Manager automatisch erneut gestartet. Weitere Informationen zu ARM finden Sie in [„z/OS Automatic Restart Manager \(ARM\) verwenden“](#) auf Seite 436.

Brechen Sie den Adressraum nur ab, wenn STOP QMGR MODE(FORCE) den Warteschlangenmanager nicht beendet.

Wenn ein Warteschlangenmanager gestoppt wird, indem entweder der Adressraum abgebrochen oder der Befehl STOP QMGR MODE (FORCE) verwendet wird, bleibt die Konsistenz mit verbundenen CICS- oder IMS-Systemen bestehen. Die Resynchronisation von Ressourcen wird gestartet, wenn ein Warteschlangenmanager erneut gestartet wird und abgeschlossen ist, wenn die Verbindung zum CICS- oder IMS-System hergestellt wird.

Anmerkung: Wenn Sie den WS-Manager stoppen, wird möglicherweise die Nachricht IEF352I ausgegeben. z/OS gibt diese Nachricht aus, wenn es feststellt, dass das Fehlschlagen der Markierung des Adressraums als unbrauchbar zu einem Integritätsrisiko führen würde. Sie können diese Nachricht ignorieren.

Nachrichten stoppen

Nachdem Sie den Befehl STOP QMGR abgesetzt haben, erhalten Sie die Nachrichten CSQY009I und CSQY002I, z. B.:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM  
USER(userid), STOP MODE(FORCE)  
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Dabei ist *userid* die Benutzer-ID, die den Befehl STOP QMGR ausgegeben hat, und der Parameter MODE hängt von dem im Befehl angegebenen Wert ab.

Wenn der Befehl STOP erfolgreich ausgeführt wurde, werden folgende Nachrichten an der z/OS-Konsole angezeigt:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION  
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

Wenn Sie ARM verwenden und MODE (RESTART) nicht angegeben haben, wird die folgende Nachricht ebenfalls angezeigt:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type  
arm-element-type successful
```

Sie können den WS-Manager erst dann erneut starten, wenn die folgende Nachricht angezeigt wurde:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```

Operationen und Steuerkonsolen für IBM MQ for z/OS

Sie können die IBM MQ- Operationen und -Steuerkonsolen verwenden, um Verwaltungstasks für IBM MQ-Objekte auszuführen. Verwenden Sie dieses Thema als Einführung in die Befehle und Steuerkonsolen.

Sie verwenden diese Konsolen, um IBM MQ-Objekte zu definieren, anzuzeigen, zu ändern oder zu löschen. Verwenden Sie die Anzeigen für die Verwaltung von Tageszeitungen und für die Durchführung kleiner Änderungen an Objekten. Wenn Sie viele Objekte einrichten oder ändern, verwenden Sie die Funktion COMMAND des Dienstprogramms CSQUTIL.

Die Operationen und Steuerkonsolen unterstützen die Steuerelemente für den Kanalinitiator (z. B. zum Starten eines Kanals oder eines TCP/IP-Listeners), für das Clustering und für die Sicherheit. Sie ermöglichen es Ihnen auch, Informationen zu Threads und Seitengarverwendung anzuzeigen.

Die Konsolen funktionieren, indem die IBM MQ-Befehle des Typs MQSC über Eingabewarteschlange für Systembefehle an einen Warteschlangenmanager gesendet werden.

Anmerkung:

1. Die z/OS IBM MQ-Operationen und -Steuerkonsolen (CSQOREXX) unterstützen möglicherweise nicht alle neuen Funktionen und Parameter, die ab Version 7 hinzugefügt wurden. Es gibt z. B. keine Anzeigen für die direkte Bearbeitung von Themenobjekten oder Subskriptionen.

Wenn Sie einen der folgenden unterstützten Mechanismen verwenden, können Sie Publish/Subscribe-Definitionen und andere Systemsteuerelemente verwalten, die nicht direkt von anderen Anzeigen aus verfügbar sind:

- a. IBM MQ Explorer
- b. z/OS-Konsole
- c. Programmable Command Format (PCF)-Nachrichten
- d. COMMAND-Funktion von CSQUTIL

Beachten Sie, dass Sie mit der generischen **Command** -Aktion in den CSQOREXX-Anzeigen jeden gültigen MQSC-Befehl ausgeben können, einschließlich SMDS-bezogener Befehle. Sie können alle Befehle verwenden, die von der Funktion COMMAND von CSQUTIL-Problemen verwendet werden.

2. Die IBM MQ-Befehle können nicht direkt über die Befehlszeile in den Konsolen ausgegeben werden.
3. Um die Operationen und Steuerkonsolen verwenden zu können, müssen Sie über die korrekte Sicherheitsberechtigung verfügen. Dies wird in den Benutzer-IDs für die Befehlsicherheit und die Sicherheit der Befehlsressourcen beschrieben.
4. Sie können keine Benutzer-ID und kein Kennwort mit CSQUTIL oder den CSQOREXX-Anzeigen angeben. Wenn Sie stattdessen die Berechtigung UPDATE für das Profil BATCH in MQCONN haben, können Sie die Einstellung **CHCKLOCL** (Einstellung *REQUIRED*) umgehen. Weitere Informationen finden Sie unter **CHCKLOCL** in lokal gebundenen Anwendungen verwenden.

Aufruf und Regeln für die Operationen und Steuerkonsolen

Sie können IBM MQ steuern und Steuerbefehle über die ISPF-Anzeigen ausgeben.

Vorgehensweise beim Zugriff auf die IBM MQ-Operationen und -Steuerkonsolen

Wenn das Menü für die primären ISPF/PDF-Optionen für IBM MQ aktualisiert wurde, können Sie über dieses Menü auf die IBM MQ-Operationen und -Steuerkonsolen zugreifen. Weitere Informationen zum Aktualisieren des Menüs finden Sie im Abschnitt Task 20: Operationen und Steuerkonsolen konfigurieren.

Sie können auf die IBM MQ-Operationen und -Steuerkonsolen über die TSO-Befehlsprozessoranzeige zugreifen (normalerweise Option 6 im Menü für die primären ISPF/PDF-Optionen). Der Name der Exec, die Sie ausführen, ist CSQOREXX. Es hat zwei Parameter: `thlqua1` ist das übergeordnete Qualifikationsmerkmal für die zu verwendenden IBM MQ -Bibliotheken und `langLetter` ist der Buchstabe, der die zu verwendenden Bibliotheken in der Landessprache angibt (z. B. E für U.S. Englisch). Die Parameter können weggelassen werden, wenn die IBM MQ-Bibliotheken permanent in der ISPF-Konfiguration installiert sind. Alternativ können Sie CSQOREXX über die TSO-Befehlszeile ausgeben.

Diese Anzeigen sind so konzipiert, dass sie von Bedienern und Administratoren mit einem Minimum an formaler Schulung verwendet werden. Lesen Sie diese Anweisungen mit den Anzeigen, die ausgeführt werden, und versuchen Sie, die verschiedenen Tasks auszuführen.

Anmerkung: Bei der Verwendung der Anzeigen temporäre dynamische Warteschlangen mit Namen im Format SYSTEM.CSQOREXX.* werden erstellt.

Regeln für die Operationen und Steuerkonsolen

Siehe Regeln für die Benennung von IBM MQ-Objekten zu den allgemeinen Regeln für Zeichenfolgen und Namen von IBM MQ-Zeichenfolgen. Es gibt jedoch einige Regeln, die nur für die Operationen und Steuerkonsolen gelten:

- Zeichenfolgen, z. B. Beschreibungen, dürfen nicht in einfache oder doppelte Anführungszeichen eingeschlossen werden.
- Wenn Sie ein Hochkomma oder ein Anführungszeichen in einem Textfeld enthalten, müssen Sie es nicht wiederholen oder ein Escapezeichen hinzufügen. Die Zeichen werden genau so gespeichert, wie Sie sie eingeben, z. B.:

This is Maria's queue

Der Anzeigenprozessor verdoppelt sie, damit Sie sie an IBM MQ übergeben können. Wenn Sie Ihre Daten jedoch abschneiden müssen, tut sie dies.

- In den meisten Feldern können Groß- oder Kleinbuchstaben verwendet werden, und bei Drücken der Eingabetaste werden die Zeichen in Großbuchstaben geschrieben. Die Ausnahmen sind:
 - Speicherklassennamen und Coupling-Facility-Strukturnamen, die mit der Großschreibung A bis Z beginnen müssen, gefolgt von einem Großbuchstaben von A bis Z oder numerischen Zeichen.
 - Bestimmte Felder, die nicht übersetzt sind. Hierzu gehören folgende Aufrufe:
 - Anwendungs-ID
 - Beschreibung
 - Umgebungsdaten
 - Objektname (wenn Sie jedoch einen Objektname in Kleinbuchstaben verwenden, können Sie ihn möglicherweise nicht an einer z/OS-Konsole eingeben.)
 - Name des fernen Systems
 - Daten des Auslösers
 - Benutzerdaten
- In Namen werden führende Leerzeichen und führende Unterstreichungszeichen ignoriert. Daher können Objektname, die mit Leerzeichen oder Unterstrichen beginnen, nicht beginnen.
- Unterstreichungszeichen werden verwendet, um den Umfang der leeren Felder anzuzeigen. Wenn Sie die Eingabetaste drücken, werden die abschließenden Unterstriche durch Leerzeichen ersetzt.
- Viele Beschreibungs- und Textfelder werden in mehreren Teilen dargestellt, wobei jeder Teil von IBM MQ unabhängig verarbeitet wird. Dies bedeutet, dass abschließende Leerzeichen beibehalten werden und der Text nicht zusammenhängend ist.

Leere Felder

Wenn Sie die Aktion **Definieren** für ein IBM MQ-Objekt angeben, enthält jedes Feld in der Definitionsanzeige einen Wert. Informationen dazu, wo IBM MQ die Werte erhält, finden Sie in der allgemeinen Hilfe (erweiterte Hilfe) für die Anzeigen. Wenn Sie in einem Feld Leerzeichen eingeben, Leerzeichen aber nicht zulässig sind, setzt IBM MQ den Installationsstandardwert in dem Feld ein oder fordert Sie zur Eingabe des erforderlichen Werts auf.

Wenn Sie die Aktion **Ändern** für ein IBM MQ-Objekt angeben, enthält jedes Feld in der Änderungsanzeige den aktuellen Wert für dieses Feld. Wenn Sie ein Feld mit Leerzeichen überschreiben, Leerzeichen jedoch nicht zulässig sind, bleibt der Wert dieses Feldes unverändert.

Objekte und Aktionen unter z/OS

Die Bedienungs- und Steuerkonsolen bieten Ihnen viele verschiedene Objekttypen und eine Reihe von Aktionen an, die Sie für diese Objekte ausführen können.

Die Aktionen werden in der Eingangsanzeige aufgelistet und ermöglichen es Ihnen, die Objekte zu bearbeiten und Informationen zu diesen Objekten anzuzeigen. Zu diesen Objekten gehören alle IBM MQ-Objekte, zusammen mit einigen zusätzlichen Objekten. Die Objekte fallen in die folgenden Kategorien.

- Warteschlangen, Prozesse, Authentifizierungsdatenobjekte, Namenslisten, Speicherklassen und CF-Strukturen
- Kanäle
- Clusterobjekte

- [Warteschlangenmanager und Sicherheit](#)
- [Verbindungen](#)
- [System](#)

Unter [Aktionen](#) finden Sie eine Querverweistabelle der Aktionen, die mit den IBM MQ -Objekten ausgeführt werden können.

Warteschlangen, Prozesse, Authentifizierungsinformationsobjekte, Namenslisten, Speicherklassen und CF-Strukturen

Dies sind die grundlegenden IBM MQ-Objekte. Es können viele von jedem Typ vorhanden sein. Sie können aufgelistet, mit Filter aufgelistet, definiert und gelöscht werden und haben Attribute, die mit den Aktionen LIST oder DISPLAY, LIST mit den Aktionen FILTER, DEFINE LIKE, MANAGE und ALTER angezeigt und geändert werden können. (Objekte werden mit der Aktion MANAGE gelöscht.)

Diese Kategorie besteht aus den folgenden Objekten:

QLOCAL	Lokale Warteschlange
QREMOTE	Ferne Warteschlange
QALIAS	Aliaswarteschlange für indirekten Verweis auf eine Warteschlange
QMODEL	Modellwarteschlange zum dynamischen Definieren von Warteschlangen
WARTESCHLANGEN	Beliebiger Warteschlangentyp
QSTATUS	Status einer lokalen Warteschlange
PROCESS	Informationen zu einer Anwendung, die beim Auftreten eines Auslöserereignisses gestartet werden soll
AUTHINFO	Authentifizierungsdaten: Definitionen, die zum Ausführen der CRL-Prüfung (Certificate Revocation List, Zertifikatswiderrufliste) mit LDAP-Servern erforderlich sind
NAMELIST	Liste der Namen, wie z. B. Warteschlangen oder Cluster
STGKLASSE	Speicherklasse
CFSTRUCT	CF-Struktur (Coupling Facility)
CFSTATUS	Status einer CF-Struktur

Kanäle

Kanäle werden für verteilte Warteschlangensteuerung verwendet. Es können viele von jedem Typ vorhanden sein, und sie können aufgelistet, mit Filter aufgelistet, definiert, gelöscht, angezeigt und geändert werden. Sie verfügen außerdem über weitere Funktionen, die mit den Aktionen START, STOP und PERFORM verfügbar sind. PERFORM stellt die Kanalfunktionen zum Zurücksetzen, Pingsignal und zum Auflösen bereit.

Diese Kategorie besteht aus den folgenden Objekten:

CHANNEL	Beliebiger Kanaltyp
Absender	Senderkanal
Server	Serverkanal
Empfänger	Empfängerkanal
REQUESTER	Requesterkanal
CLUSRCVR	Clusterempfängerkanal
CLUSSDR	Clustersenderkanal

SVRCONN	Serververbindungskanal
CLNTCONN	Clientverbindungskanal
CHSTATUS	Status einer Kanalverbindung

Clusterobjekte

Clusterobjekte werden automatisch für Warteschlangen und Kanäle erstellt, die zu einem Cluster gehören. Die Basiswarteschlange und die Kanaldefinitionen können sich auf einem anderen Warteschlangenmanager befinden. Es können viele von jedem Typ vorhanden sein, und Namen können dupliziert werden. Sie können aufgelistet, mit Filter aufgelistet und angezeigt werden. PERFORM, START und STOP sind auch über die LIST-Aktionen verfügbar.

Diese Kategorie besteht aus den folgenden Objekten:

CLUSQ	Clusterwarteschlange, erstellt für eine Warteschlange, die zu einem Cluster gehört
CLUSCHL	Clusterkanal, erstellt für einen Kanal, der zu einem Cluster gehört
CLUSQMGR	Clusterwarteschlangenmanager, der mit einem Clusterkanal identisch ist, aber durch den Namen des WS-Managers identifiziert wird

Clusterkanäle und Cluster-WS-Manager verfügen über die Aktionen PERFORM, START und STOP, aber nur indirekt über die Aktion DISPLAY.

WS-Manager und Sicherheit

WS-Manager und Sicherheitsobjekte haben eine einzelne Instanz. Sie können aufgelistet werden und haben Attribute, die angezeigt und geändert werden können (mit den Aktionen LIST oder DISPLAY und ALTER) und über andere Funktionen verfügen, die mit Hilfe der Aktion PERFORM verfügbar sind.

Diese Kategorie besteht aus den folgenden Objekten:

MANAGER	Warteschlangenmanager: Die Aktion PERFORM stellt Clusterfunktionen für Aussetzen und Wiederaufnehmen bereit.
Sicherheit	Sicherheitsfunktionen: Die Aktion AUSFÜHREN bietet Funktionen zur Aktualisierung und erneuten Prüfung.

Verbindung

Verbindungen können aufgelistet, mit Filter aufgelistet und angezeigt werden.

Diese Kategorie besteht nur aus dem Verbindungsobjekt CONNECT.

System

Eine Sammlung anderer Funktionen. Diese Kategorie besteht aus den folgenden Objekten:

SYSTEM	Systemfunktionen
STEUERUNG	Synonym zu SYSTEM

Folgende Funktionen stehen zur Verfügung:

LIST oder DISPLAY	Informationen zur Gruppe mit gemeinsamer Warteschlange, zur verteilten Steuerung von Warteschlangen, zur Seitengruppe oder zur Dateigruppe anzeigen.
AUSFÜHREN	Clustering aktualisieren oder zurücksetzen
START	Kanalinitiator oder Empfangsprogramme starten
STOPP	Kanalinitiator oder Empfangsprogramme stoppen

Aktionen

Die Aktionen, die Sie für die einzelnen Objekttyp ausführen können, werden in der folgenden Tabelle angezeigt:

Objekt	Alter	Definieren wie	Verwalten (1)	Liste oder Anzeige	Liste mit Filter	Ausführen	Starten	Stop (Stoppen)
AUTHINFO	X	X	X	X	X			
CFSTATUS				X				
CFSTRUCT	X	X	X	X	X			
CHANNEL	X	X	X	X	X	X	X	X
CHSTATUS				X	X			
CLNTCONN	X	X	X	X	X			
CLUSCHL				X	X	X (2)	X (2)	X (2)
CLUSQ				X	X			
CLUSQMGR				X	X	X (2)	X (2)	X (2)
CLUSRCVR	X	X	X	X	X	X	X	X
CLUSDR	X	X	X	X	X	X	X	X
VERBINDEN				X	X			
STEUERUNG				X		X	X	X
MANAGER	X			X		X		
NAMELIST	X	X	X	X	X			
PROCESS	X	X	X	X	X			
QALIAS	X	X	X	X	X			
QLOCAL	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QREMOTE	X	X	X	X	X			
QSTATUS				X	X			
WARTE-SCHLANGE	X	X	X	X	X			
Empfänger	X	X	X	X	X	X	X	X
REQUESTER	X	X	X	X	X	X	X	X
Sicherheit	X			X		X		
Absender	X	X	X	X	X	X	X	X
Server	X	X	X	X	X	X	X	X
SVRCONN	X	X	X	X	X		X	X
STGKLASSE	X	X	X	X	X			
SYSTEM				X		X	X	X

Anmerkung:

1. Stellt Löschen und andere Funktionen bereit.
2. Verwenden der Aktion `Liste` oder `Anzeige`

Objektdispositionen unter z/OS

Sie können die *Disposition* des Objekts angeben, mit dem gearbeitet werden soll. Die Disposition gibt an, wo die **Definition** des Objekts gespeichert ist und wie sich das Objekt verhält.

Die Disposition ist nur dann von Bedeutung, wenn Sie mit einer der folgenden Objekttypen arbeiten:

- Warteschlangen
- Kanäle
- Prozesse
- Namenslisten
- Speicherklassen
- Authentifizierungsinformationsobjekte

Wenn Sie mit anderen Objektarten arbeiten, wird die Disposition nicht berücksichtigt.

Zulässige Werte sind:

Q

QMGR. Die Objektdefinitionen befinden sich auf der Seitengruppe des Warteschlangenmanagers und sind nur durch den Warteschlangenmanager zugänglich.

C

KOPIEREN Die Objektdefinitionen befinden sich auf der Seitengruppe des Warteschlangenmanagers und sind nur durch den Warteschlangenmanager zugänglich. Es handelt sich um lokale Kopien von Objekten, die als Disposition der Gruppe GROUP definiert sind.

P

PRIVATE. Die Objektdefinitionen befinden sich auf der Seitengruppe des Warteschlangenmanagers und sind nur durch den Warteschlangenmanager zugänglich. Die Objekte wurden als Disposition von QMGR oder COPY definiert.

G

GRUPPE. Die Objektdefinitionen befinden sich im gemeinsam genutzten Repository und sind von allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange zugänglich.

S

SHARED. Diese Disposition gilt nur für lokale Warteschlangen. Die Warteschlangendefinitionen befinden sich im gemeinsam genutzten Repository und sind von allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange zugänglich.

A

ALL. Wenn der Aktionswarteschlangenmanager entweder der Zielwarteschlangenmanager oder * ist, werden Objekte von **all** dispositionen eingeschlossen; andernfalls werden nur Objekte von QMGR und COPY Dispositionen eingeschlossen. Dies ist der Standardwert.

Warteschlangenmanager, Standardwerte und Ebenen über die ISPF-Steu- erkonsole unter z/OS auswählen

Sie können die CSQOREXX-Exec in ISPF verwenden, um die WS-Manager zu steuern.

Während Sie die Eingangsanzeige anzeigen, sind Sie nicht mit einem WS-Manager verbunden. Sobald Sie allerdings die Eingabetaste drücken, wird eine Verbindung zum Warteschlangenmanager oder zu einem Warteschlangenmanager in der angegebenen Gruppe mit gemeinsamer Warteschlange im Feld **Connect name** (Verbindungsname) hergestellt. Sie können dieses Feld leer lassen. Dies bedeutet, dass Sie den Standardwarteschlangenmanager für Stapelanwendungen verwenden. Dies ist in CSQBDEFV definiert (Informationen hierzu finden Sie im Abschnitt [Task 19: Batch-, TSO- und RRS-Adapter konfigurieren](#)).

Geben Sie im Feld **Zielwarteschlangenmanager** den Warteschlangenmanager an, in dem die Aktionen ausgeführt werden sollen, die ausgeführt werden sollen. Wenn Sie dieses Feld leer lassen, wird standardmäßig der im Feld **Verbindungsname** angegebene WS-Manager verwendet. Sie können einen Zielwarteschlangenmanager angeben, der nicht mit dem Warteschlangenmanager verbunden ist. In diesem Fall würden Sie normalerweise den Namen eines fernen WS-Manager-Objekts angeben, das eine Aliasdefinition des Warteschlangenmanagers bereitstellt (der Name wird als *ObjectQMgrName* verwendet, wenn die Befehlseingabewarteschlange geöffnet wird). Dazu müssen Sie über geeignete Warteschlangen und Kanäle verfügen, die für den Zugriff auf den fernen Warteschlangenmanager konfiguriert sind.

Im Feld **Action queue manager** (Aktionswarteschlangenmanager) können Sie einen Warteschlangenmanager angeben, der sich in der gleichen Gruppe mit gemeinsamer Warteschlange wie der im Feld **Target queue manager** (Zielwarteschlangenmanager) angegebene Warteschlangenmanager befindet und auf dem die angeforderten Aktionen ausgeführt werden sollen. Wenn Sie in diesem Feld * angeben, werden die von Ihnen angeforderten Aktionen auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange ausgeführt. Wenn Sie dieses Feld leer lassen, wird standardmäßig der im Feld **Zielwarteschlangenmanager** angegebene Wert verwendet. Das Feld **Aktionswarteschlangenmanager** entspricht der Verwendung des Befehlsmodifikators CMDSCOPE, der in [MQSC-Befehle](#) beschrieben ist.

Warteschlangenmanagerstandardwerte

Wenn Sie Warteschlangenmanagerfelder leer lassen oder eine Verbindung zu einer Gruppe mit gemeinsamer Warteschlange herstellen, wird beim Drücken der **Eingabetaste** ein zweites Fenster geöffnet. In diesem Fenster werden die Namen der Warteschlangenmanager bestätigt, die Sie verwenden werden. Drücken Sie die **Eingabetaste**, um fortzufahren. Wenn Sie nach einigen Anforderungen in die Eingangsanzeige zurückkehren, werden die Felder mit den tatsächlichen Namen gefunden.

Warteschlangenmanagerstufen

Die Betriebs- und Steuerkonsolen funktionieren nur mit Warteschlangenmanagern, die unter z/OS IBM WebSphere MQ 710 oder höher ausgeführt werden, zufriedenstellend.

Wenn diese Bedingungen nicht erfüllt sind, ist es wahrscheinlich, dass Aktionen nur teilweise, nicht ordnungsgemäß oder gar nicht funktionieren und dass die Antworten vom Warteschlangenmanager nicht erkannt werden.

Wenn der Aktionswarteschlangenmanager nicht mindestens über IBM MQ 8.0.0 verfügt, werden einige Felder nicht angezeigt, und einige Werte können nicht eingegeben werden. Einige Objekte und Aktionen sind nicht zulässig. In solchen Fällen wird ein zweites Fenster geöffnet, in dem Sie aufgefordert werden, zu bestätigen, dass Sie fortfahren möchten.

Funktionstasten und Befehlszeile mit den ISPF-Steuerkonsolen unter z/OS verwenden

Wenn Sie die Anzeigen verwenden möchten, müssen Sie die Funktionstasten verwenden oder die entsprechenden Befehle in den Befehlsbereich der ISPF-Steuerkonsole eingeben.

- [Funktionstasten](#)
 - [Aktionen verarbeiten](#)
 - [„IBM MQ-Benutzernachrichten anzeigen“ auf Seite 363](#)
 - [Aktionen abbrechen](#)
 - [Hilfe anfordern](#)
- [Befehlszeile verwenden](#)

Funktionstasten

Die Funktionstasten haben spezielle Einstellungen für IBM MQ. (Dies bedeutet, dass Sie nicht die ISPF-Standardwerte für die Funktionstasten verwenden können. Wenn Sie zuvor den Befehl KEYLIST OFF ISPF an einer beliebigen Stelle verwendet haben, müssen Sie KEYLIST ON im Befehlsbereich einer

beliebigen Operation und Steuerkonsole eingeben und dann die Eingabetaste drücken, um die IBM MQ-Einstellungen zu aktivieren.)

Diese Funktionstasteinstellungen können in den Anzeigen dargestellt werden, wie in [Abbildung 45](#) auf Seite 364 gezeigt wird. Wenn die Einstellungen nicht angezeigt werden, geben Sie PFSHOW im Befehlsbereich einer beliebigen Operations- und Steuerkonsole ein und drücken Sie dann die **Eingabetaste**. Verwenden Sie den Befehl PFSHOW OFF, um die Anzeige der Einstellungen zu entfernen.

Die Funktionstasteinstellungen in den Operations- und Steuerkonsolen entsprechen CUA-Standards. Sie können die Schlüsseleinstellung zwar über normale ISPF-Prozeduren (z. B. das Dienstprogramm **KEYLIST**) ändern, dies wird jedoch nicht empfohlen.

Anmerkung: Die Verwendung der Befehle **PFSHOW** und **KEYLIST** wirkt sich auf alle anderen logischen ISPF-Anzeigen aus, die Sie haben, und ihre Einstellungen bleiben erhalten, wenn Sie die Betriebs- und Steuerkonsolen verlassen.

Ihre Aktionen verarbeiten

Drücken Sie die **Eingabetaste**, um die Aktion auszuführen, die in einer Anzeige angefordert wurde. Die Informationen aus der Anzeige werden zur Verarbeitung an den WS-Manager gesendet.

Immer wenn Sie in den Anzeigen die **Eingabetaste** drücken, generiert IBM MQ eine oder mehrere Bedienernachrichten. Wenn die Operation erfolgreich war, erhalten Sie die Bestätigungsnachricht CSQ9022I. Andernfalls erhalten Sie einige Fehlernachrichten.

IBM MQ-Benutzernachrichten anzeigen

Drücken Sie Funktionstaste F10 in einer beliebigen Anzeige, um die IBM MQ-Benutzernachrichten anzuzeigen.

Aktionen abbrechen

In der Eingangsanzeige verlassen F3 und F12 die Operationen und Steuerkonsolen und kehren Sie zu ISPF zurück. Es werden keine Informationen an den WS-Manager gesendet.

Drücken Sie auf einer anderen Anzeige die Funktionstasten F3 oder F12, um die aktuelle Anzeige **Daten, die seit dem letzten Drücken der Eingabetaste eingegeben wurden, werden ignoriert**. zu verlassen. Auch hier werden keine Informationen an den WS-Manager gesendet.

- F3 führt Sie direkt in die Eingangsanzeige zurück.
- F12 bringt Sie zurück zur vorherigen Anzeige.

Hilfe abrufen

Jedem Teilfenster sind Hilfetafeln zugeordnet. In den Hilfetafeln werden die ISPF-Protokolle verwendet:

- Drücken Sie die Funktionstaste F1 in einer beliebigen Anzeige, um die allgemeine Hilfe (erweiterte Hilfe) zu der Task anzuzeigen.
- Drücken Sie die Funktionstaste F1 mit dem Cursor in einem beliebigen Feld, um eine spezielle Hilfe zu diesem Feld anzuzeigen.
- Drücken Sie die Funktionstaste F5 in einem beliebigen Feld-Hilfefenster, um die allgemeine Hilfe zu erhalten.
- Drücken Sie die Funktionstaste F3, um zur Basisplatte zurückzukehren, d. h. der Anzeige, von der Sie die Funktionstaste F1 gedrückt haben.
- Drücken Sie die Funktionstaste F6 in einer beliebigen Hilfetaste, um Hilfe zu den Funktionstasten zu erhalten.

Wenn die Hilfeinformationen auf einer zweiten oder nachfolgenden Seite fortgesetzt werden, wird dies oben rechts im Fenster durch den Hinweis **Mehr** angezeigt. Verwenden Sie die folgenden Funktionstasten, um durch die Hilfeseiten zu navigieren:

- F11, um zur nächsten Hilfetextseite zu gelangen (falls vorhanden).
- F10, um zurück zur vorherigen Hilfetextseite zu gelangen (falls vorhanden).

Verwenden der Befehlszeile

Sie müssen die Befehlszeile nicht verwenden, um die Befehle auszugeben, die von den Operations- und Steuerkonsolen verwendet werden, da sie über Funktionstasten verfügbar sind. Die Befehlszeile wird bereitgestellt, damit Sie normale ISPF-Befehle (wie **PFSHOW**) eingeben können.

Der ISPF-Befehl `PANELID ON` zeigt den Namen der aktuellen CSQOREXX-Anzeige an.

Die Befehlszeile wird anfänglich in der Standardposition am unteren Rand der Anzeigen angezeigt, unabhängig davon, welche ISPF-Einstellungen Sie haben. Sie können den Befehl `SETTINGS ISPF` über eine der Operationen und Steuerkonsolen verwenden, um die Position der Befehlszeile zu ändern. Die Einstellungen werden für nachfolgende Sitzungen mit den Operations- und Steuerkonsolen gespeichert.

Operationen und Steuerkonsolen unter z/OS verwenden

Verwenden Sie dieses Thema, um die Eingangssteuerkonsole zu untersuchen, die von CSQOREXX angezeigt wird.

Abbildung 45 auf Seite 364 zeigt die Anzeige, die angezeigt wird, wenn Sie eine Anzeigensitzung starten.

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.
Action . . . . . 1      0. List with filter  4. Manage
                   1. List or Display  5. Perform
                   2. Define like   6. Start
                   3. Alter        7. Stop
                   8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
                      S=Shared, A=All

Connect name . . . . . M01C - local queue manager or group
Target queue manager . . . M01C
                      - connected or remote queue manager for command input
Action queue manager . . . M01C - command scope in group
Response wait time . . . . 30 5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2023. All rights reserved.

Command ==>>
F1=Help      F2=Split    F3=Exit     F4=Prompt   F9=SwapNext F10=Messages
F12=Cancel
```

Abbildung 45. Eingangsanzeige IBM MQ-Operationen und Steuerung

In dieser Anzeige können Sie Aktionen wie die folgenden ausführen:

- Wählen Sie den lokalen Warteschlangenmanager aus und geben Sie an, ob die Befehle auf diesem Warteschlangenmanager ausgegeben werden sollen, auf einem fernen Warteschlangenmanager oder auf einem anderen Warteschlangenmanager, der sich in derselben Gruppe mit gemeinsamer Warteschlange wie der lokale Warteschlangenmanager befindet. Geben Sie den Namen des Warteschlangenmanagers ein, wenn Sie ihn ändern müssen.
- Wählen Sie die Aktion aus, die Sie ausführen möchten, indem Sie die entsprechende Zahl in das Feld **Aktion** eingeben.
- Geben Sie den Objekttyp an, mit dem gearbeitet werden soll. Drücken Sie die Funktionstaste F1, um Hilfe zu den Objekttypen zu erhalten, wenn Sie sich nicht sicher sind, was sie sind.
- Geben Sie die Disposition des Objekttyps an, mit dem gearbeitet werden soll.
- Zeigt eine Liste der Objekte des angegebenen Typs an. Geben Sie einen Stern (*) in das Feld **Name** ein, und drücken Sie die **Eingabetaste**, um eine Liste der Objekte (der angegebenen Art) anzuzeigen, die bereits im Aktionswarteschlangenmanager definiert wurden. Anschließend können Sie ein oder mehrere Objekte auswählen, mit der in der Sequenz gearbeitet werden soll. Alle Aktionen sind in der Liste verfügbar.

Anmerkung: Es wird empfohlen, eine Auswahl zu treffen, die eine Liste der Objekte, die angezeigt werden, und anschließend die Arbeit aus dieser Liste zu erstellen. Verwenden Sie die Aktion **Anzeigen**, da dies für alle Objekttypen zulässig ist.

Befehlsfunktion unter z/OS verwenden

Verwenden Sie den Editor, um MQSC-Befehle einzugeben oder zu ändern, die an den Warteschlangenmanager übergeben werden sollen.

Wählen Sie in der Hauptanzeige (CSQOPRIA) die Option **8 Befehl** aus, um die Befehlsfunktion zu starten.

Es wird eine Bearbeitungssitzung mit der sequenziellen Datei '*Präfix*.CSQUTIL.COMMANDS' angezeigt, die als Eingabe für die Funktion CSQUTIL COMMAND verwendet wird (siehe [Befehle an IBM MQ absetzen](#)).

Es müssen keine Präfixbefehle mit der Befehlspräfixzeichenfolge (CPF) angegeben werden.

Sie können MQSC-Befehle in nachfolgenden Zeilen fortsetzen, indem Sie die aktuelle Zeile mit den Fortsetzungszeichen + oder -beenden. Alternativ können Sie den Zeilenbearbeitungsmodus verwenden, um lange MQSC-Befehle oder die Werte der Werte für lange Attribute im Befehl bereitzustellen.

Zeilenbearbeitung

Wenn Sie die Zeilenbearbeitung verwenden möchten, bewegen Sie den Cursor in die entsprechende Zeile in der Bearbeitungsanzeige und verwenden Sie **F4**, um eine einzelne Zeile in einer scrollbaren Anzeige anzuzeigen. Eine einzelne Zeile kann bis zu 32 760 Byte Daten enthalten.

Gehen Sie wie folgt vor, um die

- **F3-Exit** speichert Änderungen, die an der Leitung und den Exits vorgenommen wurden.
- **F12 cancel** kehrt zur Editieranzeige zurück, in der Änderungen verworfen werden, die an der Linie vorgenommen wurden.

Um Änderungen, die in der Editiersitzung vorgenommen wurden, zu verwerfen, verwenden Sie **F12 cancel**, um die Editiersitzung zu beenden und den Inhalt der Datei unverändert zu lassen. Befehle werden nicht ausgeführt.

Befehle ausführen

Wenn Sie die Eingabe von MQSC-Befehlen beendet haben, beenden Sie die Editiersitzung mit **F3 exit**, um den Inhalt der Datei zu speichern, und rufen Sie CSQUTIL auf, um die Befehle an den Warteschlangenmanager zu übergeben. Die Ausgabe der Befehlsverarbeitung wird in der Datei *präfix*.CSQUTIL.OUTPUT gehalten. Es wird automatisch eine Bearbeitungssitzung für diese Datei geöffnet, so dass Sie die Antworten anzeigen können. Drücken Sie die Taste **F3 exit**, um diese Sitzung zu beenden und zum Hauptmenü zurückzukehren.

Mit IBM MQ-Objekten unter z/OS arbeiten

Viele der in dieser Dokumentation beschriebenen Tasks schließen die Bearbeitung von IBM MQ-Objekten ein. Bei den Objekttypen handelt es sich um Warteschlangenmanager, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsinformationsobjekte.

- [Einfache Warteschlangenobjekte definieren](#)
- [Andere Objekttypen definieren](#)
- [Mit Objektdefinitionen arbeiten](#)
- [Mit Namenslisten arbeiten](#)

Einfache Warteschlangenobjekte definieren

Um ein neues Objekt zu definieren, verwenden Sie eine vorhandene Definition als Basis für diese Definition. Sie können dies auf eine der folgenden drei Arten tun:

- Durch Auswahl eines Objekts, das Teil einer Liste ist, die als Ergebnis von Optionen angezeigt wird, die in der Eingangsanzeige ausgewählt wurden. Anschließend geben Sie den Aktionstyp 2 (**Definieren wie**) in das Aktionsfeld neben dem ausgewählten Objekt ein. Das neue Objekt verfügt über die Attribute des ausgewählten Objekts mit Ausnahme der Disposition. Sie können dann alle Attribute in Ihrem neuen Objekt ändern, wie Sie es benötigen.

- Wählen Sie in der Eingangsanzeige den Aktionstyp **Definieren wie** aus, geben Sie den Typ des Objekts ein, das Sie im Feld **Objekttyp** definieren, und geben Sie den Namen eines bestimmten vorhandenen Objekts in das Feld **Name** ein. Das neue Objekt verfügt über dieselben Attribute wie das Objekt, das Sie im Feld **Name** mit Ausnahme der Disposition benannt haben. Sie können dann alle Attribute in der neuen Objektdefinition ändern, wie Sie dies wünschen.
- Wenn Sie den Aktionstyp **Definieren wie** auswählen, geben Sie einen Objekttyp an und lassen Sie das Feld **Name** leer. Anschließend können Sie das neue Objekt definieren und die Standardattribute für die Installation definieren. Sie können dann alle Attribute in der neuen Objektdefinition ändern, wie Sie dies wünschen.

Anmerkung: Sie geben nicht den Namen des Objekts ein, das Sie in der Eingangsanzeige definieren, sondern in der Anzeige **Definieren**, in der Sie den Namen des Objekts angeben.

Das folgende Beispiel veranschaulicht, wie eine lokale Warteschlange unter Verwendung einer vorhandenen Warteschlange als Schablone definiert wird.

Lokale Warteschlange definieren

Um ein lokales Warteschlangenobjekt aus den Operations- und Steuerkonsolen zu definieren, verwenden Sie eine vorhandene Warteschlangendefinition als Basis für Ihre neue Definition. Es gibt mehrere Anzeigen, die abgeschlossen werden können. Wenn Sie alle Anzeigen abgeschlossen haben und die Attribute korrekt sind, drücken Sie die Eingabetaste, um Ihre Definition an den Warteschlangenmanager zu senden, der dann die tatsächliche Warteschlange erstellt.

Verwenden Sie die Aktion **Definieren wie** entweder in der Eingangsanzeige oder gegen einen Objekteintrag in einer Liste, die als Ergebnis von Optionen angezeigt wird, die in der Eingangsanzeige ausgewählt wurden.

Geben Sie zum Beispiel in der Eingangsanzeige die folgenden Felder ein:

Aktion	2 (Definieren wie)
Objekttyp	QLOCAL
Name	QUEUE.YOU.LIKE. Dies ist der Name der Warteschlange, die die Attribute für Ihre neue Warteschlange bereitstellt.

Drücken Sie die Eingabetaste, um die Anzeige **Lokale Warteschlange definieren** aufzurufen. Das Feld für den Warteschlangennamen ist leer, so dass Sie den Namen für die neue Warteschlange angeben können. Die Beschreibung ist die der Warteschlange, auf der Sie diese neue Definition basieren. Geben Sie in diesem Feld eine eigene Beschreibung für die neue Warteschlange ein.

Die Werte in den anderen Feldern sind die Werte in der Warteschlange, auf die Sie diese neue Warteschlange (mit Ausnahme der Disposition) umstellen. Sie können diese Felder nach den erforderlichen Angaben überschreiben. Geben Sie z. B. Y in das Feld **Put enabled** ein (wenn es noch nicht Y ist), wenn entsprechend berechnete Anwendungen Nachrichten in diese Warteschlange stellen können.

Sie erhalten Feldhilfe, indem Sie den Cursor in ein Feld bewegen und die Funktionstaste F1 drücken. Die Feldhilfe enthält Informationen zu den Werten, die für jedes Attribut verwendet werden können.

Wenn Sie die erste Anzeige abgeschlossen haben, drücken Sie die Funktionstaste F8, um die zweite Anzeige aufzurufen.

Hints:

1. Drücken Sie in dieser Phase nicht die Eingabetaste, da andernfalls die Warteschlange erstellt wird, bevor Sie die verbleibenden Felder ausfüllen können. (Wenn Sie die Eingabetaste vorzeitig drücken, machen Sie keine Sorge. Sie können Ihre Definition später jederzeit ändern.)
2. Die Funktionstasten F3 oder F12 nicht drücken, oder die eingegebenen Daten gehen verloren.

Drücken Sie die Funktionstaste F8 mehrmals, um die verbleibenden Anzeigen zu sehen und zu beenden, einschließlich der Auslöserdefinition, der Ereignissteuerungs- und der Back-out-Berichtsanzeigen.

Wenn Ihre lokale Warteschlangendefinition abgeschlossen ist

Wenn die Definition abgeschlossen ist, drücken Sie die Eingabetaste, um die Informationen zur Verarbeitung an den Warteschlangenmanager zu senden. Der WS-Manager erstellt die Warteschlange entsprechend der von Ihnen angegebenen Definition. Wenn die Warteschlange nicht erstellt werden soll, drücken Sie die Funktionstaste F3, um die Definition zu beenden und die Definition abzubrechen.

Andere Typen von Objekten definieren

Wenn Sie andere Objekttypen definieren möchten, verwenden Sie eine vorhandene Definition als Basis für Ihre neue Definition, wie im Abschnitt Lokale Warteschlange definieren erläutert.

Verwenden Sie die Aktion **Definieren wie** entweder in der Eingangsanzeige oder gegen einen Objekteintrag in einer Liste, die als Ergebnis von Optionen angezeigt wird, die in der Eingangsanzeige ausgewählt wurden.

Geben Sie zum Beispiel in der Eingangsanzeige die folgenden Felder ein:

Aktion	2 (Definieren wie)
Objekttyp	QALIAS, NAMELIST, PROCESS, CHANNEL und andere Ressourcenobjekte.
Name	Lassen Sie das Feld leer, oder geben Sie den Namen eines vorhandenen Objekts desselben Typs ein.

Drücken Sie die Eingabetaste, um die entsprechenden DEFINE-Anzeigen aufzurufen. Füllen Sie die Felder nach Bedarf aus, und drücken Sie die Eingabetaste erneut, um die Informationen an den Warteschlangenmanager zu senden.

Wie die Definition einer lokalen Warteschlange erfordert die Definition eines anderen Objekttyp in der Regel mehrere Anzeigen, die abgeschlossen werden müssen. Für die Definition einer Namensliste sind einige zusätzliche Arbeiten erforderlich, wie im Abschnitt „Mit Namenslisten arbeiten“ auf Seite 368 beschrieben.

Mit Objektdefinitionen arbeiten

Wenn ein Objekt definiert wurde, können Sie eine Aktion im Feld **Aktion** angeben, um sie zu ändern, anzuzeigen oder zu verwalten.

In jedem Fall können Sie eine der folgenden Schritte ausführen:

- Wählen Sie das Objekt aus, mit dem Sie arbeiten möchten, aus einer Liste, die als Ergebnis von Optionen angezeigt wird, die in der Eingangsanzeige ausgewählt wurden. Wenn Sie beispielsweise 1 in das Feld **Aktion** eingegeben haben, um Objekte anzuzeigen, Queue im Feld **Objekttyp** und * im Feld **Name**, werden Sie eine Liste aller Warteschlangen angezeigt, die im System definiert sind. Sie können dann in dieser Liste die Warteschlange auswählen, mit der Sie arbeiten müssen.
- Starten Sie von der Eingangsanzeige aus, wo Sie das Objekt angeben, mit dem Sie arbeiten, indem Sie die Felder **Objekttyp** und **Name** ausfüllen.

Ändern einer Objektdefinition

Um eine Objektdefinition zu ändern, geben Sie Aktion 3 an, und drücken Sie die Eingabetaste, um die Anzeigen ALTER anzuzeigen. Diese Anzeigen sind sehr ähnlich wie die Anzeigen mit dem Befehl DEFINE. Sie können die Werte ändern, die Sie wünschen. Wenn Sie die Änderungen abgeschlossen haben, drücken Sie die Eingabetaste, um die Informationen an den Warteschlangenmanager zu senden.

Objektdefinition anzeigen

Wenn die Details eines Objekts angezeigt werden sollen, ohne sie ändern zu können, die Aktion 1 angeben und die Eingabetaste drücken, um die Anzeigen in der Anzeige DISPLAY anzuzeigen. Diese Anzeigen ähneln den DEFINIEREN-Anzeigen, mit der Ausnahme, dass die Felder nicht geändert werden können. Ändern Sie den Objektnamen so, dass die Details eines anderen Objekts angezeigt werden.

Objekt löschen

Wenn Sie ein Objekt löschen möchten, geben Sie die Aktion 4 (Verwalten) an, und die Aktion **Löschen** ist eine der Aktionen, die im daraufhin angezeigten Menü angezeigt werden. Wählen Sie die Aktion **Löschen** aus.

Sie werden aufgefordert, Ihre Anfrage zu bestätigen. Wenn Sie die Funktionstaste F3 oder F12 drücken, wird die Anforderung abgebrochen. Wenn Sie die Eingabetaste drücken, wird die Anforderung bestätigt und an den WS-Manager übergeben. Das von Ihnen angegebene Objekt wird dann gelöscht.

Anmerkung: Die meisten Typen von Kanalobjekten können nur gelöscht werden, wenn der Kanalinitiator gestartet wurde.

Mit Namenslisten arbeiten

Wenn Sie mit namelists arbeiten, gehen Sie wie bei anderen Objekten vor.

Für die Aktionen DEFINE LIKE oder ALTER die Funktionstaste F11 drücken, um Namen zur Liste hinzuzufügen oder um die Namen in der Liste zu ändern. Dies beinhaltet die Arbeit mit dem ISPF-Editor und alle normalen ISPF-Editierbefehle sind verfügbar. Geben Sie jeden Namen in der Namensliste in einer separaten Zeile ein.

Wenn Sie den ISPF-Editor auf diese Weise verwenden, sind die Funktionstasteinstellungen die normalen ISPF-Einstellungen und **nicht** die von den anderen Operations- und Steuerkonsolen verwendeten.

Wenn Sie in der Liste Kleinbuchstaben angeben müssen, geben Sie CAPS (OFF) in der Editorbefehlszeile an. Wenn Sie dies tun, werden alle Namenslisten, die Sie in der Zukunft bearbeiten, in Kleinbuchstaben angezeigt, bis Sie CAPS (ON) angeben.

Wenn Sie die Bearbeitung der Namensliste abgeschlossen haben, drücken Sie die Funktionstaste F3, um die ISPF-Editiersitzung zu beenden. Drücken Sie dann die Eingabetaste, um die Änderungen an den WS-Manager zu senden.

Achtung: Wenn Sie in dieser Phase nicht die Eingabetaste drücken, sondern die Funktionstaste F3 drücken, gehen alle Aktualisierungen verloren, die Sie eingegeben haben.

Implementieren des Systems unter Verwendung mehrerer Clusterübertragungswarteschlangen

Es macht keinen Unterschied, ob der Kanal in einem einzelnen Cluster oder in einem überlappenden Cluster verwendet wird. Wenn der Kanal ausgewählt und gestartet wird, wählt der Kanal die Übertragungswarteschlange in Abhängigkeit von den Definitionen aus.

Prozedur

- Wenn Sie die Option DEFCLXQ verwenden, lesen Sie den Abschnitt [„Verwenden der automatischen Definition von Warteschlangen und Wechseln“](#) auf Seite 368.
- Wenn Sie einen gestuften Ansatz verwenden, lesen Sie den Abschnitt [„Ändern der Clustersenderkanäle mit einem gestaffelten Ansatz“](#) auf Seite 369.

Verwenden der automatischen Definition von Warteschlangen und Wechseln

Verwenden Sie diese Option, wenn Sie die Verwendung der Option DEFCLXQ planen. Es wird eine Warteschlange für jeden Kanal erstellt, und jeder neue Kanal.

Vorgehensweise

1. Überprüfen Sie die Definition der Datei SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE, und ändern Sie die Attribute, falls erforderlich.
Diese Warteschlange ist in der Teildatei SCSQPROC(csq4insx) definiert.

2. Erstellen Sie die Modellwarteschlange 'SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE'.
3. Wenden Sie Sicherheitsrichtlinien für diese Modellwarteschlange und die SYSTEM.CLUSTER.TRANSMIT.**-Warteschlangen an.

Für z/OS benötigt die Benutzer-ID der gestarteten Task des Kanalinitiators folgende Berechtigungen:

- Steuerzugriff auf CLASS (MQADMIN) für

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelname
```

- Aktualisierungszugriff auf CLASS (MQQUEUE) für

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelname
```

Ändern der Clustersenderkanäle mit einem gestaffelten Ansatz

Verwenden Sie diese Option, wenn Sie planen, einen gestuften Ansatz zu verwenden. Mit diesem Prozess können Sie die neuen Clustersenderkanäle zu verschiedenen Zeiten an die Anforderungen Ihres Unternehmens anpassen.

Vorbereitende Schritte

- Identifizieren Sie Ihre Geschäftsanwendungen und welche Kanäle verwendet werden.
- Zeigen Sie für die von Ihnen verwendeten Warteschlangen die Cluster an, in der sie sich befinden.
- Zeigen Sie die Kanäle an, um die Verbindungsnamen, die Namen der fernen Warteschlangenmanager und die Cluster anzuzeigen, die vom Kanal unterstützt werden.

Informationen zu diesem Vorgang

- Erstellen Sie eine Übertragungswarteschlange. Unter z/OS könnten Sie überlegen, welche Seitengruppe Sie für die Warteschlange verwenden.
- Richten Sie die Sicherheitsrichtlinie für die Warteschlange ein.
- Ändern Sie die Warteschlangenüberwachung so, dass sie diesen Warteschlangennamen enthält.
- Entscheiden Sie, welche Kanäle diese Übertragungswarteschlange verwenden sollen. Die Kanäle sollten einen ähnlichen Namen haben, so dass die generischen Zeichen ' * ' in CLCHNAME den Kanal identifizieren.
- Wenn Sie bereit sind, die neue Funktion zu verwenden, ändern Sie die Übertragungswarteschlange, um den Namen der Kanäle anzugeben, die diese Übertragungswarteschlange verwenden sollen. Zum Beispiel CLUSTER1 .TOPARIS oder CLUSTER1 .* oder *.TOPARIS
- Kanäle starten

Vorgehensweise

1. Verwenden Sie den Befehl `DIS CLUSQMGR(xxxx) XMITQ`, um die Clustersenderkanäle anzuzeigen, die im Cluster definiert sind, wobei xxxx für den Namen des fernen Warteschlangenmanagers steht.
2. Richten Sie das Sicherheitsprofil für die Übertragungswarteschlange ein, und geben Sie den Warteschlangenzugriff auf den Kanalinitiator an.
3. Definieren Sie die Übertragungswarteschlange, die verwendet werden soll, und geben Sie USAGE (XMITQ) INDXTYPE (CORRELID) SHARE und CLCHNAME (*value*) an.

Die Benutzer-ID für die gestartete Kanalinitiator task benötigt den folgenden Zugriff:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel
update class(MQQUEUE ssid.SYSTEM.CLUSTER.TRANSMIT.channel
```

und die Benutzer-ID, die den Befehl SWITCH verwendet, benötigt den folgenden Zugriff:

```
alter cl(MQADMIN) ssid.QUEUE.queuname
```

4. Stoppen Sie die Kanäle und starten Sie sie erneut.

Die Kanaländerung tritt auf, wenn der Kanal mit einem MQSC-Befehl gestartet wird, oder Sie verwenden CSQUTIL. Sie können die Kanäle angeben, die mit dem SWITCH CHANNEL (*)STATUS von CSQUTIL erneut gestartet werden müssen.

Wenn beim Starten des Kanals Probleme auftreten, stoppen Sie den Kanal, beheben Sie die Probleme, und starten Sie den Kanal erneut.

Beachten Sie, dass Sie das Attribut CLCHNAME so oft ändern können, wie Sie es benötigen.

Der Wert von CLCHNAME wird beim Starten des Kanals verwendet, so dass Sie die CLCHNAME-Definition ändern können, während der Kanal weiterhin die Definitionen aus dem Zeitpunkt verwendet, zu dem sie gestartet wurde. Der Kanal verwendet die neue Definition, wenn er erneut gestartet wird.

Änderung unter z/OS rückgängig machen

Sie müssen einen Prozess zum Zurückstellen einer Änderung haben, wenn die Ergebnisse nicht wie erwartet sind.

Was kann schiefgehen?

Wenn die neue Übertragungswarteschlange nicht das ist, was Sie erwarten:

1. Überprüfen Sie den CLCHNAME wie erwartet.
2. Überprüfen Sie das Jobprotokoll, um zu prüfen, ob der Switchprozess beendet ist. Ist dies nicht der Fall, warten Sie, und überprüfen Sie die neue Übertragungswarteschlange des Kanals später.

Wenn Sie mehrere Clusterübertragungswarteschlangen verwenden, ist es wichtig, dass Sie die Definitionen der Übertragungswarteschlangen explizit entwerfen und eine komplizierte überlappende Konfiguration vermeiden. Auf diese Weise können Sie sicherstellen, dass, wenn Probleme auftreten, wieder die ursprünglichen Warteschlangen und die Konfiguration verwendet werden können.

Wenn bei der Verwendung einer anderen Übertragungswarteschlange Probleme auftreten, müssen Sie alle Probleme lösen, bevor Sie mit der Änderung fortfahren können.

Eine vorhandene Änderungsanforderung muss abgeschlossen sein, bevor eine neue Änderungsanforderung gestellt werden kann. Zum Beispiel:

1. Definieren Sie eine neue Übertragungswarteschlange mit einer maximalen Tiefe von 1, und es gibt 10 Nachrichten, die darauf warten, gesendet zu werden.
2. Ändern Sie die Übertragungswarteschlange, und geben Sie den Kanalnamen im Parameter CLCHNAME an.
3. Stoppen Sie den Kanal und starten Sie ihn erneut. Der Versuch, die Nachrichten zu verschieben, schlägt fehl und meldet die Probleme.
4. Ändern Sie den Parameter CLCHNAME in der Übertragungswarteschlange, die leer sein soll.
5. Stoppen Sie den Kanal und starten Sie ihn erneut. Der Kanal versucht weiterhin, die ursprüngliche Anforderung auszuführen, so dass der Kanal weiterhin die neue Übertragungswarteschlange verwendet.
6. Sie müssen die Probleme beheben und den Kanal erneut starten, damit das Versetzen von Nachrichten erfolgreich abgeschlossen wird.

Wenn der Kanal das nächste Mal erneut gestartet wird, nimmt er alle Änderungen auf. Wenn Sie CLCHNAME auf Leerzeichen gesetzt haben, verwendet der Kanal die angegebene Übertragungswarteschlange nicht.

In diesem Beispiel bedeutet die Änderung von CLCHNAME in der Übertragungswarteschlange zu Leerzeichen nicht unbedingt, dass der Kanal die Warteschlange SYSTEM.CLUSTER.TRANSMIT verwendet, da möglicherweise andere Übertragungswarteschlangen vorhanden sind, deren CLCHNAME-Parameter mit

dem Kanalnamen übereinstimmen. Beispielsweise kann ein generischer Name oder das WS-Manager-Attribut DEFCLXQ auf Kanal gesetzt werden, so dass der Kanal eine dynamische Warteschlange anstelle der Warteschlange SYSTEM.CLUSTER.TRANSMIT verwendet.

Programme zur Verwaltung von IBM MQ for z/OS schreiben

Sie können eigene Anwendungsprogramme schreiben, um einen WS-Manager zu verwalten. In diesem Thema können Sie die Anforderungen zum Schreiben eigener Verwaltungsprogramme verstehen.

Start der allgemeinen Informationen zur Programmierschnittstelle

Diese Gruppe von Themen enthält Hinweise und Anleitungen für die Ausgabe von IBM MQ-Befehlen aus einem IBM MQ-Anwendungsprogramm.

Anmerkung: In diesem Abschnitt werden die MQI-Aufrufe unter Verwendung der Programmiersprache C-Sprache beschrieben. Informationen zu typischen Aufrufen der Aufrufe in COBOL-, PL/I- und Assemblersprachen finden Sie im Handbuch [Funktionsaufrufe](#).

Informationen darüber, wie es funktioniert

In der Gliederung ist die Prozedur zum Absetzen von Befehlen aus einem Anwendungsprogramm wie folgt:

1. Erstellen Sie einen IBM MQ-Befehl in einem IBM MQ-Nachrichtentyp, der als *Anforderungsnachricht* bezeichnet wird. Der Befehl kann im MQSC- oder PCF-Format vorliegen.
2. Senden Sie diese Nachricht an eine spezielle Warteschlange mit dem Namen MQPUT, die als Eingabewarteschlange für Systembefehle bezeichnet wird. Der IBM MQ-Befehlsprozessor führt den Befehl aus.
3. Abrufen (verwenden Sie MQGET) Die Ergebnisse des Befehls als *Antwortnachrichten* in der Empfangswarteschlange für Antworten. Diese Nachrichten enthalten die Benutzernachrichten, die Sie benötigen, um zu ermitteln, ob Ihr Befehl erfolgreich war, und wenn er der war, was die Ergebnisse waren.

Dann ist es an Ihrem Anwendungsprogramm, die Ergebnisse zu verarbeiten.

Diese Gruppe enthält folgende Themen:

Warteschlangen für Verwaltungsprogramme vorbereiten

Verwaltungsprogramme erfordern eine Reihe vordefinierter Warteschlangen für Systembefehlseingabe und -empfangsantworten.

Dieser Abschnitt bezieht sich auf Befehle im MQSC-Format. Informationen zum Äquivalent in PCF finden Sie im Abschnitt [„IBM MQ Programmable Command Formats verwenden“](#) auf Seite 23.

Bevor Sie alle MQPUT - oder MQGET -Aufrufe absetzen können, müssen Sie zuerst die Warteschlangen definieren und dann öffnen, die Sie verwenden werden.

Eingabewarteschlange für Systembefehle definieren

Die Eingabewarteschlange des Systembefehls ist eine lokale Warteschlange mit dem Namen SYSTEM.COMMAND.INPUT. Die angegebene Initialisierungsdatei CSQINP2, thlqual.SCSQPROC (CSQ4INSG), enthält eine Standarddefinition für die Eingabewarteschlange des Systembefehls. Aus Gründen der Kompatibilität mit IBM MQ auf anderen Plattformen wird auch ein Alias dieser Warteschlange mit dem Namen SYSTEM.ADMIN.COMMAND.QUEUE bereitgestellt. Weitere Informationen finden Sie im Abschnitt [Mit IBM MQ gelieferte Beispieldefinitionen](#).

Antwortwarteschlange definieren

Sie müssen eine Empfangswarteschlange für Antworten definieren, um Antwortnachrichten vom IBM MQ-Befehlsprozessor zu empfangen. Es kann eine beliebige Warteschlange mit Attributen sein, die es ermöglichen, Antwortnachrichten in die Warteschlange zu stellen. Geben Sie jedoch für den normalen Betrieb die folgenden Attribute an:

- SYNTAX (NORMAL)
- NOTRIGGER (es sei denn, Ihre Anwendung verwendet Triggerung)

Vermeiden Sie es, persistente Nachrichten für Befehle zu verwenden. Wenn Sie dies jedoch auswählen, darf die Empfangswarteschlange für Antworten keine temporäre dynamische Warteschlange sein.

Die angegebene Initialisierungsdatei CSQINP2, thlqual.SCSQPROC (CSQ4INSG), enthält eine Definition für eine Modellwarteschlange mit dem Namen SYSTEM.COMMAND.REPLY.MODEL. Sie können dieses Modell verwenden, um eine dynamische Empfangswarteschlange für Antworten zu erstellen.

Anmerkung: Die vom Befehlsprozessor generierten Antworten können bis zu 15 000 Byte lang sein.

Wenn Sie eine permanente dynamische Warteschlange als Empfangswarteschlange für Antworten verwenden, sollte Ihre Anwendung die Zeit für alle Operationen PUT und GET zulassen, bevor versucht wird, die Warteschlange zu löschen. Andernfalls kann MQRC2055 (MQRC_Q_NOT_EMPTY) zurückgegeben werden. Ist dies der Fall, versuchen Sie es nach einigen Sekunden erneut, die Warteschlange zu löschen.

Eingabewarteschlange für Systembefehle öffnen

Bevor Sie die Eingabewarteschlange des Systembefehls öffnen können, muss Ihr Anwendungsprogramm mit dem Warteschlangenmanager verbunden sein. Verwenden Sie den MQI-Aufruf MQCONN oder MQCONNX , um dies zu tun.

Verwenden Sie dann den MQI-Aufruf MQOPEN , um die Eingabewarteschlange des Systembefehls zu öffnen. So verwenden Sie diesen Aufruf:

1. Setzen Sie den Parameter **Options** auf MQOO_OUTPUT.
2. Legen Sie die MQOD-Objektdeskriptorfelder wie folgt fest:

ObjectType

MQOT_Q (das Objekt ist eine Warteschlange)

ObjectName

SYSTEM.COMMAND.INPUT

ObjectQMgrName

Wenn Sie Ihre Anforderungsnachrichten an Ihren lokalen WS-Manager senden möchten, lassen Sie dieses Feld leer. Dies bedeutet, dass Ihre Befehle lokal verarbeitet werden.

Wenn Ihre IBM MQ-Befehle auf einem fernen Warteschlangenmanager verarbeitet werden sollen, geben Sie hier dessen Namen ein. Außerdem müssen Sie die richtigen Warteschlangen und Links einrichten, wie in [Verteilte Steuerung von Warteschlangen und Clustern](#) beschrieben.

Antwortwarteschlange öffnen

Um die Antworten von einem IBM MQ-Befehl abzurufen, müssen Sie eine Empfangswarteschlange für Antworten öffnen. Eine Möglichkeit dazu besteht darin, die Modellwarteschlange SYSTEM.COMMAND.REPLY.MODEL in einem Aufruf MQOPEN anzugeben, um eine permanente dynamische Warteschlange als Warteschlange für Antwortwarteschlange zu erstellen. So verwenden Sie diesen Aufruf:

1. Setzen Sie den Parameter **Options** auf MQOO_INPUT_SHARED.
2. Legen Sie die MQOD-Objektdeskriptorfelder wie folgt fest:

ObjectType

MQOT_Q (das Objekt ist eine Warteschlange)

ObjectName

Der Name der Empfangswarteschlange für Antworten. Wenn der von Ihnen angegebene Warteschlangenname der Name eines Modellwarteschlangenobjekts ist, erstellt der WS-Manager eine dynamische Warteschlange.

ObjectQMGrName

Lassen Sie dieses Feld leer, um Antworten auf Ihren lokalen WS-Manager zu erhalten.

DynamicQName

Geben Sie den Namen der dynamischen Warteschlange an, die erstellt werden soll.

 **Befehlsserver verwenden**

Der Befehlsserver ist eine IBM MQ-Komponente, die mit der Befehlsprozessorkomponente zusammenarbeitet. Sie können formatierte Nachrichten an den Befehlsserver senden, der die Nachrichten interpretiert, die Administrationsanforderungen ausführt und Antworten an Ihre Verwaltungsanwendung zurücksendet.

Der Befehlsserver liest Anforderungsnachrichten aus der Eingabewarteschlange des Systembefehls, überprüft sie und übergibt die gültigen Befehle als Befehle an den Befehlsprozessor. Der Befehlsprozessor verarbeitet die Befehle und gibt alle Antworten als Antwortnachrichten in die von Ihnen angegebene Empfangswarteschlange für Antworten ein. Die erste Antwortnachricht enthält die Benutzernachricht CSQN205I. Weitere Informationen finden Sie unter „[Interpretieren der Antwortnachrichten vom Befehlsserver](#)“ auf Seite 377. Der Befehlsserver verarbeitet auch Befehle des Kanalinitiators und der Gruppe mit gemeinsamer Warteschlange, unabhängig davon, wo sie ausgegeben wurden.

Den Warteschlangenmanager identifizieren, der Ihre Befehle verarbeitet

Der Warteschlangenmanager, der die von einem Verwaltungsprogramm abgesetzten Befehle verarbeitet, ist der Warteschlangenmanager, der Eigner der Eingabewarteschlange für Systembefehle ist, in die die Nachricht gestellt wird.

Starten des Befehlsservers

Normalerweise wird der Befehlsserver automatisch gestartet, wenn der WS-Manager gestartet wird. Es ist verfügbar, sobald die Nachricht CSQ9022I 'START QMGR' NORMAL COMPLETION vom Befehl START QMGR zurückgegeben wird. Der Befehlsserver wird gestoppt, wenn die Verbindung zu allen verbundenen Tasks während der Systemabschlussphase unterbrochen wurde.

Sie können den Befehlsserver mit den Befehlen START CMDSERV und STOP CMDSERV steuern. Um zu vermeiden, dass der Befehlsserver automatisch startet, wenn IBM MQ erneut gestartet wird, können Sie Ihren Initialisierungsdateien CSQINP1 oder CSQINP2 einen Befehl STOP CMDSERV hinzufügen. Dies wird jedoch nicht empfohlen, da dadurch die Verarbeitung von Befehlen des Kanalinitiators oder der Gruppe mit gemeinsamer Warteschlange verhindert wird.

Der Befehl STOP CMDSERV stoppt den Befehlsserver, sobald er die Verarbeitung der aktuellen Nachricht beendet hat, oder sofort, wenn keine Nachrichten verarbeitet werden.

Wenn der Befehlsserver durch einen Befehl STOP CMDSERV im Programm gestoppt wurde, können keine anderen Befehle aus dem Programm verarbeitet werden. Um den Befehlsserver erneut zu starten, müssen Sie den Befehl START CMDSERV über die z/OS-Konsole ausgeben.

Wenn Sie den Befehlsserver während der Ausführung des Warteschlangenmanagers stoppen und erneut starten, werden alle Nachrichten, die sich in der Eingabewarteschlange des Systembefehls befinden, wenn der Befehlsserver gestoppt wird, verarbeitet, wenn der Befehlsserver erneut gestartet wird. Wenn Sie den WS-Manager jedoch stoppen und erneut starten, nachdem der Befehlsserver gestoppt wurde, werden nur die persistenten Nachrichten in der Eingabewarteschlange des Systembefehls verarbeitet, wenn der Befehlsserver erneut gestartet wird. Alle nicht persistenten Nachrichten in der Eingabewarteschlange des Systembefehls gehen verloren.

Befehle an den Befehlsserver senden

Für jeden Befehl wird eine Nachricht mit dem Befehl erstellt und anschließend in die Eingabewarteschlange des Systembefehls gestellt.

Nachricht erstellen, die IBM MQ-Befehle einschließt

Sie können IBM MQ-Befehle in ein Anwendungsprogramm integrieren, indem Sie Anforderungsnachrichten erstellen, die die erforderlichen Befehle enthalten. Für jeden solchen Befehl:

1. Erstellen Sie einen Puffer, der eine Zeichenfolge enthält, die den Befehl darstellt.
2. Geben Sie einen MQPUT -Aufruf aus, der den Puffernamen im Parameter **buffer** des Aufrufs angibt.

Der einfachste Weg in C ist es, einen Puffer mit 'char' zu definieren. Beispiel:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

Wenn Sie einen Befehl erstellen, verwenden Sie eine auf null endende Zeichenfolge. Geben Sie keine Befehlspräfixzeichenfolge (CPF) am Anfang eines auf diese Weise definierten Befehls an. Dies bedeutet, dass Sie Ihre Befehlsscripts nicht ändern müssen, wenn Sie sie auf einem anderen Warteschlangenmanager ausführen wollen. Sie müssen jedoch berücksichtigen, dass ein CPF in allen Antwortnachrichten enthalten ist, die in die Warteschlange für Antwortnachrichten gestellt werden.

Der Befehlsserver setzt alle Kleinbuchstaben in Großbuchstaben um, sofern sie nicht in Anführungszeichen stehen.

Befehle können eine beliebige Länge von bis zu maximal 32 762 Zeichen haben.

Nachrichten in die Eingabewarteschlange des Systembefehls stellen

Verwenden Sie den Aufruf MQPUT , um Anforderungsnachrichten mit Befehlen in die Eingabewarteschlange des Systembefehls zu stellen. Geben Sie in diesem Aufruf den Namen der Warteschlange für Antwortwarteschlange an, die Sie bereits geöffnet haben.

Gehen Sie wie folgt vor, um den Aufruf MQPUT zu

1. Legen Sie die folgenden MQPUT -Parameter fest:

Hconn

Die Verbindungskennung, die vom Aufruf MQCONN oder MQCONNX zurückgegeben wird.

Hobj

Die Objektkennung, die vom Aufruf MQOPEN für die Eingabewarteschlange des Systembefehls zurückgegeben wird.

BufferLength

Die Länge des formatierten Befehls.

Buffer

Der Name des Puffers, der den Befehl enthält.

2. Legen Sie die folgenden MQMD-Felder fest:

MsgType

MQMT_REQUEST

Format

MQFMT_STRING oder MQFMT_NONE

Wenn Sie nicht dieselbe Codepage wie der Warteschlangenmanager verwenden, setzen Sie *CodedCharSetId* entsprechend und setzen MQFMT_STRING, so dass der Befehlsserver die Nachricht konvertieren kann. Geben Sie MQFMT_ADMIN nicht an, da dies dazu führt, dass Ihr Befehl als PCF interpretiert wird.

ReplyToQ

Name der Empfangswarteschlange für Antworten.

ReplyToQMgr

Wenn Antworten an den lokalen WS-Manager gesendet werden sollen, lassen Sie dieses Feld leer. Wenn Ihre IBM MQ-Befehle an einen fernen Warteschlangenmanager gesendet werden sollen, geben Sie hier dessen Namen ein. Außerdem müssen Sie die richtigen Warteschlangen und Links einrichten, wie in [Verteilte Steuerung von Warteschlangen und Clustern](#) beschrieben.

3. Legen Sie alle anderen MQMD-Felder nach Bedarf fest. Sie sollten normalerweise nicht persistente Nachrichten für Befehle verwenden.
4. Legen Sie bei Bedarf alle *PutMsgOpts* -Optionen fest.

Wenn Sie MQPMO_SYNCPOINT (die Standardeinstellung) angeben, müssen Sie den Aufruf MQPUT mit einem Synchronisationspunktaufruf ausführen.

Verwenden von MQPUT1 und der Eingabewarteschlange des Systembefehls

Wenn Sie nur eine Nachricht in die Eingabewarteschlange des Systembefehls einlegen möchten, können Sie den Aufruf **MQPUT1** verwenden. Dieser Aufruf kombiniert die Funktionen eines **MQOPEN** -Aufrufs, gefolgt von einem **MQPUT** einer Nachricht, gefolgt von einem **MQCLOSE** -Aufruf in einem Aufruf. Wenn Sie diesen Aufruf verwenden, ändern Sie die Parameter entsprechend. Weitere Informationen finden Sie im Abschnitt [Nachricht in eine Warteschlange mit dem MQPUT1-Aufruf einschalten](#).

Antworten auf Ihre Befehle abrufen

Der Befehlsserver sendet für jede Anforderungsnachricht, die er empfängt, eine Antwort an eine Antwortwarteschlange. Jede Verwaltungsanwendung muss die Antwortnachrichten empfangen und bearbeiten.

Wenn der Befehlsprozessor Ihre Befehle verarbeitet, werden alle Antwortnachrichten in die Warteschlange für Antwortnachrichten gestellt, die im Aufruf MQPUT angegeben ist. Der Befehlsserver sendet die Antwortnachrichten mit derselben Persistenz wie die Befehlsnachricht, die sie empfangen hat.

Warten auf Antwort

Verwenden Sie den Aufruf MQGET , um eine Antwort aus Ihrer Anforderungsnachricht abzurufen. Eine Anforderungsnachricht kann mehrere Antwortnachrichten erzeugen. Einzelheiten hierzu finden Sie im Abschnitt [„Interpretieren der Antwortnachrichten vom Befehlsserver“](#) auf Seite 377.

Sie können ein Zeitintervall angeben, das ein MQGET -Aufruf wartet, bis eine Antwortnachricht generiert wird. Wenn Sie keine Antwort erhalten, verwenden Sie die Prüfliste, die im Abschnitt [„Wenn Sie keine Antwort erhalten“](#) auf Seite 377 beginnt.

Gehen Sie wie folgt vor, um den Aufruf MQGET zu

1. Legen Sie die folgenden Parameter fest:

Hconn

Die Verbindungskennung, die vom Aufruf MQCONN oder MQCONNX zurückgegeben wird.

Hobj

Die Objektkennung, die vom Aufruf MQOPEN für die Empfangswarteschlange zurückgegeben wurde.

Buffer

Der Name des Bereichs, in dem die Antwort empfangen werden soll.

BufferLength

Die Länge des Puffers zum Empfangen der Antwort. Dieser Wert muss mindestens 80 Byte lang sein.

2. Um sicherzustellen, dass die Antworten nur über den von Ihnen ausgegebenen Befehl abgerufen werden, müssen Sie die entsprechenden Felder *MsgId* und *CorrelId* angeben. Diese hängen von den Berichtsoptionen MQMD_REPORT ab, die Sie im Aufruf MQPUT angegeben haben:

MQRO_NONE

Binäre Null, '00 ... 00' (24 Nullen).

MQRO_NEW_MSG_ID

Binäre Null, '00 ... 00' (24 Nullen).

Dies ist der Standardwert, wenn keine dieser Optionen angegeben wurde.

MQRO_PASS_MSG_ID

Die *MsgId* aus dem **MQPUT**.

MQRO_NONE

Der *MsgId* aus dem Aufruf MQPUT .

MQRO_COPY_MSG_ID_TO_CORREL_ID

Der *MsgId* aus dem Aufruf MQPUT .

Dies ist der Standardwert, wenn keine dieser Optionen angegeben wurde.

MQRO_PASS_CORREL_ID

Der *CorrelId* aus dem Aufruf MQPUT .

Weitere Informationen zu Berichtsoptionen finden Sie unter [Berichtsoptionen und Nachrichtenmarkierungen](#) .

3. Legen Sie die folgenden *GetMsgOpts* -Felder fest:

Options

MQGMO_WAIT

Wenn Sie nicht die gleiche Codepage wie der Warteschlangenmanager verwenden, setzen Sie MQGMO_CONVERT und setzen Sie *CodedCharSetId* entsprechend dem MQMD-Wert.

WaitInterval

Bei Antworten vom lokalen WS-Manager versuchen Sie es 5 Sekunden. Coded in Millisekunden, das wird 5 000. Bei Antworten von einem fernen Warteschlangenmanager und Kanalsteuerungs- und Statusbefehlen versuchen Sie es mit 30 Sekunden. Coded in Millisekunden, das wird 30 000.

Verdeckte Nachrichten

Wenn der Befehlsserver feststellt, dass eine Anforderungsnachricht nicht gültig ist, löscht er diese Nachricht und schreibt die Nachricht **CSQN205I** in die angegebene Warteschlange für Antwortnachrichten. Wenn keine Empfangswarteschlange für Antworten vorhanden ist, wird die Nachricht **CSQN205I** in die Warteschlange für dead-letter gestellt. Der Rückkehrcode in dieser Nachricht zeigt, warum die ursprüngliche Anforderungsnachricht nicht gültig war:

00D5020F Es ist nicht vom Typ MQMT_REQUEST.

00D50210 Sie hat die Länge null.

00D50212 Sie ist länger als 32 762 Byte.

00D50211 Sie enthält alle Leerzeichen.

00D5483E Es musste konvertiert werden, aber *Format* war nicht MQFMT_STRING.

Andere Siehe [Befehlsservercodes](#) .

Der Antwortnachrichtendeskriptor des Befehlsservers

Für jede Antwortnachricht werden die folgenden MQMD-Nachrichtendeskriptorfelder festgelegt:

MsgType MQMT_REPLY

Feedback MQFB_NONE

Encoding MQENC_NATIVE

Priority Wie für den MQMD in der Nachricht, die Sie ausgegeben haben.

*Persisten-
ce* Wie für den MQMD in der Nachricht, die Sie ausgegeben haben.

CorrelId Hängt von den MQPUT -Berichtsoptionen ab.

ReplyToQ Keine.

Der Befehlsserver setzt das *Options* -Feld der MQPMO-Struktur auf MQPMO_NO_SYNCPOINT. Dies bedeutet, dass Sie die Antworten, wie sie erstellt wurden, und nicht als Gruppe am nächsten Synchronisationspunkt abrufen können.

Interpretieren der Antwortnachrichten vom Befehlsserver

Jede Anforderungsnachricht, die von IBM MQ ordnungsgemäß verarbeitet wird, erzeugt mindestens zwei Antwortnachrichten. Jede Antwortnachricht enthält eine einzelne IBM MQ-Benutzernachricht.

Die Länge einer Antwort hängt von dem Befehl ab, der ausgegeben wurde. Die längste Antwort, die Sie erhalten können, ist von einer DISPLAY NAMELIST, die bis zu 15 000 Byte lang sein kann.

Die erste Benutzernachricht (CSQN205I) enthält immer:

- Die Anzahl der Antworten (dezimal), die Sie als Zähler in einer Schleife verwenden können, um den Rest der Antworten zu erhalten. Der Zähler schließt diese erste Nachricht ein.
- Der Rückkehrcode des Vorprozessors des Befehls.
- Ein Ursachencode, der der Ursachencode des Befehlsprozessors ist.

Diese Nachricht enthält keine CPF-Nachricht.

Beispiel:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
```

Das Feld COUNT ist 8 Byte lang und hat rechtsbündig angegeben. Sie beginnt immer bei Position 18, also unmittelbar nach COUNT=. Das Feld RETURN ist 8 Byte lang im Hexadezimalformat und befindet sich unmittelbar hinter RETURN= an Position 35. Das Feld REASON ist 8 Byte lang im Hexadezimalformat und befindet sich unmittelbar hinter REASON= an Position 52.

Wenn der Wert für RETURN= 00000000 und der Wert REASON= 00000004 lautet, ist die Gruppe der Antwortnachrichten unvollständig. Nachdem Sie die Antworten, die durch die Nachricht CSQN205I angegeben wurden, abgerufen haben, geben Sie einen weiteren MQGET -Aufruf aus, um auf eine weitere Gruppe von Antworten zu warten. Die erste Nachricht in der nächsten Gruppe von Antworten ist erneut CSQN205I, die angibt, wie viele Antworten es gibt, und ob noch mehr zu kommen sind.

Weitere Informationen zu den einzelnen Nachrichten finden Sie in der [IBM MQ for z/OS -Nachrichten, -Beendigungscode und -Ursachencodes-Dokumentation](#).

Wenn Sie eine nicht englische Sprachfunktion verwenden, unterscheidet sich der Text und das Layout der Antworten von denen, die hier angezeigt werden. Die Größe und die Position der Zähler- und Rückkehrcodes in der Nachricht CSQN205I sind jedoch identisch.

Wenn Sie keine Antwort erhalten

Es gibt eine Reihe von Schritten, die Sie ausführen können, wenn Sie keine Antwort auf die Anforderung an den Befehlsserver erhalten.

Wenn Sie keine Antwort auf Ihre Anforderungsnachricht erhalten, arbeiten Sie mit dieser Prüfliste:

- Ist der Befehlsserver aktiv?

- Ist der *waitInterval* lang genug?
- Sind die Eingabe- und Antwortwarteschlangen des Systembefehls korrekt definiert?
- Wurden die MQOPEN -Aufrufe erfolgreich in diese Warteschlangen aufrufen?
- Sind sowohl die Systembefehlseingabe als auch die Antwortwarteschlange für MQPUT - und MQGET -Aufrufe aktiviert?
- Haben Sie die Erhöhung der MAXDEPTH- und MAXMSGL-Attribute Ihrer Warteschlangen in Betracht gezogen?
- Verwenden Sie die Felder *CorrelId* und *MsgId* ordnungsgemäß?
- Ist der WS-Manager noch aktiv?
- Wurde der Befehl ordnungsgemäß erstellt?
- Sind alle Ihre fernen Verbindungen definiert und ordnungsgemäß funktioniert?
- Wurden die MQPUT -Aufrufe korrekt definiert?
- Wurde die Warteschlange für Antwortantworten als temporäre dynamische Warteschlange anstelle einer permanenten dynamischen Warteschlange definiert? (Wenn die Anforderungsnachricht persistent ist, müssen Sie für die Antwort eine permanente dynamische Warteschlange verwenden.)

Wenn der Befehlsserver Antworten generiert, diese jedoch nicht in die Warteschlange für Antwortwarteschlange schreiben kann, die Sie angeben, schreibt er sie in die Warteschlange für den Deadbuchstaben.

Befehle mit MGCRE übergeben

Mit der entsprechenden Berechtigung kann ein Anwendungsprogramm über eine z/OS-Serviceroutine Anforderungen an mehrere Warteschlangenmanager stellen.

Wenn Sie über die richtige Berechtigung verfügen, können Sie IBM MQ-Befehle aus Ihrem Programm über den z/OS-Service MGCRE (SVC 34) an mehrere Warteschlangenmanager übergeben. Der Wert des CPF identifiziert den jeweiligen WS-Manager, an den der Befehl gerichtet ist. Informationen zu CPFs finden Sie in den Abschnitten [Benutzer-IDs für die Befehlssicherheit und die Sicherheit der Befehlsressourcen](#) und „[Warteschlangenmanagerbefehle unter z/OS ausgeben](#)“ auf Seite 349.

Wenn Sie MGCRE verwenden, können Sie ein Command and Response Token (CART) verwenden, um die direkten Antworten auf den Befehl zu erhalten.

Beispiele für Befehle und ihre Antworten

Verwenden Sie dieses Thema als eine Reihe von Befehlsbeispielen für den Befehlsserver und die Antworten von dem Befehlsserver.

Im Folgenden finden Sie einige Beispiele für Befehle, die in IBM MQ-Nachrichten integriert werden können, sowie die Benutzernachrichten, bei denen es sich um die Antworten handelt. Wenn nicht anders angegeben, ist jede Zeile der Antwort eine separate Nachricht.

- [Nachrichten von einem DEFINE-Befehl](#)
- [Nachrichten von einem DELETE-Befehl](#)
- [Nachrichten von DISPLAY-Befehlen](#)
- [Nachrichten von Befehlen mit CMDSCOPE](#)
- [Nachrichten von Befehlen, die Befehle mit CMDSCOPE generieren](#)

Nachrichten von einem Befehl DEFINE

Der folgende Befehl:

```
DEFINE QLOCAL(Q1)
```

erzeugt folgende Nachrichten:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000  
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

Diese Antwortnachrichten werden bei normalem Abschluss erstellt.

Nachrichten von einem Befehl DELETE

Der folgende Befehl:

```
DELETE QLOCAL(Q2)
```

erzeugt folgende Nachrichten:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000008  
CSQM125I +CSQ1 CSQMUQLC QLOCAL (Q2) QSGDISP(QMGR) WAS NOT FOUND  
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D4002'  
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

Diese Nachrichten weisen darauf hin, dass eine lokale Warteschlange mit dem Namen Q2 nicht vorhanden ist.

Nachrichten von DISPLAY-Befehlen

In den folgenden Beispielen werden die Antworten von einigen DISPLAY-Befehlen angezeigt.

Namen der Warteschlange für dead-Mail-Nachrichten ermitteln

Wenn Sie den Namen der Warteschlange für dead-letter für einen Warteschlangenmanager ermitteln möchten, setzen Sie diesen Befehl von einem Anwendungsprogramm ab:

```
DISPLAY QMGR DEADQ
```

Es werden die folgenden drei Benutzernachrichten zurückgegeben, aus denen Sie den erforderlichen Namen extrahieren können:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000  
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE )  
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION
```

Nachrichten aus dem Befehl DISPLAY QUEUE

Die folgenden Beispiele zeigen, wie die Ergebnisse eines Befehls von den Attributen abhängen, die in diesem Befehl angegeben sind.

Beispiel 1

Sie definieren eine lokale Warteschlange mit dem folgenden Befehl:

```
DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE
```

Wenn Sie den folgenden Befehl aus einem Anwendungsprogramm absetzen:

```
DISPLAY QUEUE(Q1) SHARE GET DESCR
```

werden die folgenden drei Benutzernachrichten zurückgegeben:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1) TYPE(
QLOCAL ) QSGDISP(QMGR )
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMDMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Anmerkung: Die zweite Nachricht (CSQM401I) wird hier in vier Zeilen dargestellt.

Beispiel 2

Zwei Warteschlangen haben Namen, die mit dem Buchstaben A beginnen:

- A1 ist eine lokale Warteschlange mit dem Attribut PUT, das auf DISABLED gesetzt ist.
- A2 ist eine ferne Warteschlange mit dem Attribut PUT, die auf ENABLED gesetzt ist.

Wenn Sie den folgenden Befehl aus einem Anwendungsprogramm absetzen:

```
DISPLAY QUEUE(A*) PUT
```

werden die folgenden vier Benutzernachrichten zurückgegeben:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1) TYPE(
QLOCAL ) QSGDISP(QMGR )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2) TYPE(
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMDMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Anmerkung: Die zweite und dritte Nachricht (CSQM401I und CSQM406I) werden hier dargestellt, wobei drei und zwei Zeilen belegt sind.

Nachrichten vom Befehl DISPLAY NAMELIST

Sie definieren eine Namensliste mit dem Befehl:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

Wenn Sie den folgenden Befehl aus einem Anwendungsprogramm absetzen:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

werden die folgenden drei Benutzernachrichten zurückgegeben:

```
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1
          ) QS
GDISP(QMGR  ) NAMCOUNT(    2) NAMES(Q1
,SAMPLE_QUEUE
)
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY NAMELIST' NORMAL COMPLETION
```

Anmerkung: Die zweite Nachricht (CSQM407I) wird hier dargestellt, die drei Zeilen belegt.

Nachrichten von Befehlen mit CMDSCOPE

In den folgenden Beispielen werden die Antworten von Befehlen angezeigt, die mit dem Attribut CMDSCOPE eingegeben wurden.

Nachrichten des Befehls ALTER PROCESS

Der folgende Befehl:

```
ALT PRO(V4) CMDSCOPE(*)
```

erzeugt die folgenden Nachrichten:

```
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'ALT PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I  COUNT=    5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ26
CSQM125I !MQ26 CSQMMSGP PROCESS(V4) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E !MQ26 CSQMMSGP FAILURE REASON CODE X'00D44002'
CSQ9023E !MQ26 CSQMMSGP ' ALT PRO' ABNORMAL COMPLETION
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' ALT PRO' NORMAL COMPLETION
CSQN205I  COUNT=    2, RETURN=0000000C, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

Diese Nachrichten geben an, dass der Befehl in Warteschlangenmanager MQ25 eingegeben und an zwei WS-Manager (MQ25 und MQ26) gesendet wurde. Der Befehl war in MQ25 erfolgreich, aber die Prozessdefinition war in MQ26 nicht vorhanden, so dass der Befehl auf diesem WS-Manager fehlgeschlagen ist.

Nachrichten aus dem Befehl DISPLAY PROCESS

Der folgende Befehl:

```
DIS PRO(V*) CMDSCOPE(*)
```

erzeugt die folgenden Nachrichten:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ26
CSQM408I !MQ26 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ26 PROCESS(V3) QSGDISP(QMGR)
CSQ9022I !MQ26 CSQMDRTS ' DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 7, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ25
CSQM408I !MQ25 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ25 PROCESS(V2) QSGDISP(GROUP)
CSQM408I !MQ25 PROCESS(V3) QSGDISP(QMGR)
CSQM408I !MQ25 PROCESS(V4) QSGDISP(QMGR)
CSQ9022I !MQ25 CSQMDRTS ' DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS PRO' command for CMDSCOPE(*) normal completion

```

Diese Nachrichten geben an, dass der Befehl in Warteschlangenmanager MQ25 eingegeben und an zwei WS-Manager (MQ25 und MQ26) gesendet wurde. Es werden Informationen zu allen Prozessen in jedem WS-Manager angezeigt, deren Namen mit dem Buchstaben V beginnen.

Nachrichten aus dem Befehl DISPLAY CHSTATUS

Der folgende Befehl:

```
DIS CHS(VT) CMDSCOPE(*)
```

erzeugt die folgenden Nachrichten:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS CHS' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ25
CSQM422I !MQ25 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ25 CSQXDRTS ' DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ26
CSQM422I !MQ26 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ26 CSQXDRTS ' DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS CHS' command for CMDSCOPE(*) normal completion

```

Diese Nachrichten geben an, dass der Befehl in Warteschlangenmanager MQ25 eingegeben und an zwei WS-Manager (MQ25 und MQ26) gesendet wurde. Es werden Informationen zum Kanalstatus auf den einzelnen Warteschlangenmanagern angezeigt.

Nachrichten aus dem Befehl STOP CHANNEL

Der folgende Befehl:

```
STOP CHL(VT) CMDSCOPE(*)
```

erzeugt folgende Nachrichten:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'STOP CHL' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQM134I !MQ25 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion

```

Diese Nachrichten geben an, dass der Befehl in Warteschlangenmanager MQ25 eingegeben und an zwei WS-Manager (MQ25 und MQ26) gesendet wurde. Der Kanal VT wurde auf jedem WS-Manager gestoppt.

Nachrichten von Befehlen, die Befehle mit CMDSCOPE generieren

Der folgende Befehl:

```
DEF PRO(V2) QSGDISP(GROUP)
```

erzeugt folgende Nachrichten:

```

CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQM122I !MQ25 CSQMMSGP ' DEF PRO' COMPLETED FOR QSGDISP(GROUP)
CSQN138I !MQ25 'DEFINE PRO' command generated for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ26
CSQ9022I !MQ26 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DEFINE PRO' command for CMDSCOPE(*) normal completion

```

Diese Nachrichten geben an, dass der Befehl in den WS-Manager MQ25 eingegeben wurde. Wenn das Objekt im gemeinsam genutzten Repository erstellt wurde, wurde ein anderer Befehl generiert und an alle aktiven Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange (MQ25 und MQ26) gesendet.

z/OS

IBM MQ-Ressourcen unter z/OS verwalten

Über die Links in diesem Abschnitt erfahren Sie, wie die von IBM MQ for z/OS verwendeten Ressourcen verwaltet werden, z. B. Protokolldateien, Dateigruppen, Seitengruppen, Pufferpools und Coupling-Facility-Strukturen.

Unter den folgenden Links finden Sie Details zu den verschiedenen Verwaltungstasks, die Sie möglicherweise bei der Verwendung von IBM MQ for z/OS ausführen müssen:

- „[Protokolle verwalten](#)“ auf Seite 384
- „[Bootstrap Data Set \(BSDS\) verwalten](#)“ auf Seite 393
- „[Seitengruppen verwalten](#)“ auf Seite 402

- [„Seitengruppen sichern und wiederherstellen“ auf Seite 409](#)
- [„Vorgehensweise zum Sichern und Zurückschreiben von Warteschlangen mit CSQUTIL“ auf Seite 413](#)
- [„Pufferpools verwalten“ auf Seite 414](#)
- [„Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzte Warteschlangen unter z/OS verwalten“ auf Seite 415](#)

Zugehörige Konzepte

[„IBM MQ for z/OS verwalten“ auf Seite 337](#)

Die Verwaltung von Warteschlangenmanagern und zugeordneten Ressourcen umfasst die Tasks, die Sie häufig ausführen, um diese Ressourcen zu aktivieren und zu verwalten. Wählen Sie die Methode aus, die Sie bevorzugen, um die Warteschlangenmanager und die zugehörigen Ressourcen zu verwalten.

[„Befehle für IBM MQ for z/OS ausgeben“ auf Seite 338](#)

Sie können Warteschlangenmanager mit IBM MQ-Scriptbefehlen (MQSC) im Stapel- oder Dialogmodus steuern.

[„Wiederherstellung und Neustart unter z/OS“ auf Seite 426](#)

In diesem Abschnitt werden die Wiederherstellungs- und Neustartmechanismen von IBM MQ beschrieben.

Zugehörige Verweise

[„Dienstprogramme von IBM MQ for z/OS“ auf Seite 346](#)

IBM MQ for z/OS stellt eine Reihe von Dienstprogrammen bereit, die Sie zur Unterstützung bei der Systemverwaltung verwenden können.

Zugehörige Informationen

[IBM MQ for z/OS - Konzepte](#)

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

[Warteschlangenmanager unter z/OS erstellen](#)

[Programmierbare Befehlsformate-Referenz](#)

[Referenzinformationen zu MQSC](#)

[IBM MQ for z/OS-Dienstprogramme verwenden](#)

z/OS Protokolle verwalten

In diesem Abschnitt wird beschrieben, wie IBM MQ-Protokolldateien verwaltet werden, einschließlich Protokollarchivierungsprozess, Verwendung der Protokollsatzkomprimierung, Protokollsatzwiederherstellung und Drucken von Protokollsätzen.

In diesem Abschnitt werden die Aufgaben bei der Verwaltung der IBM MQ-Protokolle beschrieben. Er enthält folgende Abschnitte:

z/OS Protokolle mit dem Befehl ARCHIVE LOG archivieren

Ein berechtigter Operator kann die aktuellen aktiven IBM MQ-Protokolldateien bei Bedarf mit dem Befehl ARCHIVE LOG archivieren.

Wenn Sie den Befehl ARCHIVE LOG ausgeben, schneidet IBM MQ die aktuellen aktiven Protokolldateien ab, führt dann einen asynchronen Auslagerungsprozess durch und aktualisiert die Bootstrap-Datei (BSDS) mit einem Datensatz des Auslagerungsprozesses.

Der Befehl ARCHIVE LOG verfügt über eine Option MODE (QUIESCE). Mit dieser Option werden IBM MQ-Jobs und -Benutzer nach einem Commitpunkt in den Quiescemodus versetzt und der sich daraus ergebende Konsistenzzustand wird im aktiven Protokoll erfasst, bevor es ausgelagert wird.

Sie können die Option MODE (QUIESCE) verwenden, wenn Sie eine Sicherungsstrategie für die Wiederherstellung vor Ort planen. Es erstellt einen systemweiten Konsistenzpunkt, der die Anzahl der Dateninkonsistenzen minimiert, wenn das Archivprotokoll während der Wiederherstellung mit der aktuellen Kopie der Sicherungsseite verwendet wird. Beispiel:


```
ARCHIVE LOG MODE(QUIESCE)
```

Wenn Sie den Befehl ARCHIVE LOG ohne Angabe eines Parameters TIME ausgeben, wird für den Quiesce-Zeitraum standardmäßig der Wert des Parameters QUIESCE des Makros CSQ6ARVP verwendet. Wenn die Zeit, die für den Abschluss des Befehls ARCHIVE LOG MODE (QUIESCE) erforderlich ist, kleiner ist als die angegebene Zeit, wird der Befehl erfolgreich ausgeführt. Andernfalls schlägt der Befehl fehl, wenn der Zeitraum abläuft. Sie können den Zeitraum explizit angeben, indem Sie die Option TIME verwenden. Beispiel:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

Mit diesem Befehl wird ein Quiesce-Zeitraum von bis zu 60 Sekunden angegeben, bevor die ARCHIVE LOG-Verarbeitung erfolgt.

Achtung: Die Verwendung der Option TIME kann in einer zeitkritischen Situation zu einer erheblichen Störung der IBM MQ-Verfügbarkeit für alle Jobs und Benutzer, die IBM MQ-Ressourcen verwenden, führen.

Standardmäßig wird der Befehl asynchron von der Zeit verarbeitet, zu der Sie den Befehl übergeben haben. (Um den Befehl synchron mit anderen IBM MQ-Befehlen zu verarbeiten, verwenden Sie die Option WAIT (YES) mit QUIESCE, beachten Sie jedoch, dass die z/OS-Konsole während der gesamten QUIESCE-Phase für IBM MQ-Befehlseingaben gesperrt ist.)

Während der Quiesce-Periode:

- Jobs und Benutzer auf dem Warteschlangenmanager können die Commitverarbeitung durchlaufen, werden jedoch ausgesetzt, wenn sie versuchen, eine IBM MQ-Ressource nach dem Commit zu aktualisieren.
- Jobs und Benutzer, die nur Daten lesen, können betroffen sein, da sie möglicherweise auf Sperren warten, die von Jobs oder Benutzern gehalten werden, die ausgesetzt wurden.
- Neue Tasks können gestartet werden, aber sie können keine Daten aktualisieren.

Die Ausgabe des Befehls DISPLAY LOG verwendet die Nachricht CSQV400I, um anzuzeigen, dass ein Quiesce in Kraft ist. Beispiel:

```

CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter   Initial value   SET value
-----
INBUFF      60
OUTBUFF     400
MAXRTU      2
MAXARCH     2
TWOACTV     YES
TWOARCH     YES
TWOBSDS     YES
OFFLOAD     YES
MAXCNOFF    0
WRTHRS      20
DEALLCT     0
COMPLOG     NONE
ZHYWRITE    NO
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full PPRC DSName
  1      68 NO VICY.CSQ1.LOGCOPY1.DS01
  2      68 NO VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2014-04-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION

```



Achtung: zHyperWrite ist in IBM MQ 9.0 nicht aktiviert, daher ist *NO* der einzige zulässige Wert.

Wenn alle Aktualisierungen in den Quiescemodus versetzt werden, wird der Quiesce-Protokollsatz im BSDS mit dem Datum und der Uhrzeit aktualisiert, zu dem die aktiven Protokolldateien abgeschnitten wurden, und mit der zuletzt geschriebenen RBA in den aktuellen aktiven Protokolldatengruppen. IBM MQ schneidet die aktuellen aktiven Protokolldateien ab, wechselt zu den nächsten verfügbaren aktiven Protokolldateien und gibt die Nachricht CSQJ311I aus, die angibt, dass der Auslagerungsprozess gestartet wurde.

Wenn Aktualisierungen nicht vor Ablauf der Quiescephase in den Quiescemodus versetzt werden können, gibt IBM MQ die Nachricht CSQJ317I aus, und die ARCHIVE LOG-Verarbeitung wird beendet. Die aktuellen aktiven Protokolldateien werden nicht abgeschnitten oder auf die nächsten verfügbaren Protokolldatensätze umgestellt, und der Ausladeprozess wird nicht gestartet.

Unabhängig davon, ob das Quiesce erfolgreich war oder nicht, werden alle ausgesetzten Benutzer und Jobs wiederaufgenommen, und IBM MQ gibt die Nachricht CSQJ312I aus, die angibt, dass das Quiesce beendet ist und die Aktivität fortgesetzt wird.

Wenn ARCHIVE LOG ausgegeben wird, wenn das aktuelle aktive Protokoll die letzte verfügbare aktive Protokolldatei ist, wird der Befehl nicht verarbeitet und IBM MQ gibt folgende Nachricht aus:

```

CSQJ319I - csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING
WILL BE TERMINATED

```

Wenn ARCHIVE LOG ausgegeben wird, während bereits ein anderer ARCHIVE LOG-Befehl ausgeführt wird, wird der neue Befehl nicht verarbeitet, und IBM MQ gibt folgende Nachricht aus:

```

CSQJ318I - ARCHIVE LOG COMMAND ALREADY IN PROGRESS

```

Informationen zu den Nachrichten, die während der Archivierung ausgegeben werden, finden Sie im Abschnitt [Nachrichten für IBM MQ for z/OS](#).

Protokollarchivierungsprozess nach einem Fehler erneut starten

Wenn während des Protokollarchivierungsprozesses ein Problem auftritt (z. B. ein Problem mit Zuordnung oder Bandladevorgang), kann die Archivierung des aktiven Protokolls ausgesetzt werden. Sie können den Archivierungsprozess abbrechen und ihn mit dem Befehl ARCHIVE LOG CANCEL OFFLOAD erneut starten. Mit diesem Befehl wird die momentan in Bearbeitung vorliegende Offload-Verarbeitung abgebrochen und der Archivierungsprozess erneut gestartet. Sie beginnt mit der ältesten Protokolldatei, die nicht archiviert wurde, und setzt alle aktiven Protokollatengruppen fort, die ausgelagert werden müssen. Alle Protokollarchivierungsoperationen, die ausgesetzt wurden, werden erneut gestartet.

Verwenden Sie diesen Befehl nur, wenn Sie sicher sind, dass die aktuelle Task zum Protokollarchivieren nicht mehr funktioniert, oder wenn Sie einen vorherigen fehlgeschlagenen Versuch erneut starten wollen. Dies liegt daran, dass der Befehl eine abnormale Beendigung der Auslastungstask verursachen kann, die zu einem Speicherauszug führen kann.

Archivierung und Protokollierung steuern

Mit den Makros CSQ6LOGP, CSQ6ARVP und CSQ6SYSP können Sie die Komprimierung, das Drucken, die Archivierung, die Wiederherstellung und die Protokollierung steuern. Beachten Sie, dass Änderungen an privaten Objekten in IBM MQ-Protokollen protokolliert werden. Änderungen an GROUP-Objekten (z. B. gemeinsam genutzte eingehende Kanäle) werden ebenfalls protokolliert, da die Definitionen in der Gruppe weitergegeben und lokal gespeichert werden.

Viele Aspekte der Archivierung und Protokollierung werden von Parametern gesteuert, die mit den Makros CSQ6LOGP, CSQ6ARVP und CSQ6SYSP des Systemparametermoduls festgelegt werden, wenn der Warteschlangenmanager angepasst wird. Ausführliche Informationen zu diesen Makros finden Sie im Abschnitt [Task 17: Tailor your system parameter module](#).

Einige dieser Parameter können geändert werden, während ein Warteschlangenmanager mit den IBM MQ-MQSC-Befehlen SET LOG, SET SYSTEM und SET ARCHIVE ausgeführt wird. Sie werden in [Tabelle 26](#) auf Seite 387 gezeigt:

SET, Befehl	Parameter
LOG	WRTHRSH, MAXARCH, DEALLCT, MAXRTU, COMPLOG
Archiv	Alle
SYSTEM	LOGLOAD

Sie können die Einstellungen aller Parameter mit den Befehlen MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) und [DISPLAY SYSTEM](#) anzeigen. Diese Befehle zeigen auch Statusinformationen zum Archivieren und Protokollieren an.

Protokollkomprimierung steuern

Sie können die Komprimierung von Protokollsätzen unter Verwendung von aktivieren und inaktivieren.

- Die Befehle SET und DISPLAY LOG in MQSC; siehe [MQSC-Befehle](#)
- Die PCF-Schnittstelle wird angezeigt. Siehe [„Einführung in IBM MQ Programmable Command Formats“ auf Seite 22.](#)
- Verwendung des Makros CSQ6LOGP im Systemparametermodul; siehe [CSQ6LOGP verwenden](#)

Protokollsätze drucken

Sie können Protokollsätze mit Hilfe des Dienstprogramms CSQ1LOGP extrahieren und drucken. Anweisungen hierzu finden Sie im Abschnitt [Protokolldruckdienstprogramm](#).

Protokolle wiederherstellen

Normalerweise müssen Sie die IBM MQ-Protokolle nicht sichern und wiederherstellen, insbesondere wenn Sie die doppelte Protokollierung verwenden. In seltenen Fällen, wie z. B. bei einem E/A-Fehler in einem Protokoll, müssen Sie die Protokolle möglicherweise wiederherstellen. Verwenden Sie die Zugriffsmethodenservices, um den Datensatz zu löschen und neu zu definieren, und kopieren Sie anschließend das entsprechende doppelte Protokoll in das Datenset.

Löschen der Archivprotokolldateien

Sie können Ihre Archivierungsprotokolldatensätze löschen und die Protokolle automatisch oder manuell löschen.

Sie müssen genügend Protokolldaten zur Ausführung der UOWs (Unit of Work Recovery, Seitengruppe) aufbewahren, wenn eine Seitengruppe verloren gegangen ist, oder die CF-Struktur wiederverwenden, wenn eine CF-Struktur verloren geht. Verwerfen Sie keine Archivprotokolldateien, die möglicherweise für die Wiederherstellung erforderlich sind. Wenn Sie diese Archivprotokolldateien löschen, können Sie möglicherweise nicht die erforderlichen Wiederherstellungsoperationen ausführen.

Wenn Sie bestätigt haben, dass Ihre Archivierungsprotokolldatensätze verworfen werden können, können Sie dies auf eine der folgenden Arten tun:

- [Automatisches Löschen der Archivprotokolldatei](#)
- [Archivprotokolldateien manuell löschen](#)

Löschen automatischer Archivierungsprotokollaten

Sie können ein DASD-System oder ein Bandverwaltungssystem verwenden, um Archivprotokolldateien automatisch zu löschen. Der Aufbewahrungszeitraum für IBM MQ-Archivprotokolldateien wird im Feld ARCRETN für den Aufbewahrungszeitraum im Installationsmakro CSQ6ARVP angegeben (weitere Informationen finden Sie im Abschnitt [CSQ6ARVP verwenden](#)).

Der Standardwert für den Aufbewahrungszeitraum gibt an, dass die Archivprotokolle für 9999 Tage (das Maximum) aufbewahrt werden sollen.

Wichtig: Sie können den Aufbewahrungszeitraum ändern, aber Sie müssen sicherstellen, dass Sie die Anzahl der Sicherungszyklen, für die Sie geplant haben, aufnehmen können.

IBM MQ verwendet den Wert für den Aufbewahrungszeitraum als Wert für den JCL-Parameter RETPD, wenn Archivprotokolldateien erstellt werden.

Der vom MVS/DFP -Speicherverwaltungssystem (SMS) festgelegte Aufbewahrungszeitraum kann durch diesen IBM MQ -Parameter überschrieben werden. In der Regel wird der Aufbewahrungszeitraum auf den kleineren der von IBM MQ und SMS angegebenen Werte festgelegt. Der Speicheradministrator und der IBM MQ-Administrator müssen sich auf einen Wert für den Aufbewahrungszeitraum einigen, der für IBM MQ geeignet ist.

Anmerkung: IBM MQ verfügt nicht über eine automatisierte Methode zum Löschen von Informationen über Archivprotokolldateien aus der Bootstrap-Datei (BSDS), da einige Bandverwaltungssysteme externe manuelle Überschreibungen von Aufbewahrungszeiträumen bereitstellen. Daher können Informationen über eine Archivprotokolldatei noch lange im BSDS vorhanden sein, nachdem der Aufbewahrungszeitraum für die Dateigruppe abgelaufen ist und die Datei durch das Bandverwaltungssystem zerkratzt wurde.

Umgekehrt kann die maximale Anzahl von Archivierungsprotokolldatensätzen überschritten worden sein, und die Daten aus dem BSDS wurden möglicherweise gelöscht, bevor die Datei ihr Verfallsdatum erreicht hat.

Wenn Archivierungsprotokolldatensätze automatisch gelöscht werden, denken Sie daran, dass die Operation die Liste der Archivprotokolle im BSDS nicht aktualisiert. Sie können die BSDS mit dem Dienstprogramm zum Ändern des Protokollbestands aktualisieren, wie im Abschnitt „[BSDS ändern](#)“ auf Seite 395 beschrieben. Die Aktualisierung ist nicht von wesentlicher Bedeutung. Durch das Aufzeichnen alter Archivprotokolle wird Speicherbereich im BSDS vergeudet, aber es wird kein anderer Schaden angerichtet.

Archivierungsprotokolldatensätze manuell löschen

Sie müssen alle Protokollsätze so weit zurückhalten, wie die niedrigste RBA, die in den Nachrichten CSQI024I und CSQI025I angegeben ist. Dieser RBA wird mit dem Befehl DISPLAY USAGE abgerufen, den Sie bei der Erstellung eines Wiederherstellungspunkts unter Verwendung von [Methode 1: Gesamtsicherung](#) ausgegeben haben.

Lesen Sie Wiederherstellungspunkt für nicht gemeinsam genutzte Ressourcen erstellen before discarding any logs.

Lokalisieren und Löschen der Archivprotokolldateien

Nachdem Sie die für die Wiederherstellung erforderliche Mindestprotokoll-RBA festgelegt haben, können Sie Archivprotokolldateien finden, die nur frühere Protokollsätze enthalten, indem Sie die folgende Prozedur ausführen:

1. Verwenden Sie das Dienstprogramm zum Drucken der Protokollzuordnung, um den Inhalt des BSDS zu drucken. Ein Beispiel für die Ausgabe finden Sie im Abschnitt [Dienstprogramm für Druckprotokollzuordnung](#).
2. Suchen Sie die Abschnitte der Ausgabe mit dem Titel ARCHIVE LOG COPY n DATA SETS. Wenn Sie die doppelte Protokollierung verwenden, gibt es zwei Abschnitte. Die Spalten mit der Bezeichnung STARTRBA und ENDRBA zeigen den Bereich der RBAs an, die in jedem Datenträger enthalten sind. Suchen Sie die Datenträger mit Bereichen, die den minimalen RBA enthalten, den Sie mit den Nachrichten CSQI024I und CSQI025I gefunden haben. Dies sind die frühesten Datenträger, die Sie behalten müssen. Wenn Sie die zweifache Protokollierung verwenden, gibt es zwei solcher Datenträger.

Wenn keine Datenträger über einen entsprechenden Bereich verfügen, gilt einer der folgenden Fälle:

- Die minimale RBA wurde noch nicht archiviert, und Sie können alle Archivierungsprotokolldatenträger verwerfen.
- Die Liste der Archivierungsprotokolldatenträger im BSDS, die umgebrochen ist, wenn die Anzahl der Datenträger die durch den Parameter MAXARCH des Makros CSQ6LOGP zulässige Anzahl überschritten hat. Wenn der BSDS keinen Archivierungsprotokolldatenträger registriert, kann dieser Datenträger nicht für die Wiederherstellung verwendet werden. Daher sollten Sie in Erwägung ziehen, dem BSDS Informationen zu vorhandenen Datenträgern hinzuzufügen. Anweisungen hierzu erhalten Sie im Abschnitt [„Änderungen für Archivprotokolle“](#) auf Seite 397.

Sie sollten auch den Wert von MAXARCH erhöhen. Weitere Informationen finden Sie im Abschnitt [CSQ6LOGP verwenden](#).

3. Löschen Sie alle Archivierungsprotokolldatensätze oder -datenträger mit einem ENDRBA-Wert, der kleiner als der STARTRBA-Wert des frühesten Datenträgers ist, den Sie behalten möchten. Wenn Sie die doppelte Protokollierung verwenden, löschen Sie beide Kopien.

Da BSDS-Einträge umgebrochen werden, sind möglicherweise die ersten Einträge im BSDS-Archivprotokollabschnitt aktueller als die Einträge am Ende. Sehen Sie sich die Kombination von Datum und Zeit an und vergleichen Sie deren Alter. Es darf nicht angenommen werden, dass alle Einträge vor dem Eintrag für das Archivprotokoll gelöscht werden können, das den Mindestwert für LOGRBA enthält.

Löschen Sie die Datensätze. Befinden sich die Archive auf Band, löschen Sie die Bänder. Wenn sie sich auf einer DASD-Einheit befinden, führen Sie ein z/OS-Dienstprogramm aus, um die einzelnen Dateien zu löschen. Wenn anschließend im BSDS nur vorhandene Archivdatenträger aufgelistet werden sollen, verwenden Sie das Dienstprogramm zum Ändern des Protokollbestands (CSQJU003), um die Einträge der gelöschten Datenträger zu löschen. Ein Beispiel finden Sie unter „Änderungen für Archivprotokolle“ auf Seite 397.

z/OS Auswirkungen des Protokollabschranke

Transaktionen mit langer Laufzeit können die UO- (Unit of Work-) Protokollsätze verursachen, die sich über Protokoll Datensätze erstrecken IBM MQ handhabt dieses Szenario durch Verwendung der Protokollverzögerung (Shunting), wobei die Protokollsätze verschoben werden, um die Anzahl der aufbewahrten Protokoll Daten und die Zeit für den Neustart des Warteschlangenmanagers zu optimieren.

Wenn eine Arbeitseinheit als lang angesehen wird, wird eine Darstellung jedes Protokollsatzes weiter unten in das Protokoll geschrieben. Dies wird als *Protokollshundemodus* bezeichnet. Sie wird in [Protokoll-dateien](#) ausführlicher beschrieben.

Der WS-Manager verwendet diese nicht mehr verwendeten Protokollsätze anstelle der Originale nach einem Fehler, um die Integrität der Arbeitseinheit sicherzustellen. Dies hat zwei Vorteile:

- die Menge der Protokoll Daten, die für die Koordination der Arbeitseinheit aufbewahrt werden müssen, wird verringert.
- weniger Protokoll Daten müssen beim Neustart des Warteschlangenmanagers durchlaufen werden, damit der WS-Manager schneller erneut gestartet wird.

Protokollsätze ohne Daten enthalten keine ausreichenden Informationen für Operationen zur Datenträgerwiederherstellung.

Daten, die im Protokoll festgehalten werden, werden für zwei unterschiedliche Zwecke verwendet: die Datenträgerwiederherstellung und die Einheit für die Koordination der Arbeit. Wenn ein Datenträgerfehler auftritt, der sich auf eine CF-Struktur oder eine Seitengruppe auswirkt, kann der Warteschlangenmanager den Datenträger bis zum Fehlerpunkt wiederherstellen, indem er eine vorherige Kopie zurückschreiben und diese mit den im Protokoll enthaltenen Daten aktualisieren kann. Die persistente Aktivität, die in einer Arbeitseinheit ausgeführt wird, wird im Protokoll aufgezeichnet, so dass im Falle eines Fehlers entweder zurückgesetzt werden kann oder dass Sperren für geänderte Ressourcen wiederhergestellt werden können. Die Anzahl der Protokoll Daten, die Sie beibehalten müssen, um die Wiederherstellung des Warteschlangenmanagers zu aktivieren, ist von diesen beiden Elementen betroffen.

Bei einer Datenträgerwiederherstellung müssen Sie genügend Protokoll Daten aufbewahren, um die Datenträgerwiederherstellung von mindestens der letzten Datenträgerkopie durchführen zu können, und um eine Sicherungskopie zu erhalten. (Ihr Standort kann die Möglichkeit zur Wiederherstellung nach älteren Sicherungen festlegen.) Für die Einheit der Arbeitsintegrität müssen Sie die Protokoll Daten für die ältesten in unbestärkten Arbeitseinheiten oder in unbestärkten Arbeitseinheiten aufbewahren.

Um Sie bei der Verwaltung des Systems zu unterstützen, erkennt der Warteschlangenmanager die alten Arbeitseinheiten in jedem Protokollarchiv und meldet sie in den Nachrichten CSQJ160 und CSQJ161. Eine interne Task liest die Arbeitsprotokoll Daten für diese alten Arbeitseinheiten und schreibt sie in einer weiteren prägnanteren Form in die aktuelle Position im Protokoll. Nachricht CSQR026 gibt an, wann dies geschehen ist. Der MQSC-Befehl DISPLAY USAGE TYPE (DATASET) kann Ihnen auch helfen, die Aufbewahrung von Protokoll Daten zu verwalten. Der Befehl meldet 3 Teile der Wiederherstellungsinformationen:

1. wie viel des Protokolls für die Wiederherstellung der Arbeitseinheit aufbewahrt werden muss
2. wie viel des Protokolls für die Datenträgerwiederherstellung von Seitengruppen aufbewahrt werden muss
3. wie viel des Protokolls für einen Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange für die Datenträgerwiederherstellung von CF-Strukturen aufbewahrt werden muss

Für jede dieser Daten wird versucht, die ältesten Protokoll Daten zuzuordnen, die in einem Datensatz erforderlich sind. Beim Starten und Stoppen neuer Arbeitseinheiten würde erwartet, dass (1) an eine aktuellere Position im Protokoll verschoben wird. Erfolgt keine Verschiebung, geben die Nachrichten für

die Arbeitseinheit mit langer Laufzeit die Warnung aus, dass es ein Problem gibt. (2) bezieht sich auf die Wiederherstellung der Datenträgerwiederherstellung, wenn der WS-Manager jetzt heruntergefahren und erneut gestartet werden sollte. Es ist nicht bekannt, wann Sie Ihre Seitengruppen zuletzt gesichert haben, oder welche Sicherung Sie verwenden müssen, wenn ein Seitengruppen-Fehler aufgetreten ist. Sie wird normalerweise während der Prüfpunktverarbeitung in eine aktuellere Position im Protokoll verschoben, da die Änderungen in den Pufferpools in die Seitengruppen geschrieben werden. In (3) hat der Warteschlangenmanager keine Kenntnisse über Backups von CF-Strukturen, die in diesem Warteschlangenmanager oder in anderen Warteschlangenmanagern innerhalb der Gruppe mit gemeinsamer Warteschlange erstellt wurden. Für die Wiederherstellung der CF-Struktur ist allerdings eine Zusammenführung von Protokolldaten aus allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange erforderlich, die seit dem letzten Backup mit der CF-Struktur kommuniziert haben. Das bedeutet, dass die Protokolldaten nicht von einer RBA ermittelt werden, die auf verschiedenen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange unterschiedliche Werte hat, sondern von einer Protokollsatzfolgennummer (Log Record Sequence Number, LRSN), die auf der Zeitmarke basiert und daher in der gesamten Gruppe mit gemeinsamer Warteschlange angewendet werden kann. Die Daten werden normalerweise in eine aktuellere Position im Protokoll verschoben, wenn BACKUP CFSTRUCT-Befehle auf diesem oder anderen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange ausgeführt werden.

Protokoll des Warteschlangenmanagers zurücksetzen

In diesem Thema wird erläutert, wie Sie das Protokoll des Warteschlangenmanagers zurücksetzen können.

Sie dürfen nicht zulassen, dass das WS-Manager-Protokoll RBA vom Ende des Protokoll-RBA-Bereichs auf 0 umgebrochen wird, da dies zu einem Ausfall des Warteschlangenmanagers führt und alle persistenten Daten nicht wiederherstellbar werden. Das Ende des Protokolls RBA ist entweder der Wert FFFFFFFF (wenn 6-Byte-RBAs wie im Gebrauch) oder FFFFFFFFFF (wenn 8-Byte-RBAs im Gebrauch sind).

Der Warteschlangenmanager gibt die Nachrichten [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031D](#) und [CSQJ032E](#) aus, um anzuzeigen, dass der verwendete Protokollbereich von Bedeutung ist und Sie Maßnahmen ergreifen sollten, um einen ungeplanten Ausfall zu vermeiden.

Der WS-Manager wird mit dem Ursachencode [00D10257](#) beendet, wenn der RBA-Wert FFF800000000 erreicht (wenn 6-Byte-Protokoll-RBAs verwendet werden) oder FFFFFFFC00000000 (wenn 8-Byte-Protokoll-RBAs verwendet werden).

Wenn 6-Byte-Protokoll-RBAs im Gebrauch sind, ist es in Betracht zu ziehen, den Warteschlangenmanager so zu konvertieren, dass er 8-Byte-Protokoll-RBAs verwendet, anstatt das Protokoll des Warteschlangenmanagers nach dem Prozess zurückzusetzen, der im Abschnitt [Relative Byteadresse des größeren Protokolls implementieren](#) beschrieben ist. Das Konvertieren eines Warteschlangenmanagers in die Verwendung von 8-Byte-Protokoll-RBAs erfordert einen kürzeren Ausfall als das Zurücksetzen des Protokolls und erhöht den Zeitraum, bevor Sie das Protokoll zurücksetzen müssen.

Die Nachricht [CSQJ034I](#), die während der Initialisierung des Warteschlangenmanagers ausgegeben wird, gibt das Ende des ProtokollrBA-Bereichs für den Warteschlangenmanager als konfiguriert an und kann verwendet werden, um zu bestimmen, ob die RBAs mit 6 Byte oder 8 Byte im Gebrauch sind.

Gehen Sie wie folgt vor, um das Protokoll des Warteschlangenmanagers zurückzusetzen:

1. Beheben Sie alle unaufgelösten Arbeitseinheiten. Die Anzahl der unaufgelösten Arbeitseinheiten wird beim Start des Warteschlangenmanagers in der Nachricht [CSQR005I](#) als INDOUBT-Zähler angezeigt. An jedem Prüfpunkt und beim Beenden des Warteschlangenmanagers setzt der Warteschlangenmanager den Befehl automatisch ab.

DISPLAY CONN(*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED), um Informationen zu unaufgelösten Arbeitseinheiten bereitzustellen.

Informationen zum Auflösen von Wiederherstellungseinheiten finden Sie in [Wie unbestätigte Einheiten der Wiederherstellung aufgelöst werden](#). Die letzte Möglichkeit besteht darin, den MQSC-Befehl **RESOLVE INDOUBT** zu verwenden, um unbestätigte Arbeitseinheiten mit Wiederherstellung manuell aufzulösen.

2. Fahren Sie den WS-Manager sauber herunter.

Sie können entweder **STOP QMGR** oder **STOP QMGR MODE(FORCE)** verwenden, da beide Befehle alle geänderten Seiten aus Pufferpools in die Seitengruppen löschen.

3. Wenn ein Warteschlangenmanager Teil einer Gruppe mit gemeinsamer Warteschlange ist, erstellen Sie CFSTRUCT-Backups auf anderen Warteschlangenmanagern für alle Strukturen in der Gruppe mit gemeinsamer Warteschlange. :NONE.

4. Definieren Sie mit CSQJU003 neue Protokolle und BSDS (siehe [Dienstprogramm zum Ändern des Protokollbestands](#)), um weitere Informationen zur Verwendung des Dienstprogramms zum Ändern des Protokollbestands zu erhalten.

5. Führen Sie **CSQUTIL RESETPAGE** für alle Seitengruppen für diesen Warteschlangenmanager aus (weitere Informationen zur Verwendung dieser Funktion finden Sie unter [Seite kopieren und Protokoll zurücksetzen](#)). Beachten Sie, dass die Seitengruppe RBAs unabhängig voneinander zurückgesetzt werden kann, sodass mehrere gleichzeitig ablaufende Jobs (z. B. eine pro Seitengruppe) übergeben werden können, um die abgelaufene Zeit für diesen Schritt zu reduzieren.

6. Starten Sie den Warteschlangenmanager erneut.

Zugehörige Konzepte

„Relative Byteadresse für größere Protokolle implementieren“ auf Seite 392

Vor IBM MQ 8.0 verwendete IBM MQ for z/OS eine 6-Byte-Protokoll-RBA, um die Position von Daten im Protokoll anzugeben. Ab IBM MQ 8.0 kann die Protokoll-RBA 8 Byte lang sein, was den Zeitraum bis zur erforderlichen Zurücksetzung des Protokolls vergrößert.


Relative Byteadresse für größere Protokolle implementieren


Vor IBM MQ 8.0 verwendete IBM MQ for z/OS eine 6-Byte-Protokoll-RBA, um die Position von Daten im Protokoll anzugeben. Ab IBM MQ 8.0 kann die Protokoll-RBA 8 Byte lang sein, was den Zeitraum bis zur erforderlichen Zurücksetzung des Protokolls vergrößert.

Diese neue Funktion muss explizit aktiviert werden. Hinweise zur Aktivierung von 8-Byte-Protokoll-RBA finden Sie unter [Planung für die Erhöhung des maximalen adressierbaren Protokollbereichs](#).

Führen Sie die folgenden Anweisungen in der angegebenen Reihenfolge aus, um 8-Byte-Protokoll-RBAs auf einem einzelnen IBM MQ for z/OS-Warteschlangenmanager zu aktivieren:

1. Aktivieren Sie neue Funktionen von IBM MQ 8.0 mithilfe von [OPMODE](#).

 Bei Warteschlangenmanagern in einer Gruppe mit gemeinsamer Warteschlange müssen Sie keinen Ausfall der gesamten Gruppe mit gemeinsamer Warteschlange verwenden. Sie können jeden WS-Manager wiederum stoppen, ihn für `OPMODE = (NEWFUNC, 800)` oder `OPMODE = (NEWFUNC, 900)` aktivieren und erneut starten.

 Wenn alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange mit `OPMODE=(NEWFUNC,800)` oder `OPMODE=(NEWFUNC,900)` ausgeführt werden, führen Sie die folgenden Schritte für jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange aus, bis alle Warteschlangenmanager mit dem neuen BSDS ausgeführt werden.

2. Zuordnen neuer BSDS-Datensätze mit ähnlichen Attributen zu den aktuellen BSDS. Sie können Muster CSQ4BSDS anpassen und jede irrelevante Anweisung löschen, oder Sie können Ihre vorhandene JCL verwenden, aber den BSDS-Namen in etwa `++HLQ++ . NEW . BSDS01` ändern.

Anmerkungen:

- Überprüfen Sie die Attribute Ihres neuen BSDS. Das einzige Attribut, das geändert werden kann, ist die Größe des BSDS.
- Das neue BSDS enthält mehr Daten, die das aktuelle BSDS enthält. Daher müssen Sie sicherstellen, dass die neuen Dateien mit ausreichend verfügbarer Speicherplatz zugeordnet werden. Empfohlene Werte für ein neues Bootstrap-Dataset finden Sie im Abschnitt [Protokollierungsumgebung planen](#) und dessen Unterthemen.

3. Fahren Sie den WS-Manager sauber herunter.

4. Führen Sie das BSDS-Konvertierungsdienstprogramm (CSQJUCNV) aus, um die vorhandenen BSDS-Dateien in die neuen BSDS-Dateien zu konvertieren. Dies dauert in der Regel einige Sekunden.

Ihre vorhandenen BSDS werden während dieses Prozesses nicht geändert, und Sie können dies für die Initialisierung des Warteschlangenmanagers im Falle einer nicht erfolgreichen Konvertierung verwenden.

5. Benennen Sie die aktuellen BSDS um, und die neuen BSDS werden zum aktuellen BSDS, so dass die neuen Dateien beim nächsten Neustart des Warteschlangenmanagers verwendet werden. Sie können den Befehl "DFSMS Access Method Services ALTER" verwenden, z. B.:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

Stellen Sie sicher, dass Sie auch Befehle absetzen, um sowohl die Daten-als auch die Indexabschnitte des VSAM-Clusters umzubenennen.

6. Starten Sie den Warteschlangenmanager erneut. Es sollte in der gleichen Zeit gestartet werden, wie dies bei Verwendung von 6-Byte-Protokoll-RBA geschehen wäre.


Wenn der WS-Manager aufgrund eines Fehlers beim Zugriff auf die konvertierten BSDS nicht erneut gestartet werden kann, versuchen Sie, die Ursache des Fehlers zu ermitteln, beheben Sie das Problem und wiederholen Sie die Operation. Falls erforderlich, wenden Sie sich an Ihr IBM Support Center.

Falls erforderlich, kann die Änderung an dieser Stelle durch folgende Punkte zurückgesetzt werden:

- a. Umbenennen des aktuellen BSDS in das neue BSDS.
- b. Die alte BSDS wird umbenannt, um die aktuelle BSDS zu werden.
- c. Der WS-Manager wird erneut gestartet.

Nachdem der WS-Manager erfolgreich mit den konvertierten BSDS erneut gestartet wurde, versuchen Sie nicht, den Warteschlangenmanager mit dem alten BSDS zu starten.

7. Die Nachricht CSQJ034I wird während der Initialisierung des Warteschlangenmanagers ausgegeben, um das Ende des Protokoll-RBA für den Warteschlangenmanager als konfiguriert anzuzeigen. Bestätigen Sie, dass das Ende des angezeigten RBA-Bereichs FFFFFFFFFFFFFFFF angezeigt wird. Dies weist darauf hin, dass die RBA mit 8 Byte im Gebrauch ist.

Anmerkung:  Damit ein 8-Byte-Protokoll-RBA auf einem neuen Warteschlangenmanager von IBM MQ 9.0 aktiviert werden kann, müssen Sie zunächst ein leeres BSDS der Version 1 erstellen und dieses als Eingabe für das BSDS-Konvertierungsdienstprogramm verwenden, um ein BSDS der Version 2 zu erstellen. Informationen dazu, wie Sie diesen Prozess ausführen, finden Sie im Abschnitt [Bootstrap- und Protokollgruppendaten erstellen](#).

Zugehörige Informationen

[Planung für die Erhöhung des maximal adressierbaren Protokollbereichs](#)

[Größeres Protokoll-Relative Byteadresse](#)

[Das BSDS-Konvertierungsdienstprogramm \(CSQJUCNV\)](#)

Bootstrap Data Set (BSDS) verwalten

Die Bootstrap-Datei (BSDS) wird zum Versehen von Protokoll Datensätzen und Protokoll Sätzen verwendet. In diesem Thema wird erläutert, wie Sie das BSDS untersuchen, ändern und wiederherstellen können.

Weitere Informationen finden Sie unter [Das Bootstrap-Dataset](#).

In diesem Abschnitt werden die Tasks beschrieben, die für die Verwaltung der Bootstrap-Datei ausgeführt werden müssen. Er enthält folgende Abschnitte:

- „Herausfinden, was das BSDS enthält“ auf Seite [394](#)
- „BSDS ändern“ auf Seite [395](#)

- „[BSDS wiederherstellen](#)“ auf Seite 399

Herausfinden, was das BSDS enthält

Sie können das Dienstprogramm für die Druckprotokollzuordnung (CSQJU004) verwenden, um den Inhalt des BSDS zu untersuchen.

Das Dienstprogramm für die Druckprotokollzuordnung (CSQJU004) ist ein Stapeldienstprogramm, in dem die im BSDS gespeicherten Informationen aufgelistet werden. Anweisungen zum Ausführen dieser Datei finden Sie im Abschnitt [Dienstprogramm für Druckprotokollzuordnung](#).

Das BSDS enthält:

- [Zeitmarken](#)
- [Status der aktiven Protokolldatei](#)

Zeitmarken im BSDS

Die Ausgabe des Dienstprogramms für die Druckprotokollzuordnung zeigt die Zeitmarken an, die zum Aufzeichnen des Datums und der Uhrzeit der verschiedenen Systemereignisse verwendet werden, die im BSDS gespeichert sind.

Die folgenden Zeitmarken werden in den Kopfzeilenabschnitt des Berichts eingefügt:

SYSTEMZEITMARKTAMP

Gibt den Zeitpunkt (Datum und Uhrzeit) an, zu dem das BSDS zuletzt aktualisiert wurde. Der BSDS-Zeitstempel kann aktualisiert werden, wenn:

- Der WS-Manager wird gestartet.
- Der Schreibschwellenwert wird während der Protokollschreibaktivitäten erreicht. Abhängig von der Anzahl der Ausgabepuffer, die Sie angegeben haben, und der Systemaktivitätsrate kann das BSDS mehrmals pro Sekunde aktualisiert werden, oder es kann nicht für mehrere Sekunden, Minuten oder sogar Stunden aktualisiert werden. Details zum Schreibschwellenwert finden Sie im Parameter WRTHRSH des Makros CSQ6LOGP im Abschnitt [CSQ6LOGP verwenden](#).
- IBM MQ wechselt aufgrund eines Fehlers vom normalen BSDS-Dualmodus in den BSDS-Einzelmodus. Dies kann auftreten, wenn eine Anforderung zum Abrufen, Einfügen, Punkt-Punkt, Aktualisieren oder Löschen eines BSDS-Datensatzes nicht erfolgreich ist. Wenn dieser Fehler auftritt, aktualisiert IBM MQ die Zeitmarke in der verbleibenden BSDS, um eine Zeitmarkenabweichung von der inaktivierten BSDS zu erzwingen.

UTILITY TIMESTAMP

Der Zeitpunkt (Datum und Uhrzeit), zu dem der Inhalt des BSDS durch das Dienstprogramm zum Ändern des Protokollbestands geändert wurde (CSQJU003).

Die folgenden Zeitmarken sind in den aktiven und Archivierungsprotokoll Daten enthalten, die Teil des Berichts sind:

Aktives Protokolldatum

Das Datum, an dem der aktive Protokolleintrag im BSDS erstellt wurde, d. a. wenn der Befehl CSQJU003 NEWLOG ausgeführt wurde.

Aktive Protokollzeit

Die Uhrzeit, zu der der aktive Protokolleintrag im BSDS erstellt wurde, d. a. wenn CSQJU003 NEWLOG ausgeführt wurde.

Datum des Archivprotokolls

Das Datum, an dem der Archivprotokolleintrag im BSDS erstellt wurde, d. a. wenn CSQJU003 NEWLOG ausgeführt wurde oder das Archiv selbst ausgeführt wurde.

Archivprotokollzeit

Die Uhrzeit, zu der der Archivprotokolleintrag im BSDS erstellt wurde, d. a. wenn CSQJU003 NEWLOG ausgeführt wurde oder das Archiv selbst ausgeführt wurde.

Status der aktiven Protokolldatei

Der BSDS zeichnet den Status einer aktiven Protokolldatei auf, wie eine der folgenden Angaben:

neu

Die Datei wurde definiert, aber nie von IBM MQ verwendet, oder das Protokoll wurde an einem Punkt vor der ersten Verwendung der Datei abgeschnitten. In beiden Fällen werden die Daten zum Starten und Beenden von RBA-Werten auf null zurückgesetzt.

REUSABLE

Entweder wurde die Datei definiert, aber nie von IBM MQ verwendet, oder die Datei wurde ausgelagert. In der Ausgabe der Druckprotokollzuordnung ist der Start-RBA-Wert für die letzte REUSABLE-Datei gleich dem Start-RBA-Wert des letzten Datensatzes der Archivprotokolldatei.

NICHT REUSABLE

Der Datensatz enthält Datensätze, die nicht ausgelagert wurden.

STOPPED

Der Offload-Prozessor hat beim Lesen eines Datensatzes einen Fehler festgestellt, und dieser Datensatz konnte nicht von der anderen Kopie des aktiven Protokolls abgerufen werden.

TRUNCATED

Entweder:

- Es ist ein E/A-Fehler aufgetreten und IBM MQ hat das Schreiben in diese Datei gestoppt. Die aktive Protokolldatei wird ausgelagert, beginnend mit dem Start-RBA und dem Fortfahren bis zum letzten gültigen Satzsegment in der abgeschnittenen aktiven Protokolldatei. Der RBA des letzten gültigen Satzsegments ist niedriger als der letzte RBA des aktiven Protokoll Datensatzes. Die Protokollierung wird auf die nächste verfügbare aktive Protokolldatei umgeschaltet und setzt die ununterbrochene Verarbeitung fort.

oder

- Es wurde eine ARCHIVE LOG-Funktion aufgerufen, die das aktive Protokoll abgeschnitten hat.

Der Status wird in der Ausgabe des Dienstprogramms für die Druckprotokollzuordnung angezeigt.

BSDS ändern

Sie müssen keine speziellen Schritte ausführen, um die BSDS mit Datensätzen von Protokollierungsereignissen zu aktualisieren, da IBM MQ dies automatisch tut.

Es kann jedoch sein, dass Sie die BSDS ändern, wenn Sie eine der folgenden Schritte ausführen:

- Fügen Sie weitere aktive Protokolldateien hinzu.
- Kopieren Sie aktive Protokoll Datensätze in neu zugeordnete Dateien, z. B., wenn Sie größere aktive Protokollzuordnungen bereitstellen.
- Versetzen von Protokoll Datensätzen auf andere Einheiten.
- Beschädigtes BSDS wiederherstellen.
- Veraltete Archivierungsprotokoll Datensätze verwerfen.

Sie können das BSDS ändern, indem Sie das Dienstprogramm zum Ändern des Protokollbestands ausführen (CSQJU003). Führen Sie dieses Dienstprogramm nur aus, wenn der Warteschlangenmanager inaktiv ist, oder Sie können inkonsistente Ergebnisse erhalten. Die Aktion des Dienstprogramms wird durch Anweisungen in der SYSIN-Datei gesteuert. In diesem Abschnitt werden mehrere Beispiele gezeigt. Die vollständigen Anweisungen finden Sie im Abschnitt [Dienstprogramm zum Ändern des Protokollbestands](#).

Sie können eine aktive Protokolldatei nur dann kopieren, wenn der Warteschlangenmanager inaktiv ist, da IBM MQ die aktiven Protokolldateien beim Start des Warteschlangenmanagers als exklusiv (DISP=OLD) zuordnet.

Änderungen für aktive Protokolle

In diesem Thema wird erläutert, wie Sie die aktiven Protokolle mit Hilfe des BSDS ändern können.

Mit Hilfe des Dienstprogramms zum Ändern des Protokolls können Sie Einträge im BSDS für aktive Protokolle hinzufügen, aus- und aufzeichnen. Hier werden nur Beispiele angezeigt. Ersetzen Sie die Dateinamen, die mit den Namen angezeigt werden, die Sie verwenden möchten. Weitere Informationen zum Dienstprogramm finden Sie im Abschnitt Dienstprogramm zum Ändern des Protokollbestands.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- Datensatzeinträge zu BSDS hinzufügen
- Informationen zur aktiven Protokolldatei aus BSDS löschen
- Informationen zur Protokolldatei in BSDS aufzeichnen
- Aktives Protokoll vergrößern
- Verwendung von CSQJUFMT

Satzeinträge zum BSDS hinzufügen

Wenn ein aktives Protokoll als "gestoppt" markiert wurde, wird es nicht für die Protokollierung wiederverwendet; es wird jedoch weiterhin zum Lesen verwendet. Verwenden Sie die Services der Zugriffsmethode, um neue aktive Protokolldateien zu definieren, und verwenden Sie anschließend das Dienstprogramm zum Ändern des Protokollbestands, um die neuen Dateien im BSDS zu registrieren. Verwenden Sie zum Beispiel:

```
NEWLOG DSNNAME=MQM111.LOGCOPY1.DS10,COPY1  
NEWLOG DSNNAME=MQM111.LOGCOPY2.DS10,COPY2
```

Wenn Sie den Inhalt einer alten aktiven Protokolldatei in die neue Datei kopieren, können Sie auch den RBA-Bereich und die Anfangs- und Endzeitmarken in der Funktion NEWLOG angeben.

Informationen über die aktive Protokolldatei aus dem BSDS löschen

Wenn Sie Informationen zu einer aktiven Protokolldatei aus dem BSDS löschen möchten, können Sie Folgendes verwenden:

```
DELETE DSNNAME=MQM111.LOGCOPY1.DS99  
DELETE DSNNAME=MQM111.LOGCOPY2.DS99
```

Informationen über die Protokolldatei aufzeichnen, die im BSDS festgelegt ist

Gehen Sie wie folgt vor, um Informationen zu einer vorhandenen aktiven Protokolldatei im BSDS aufzuzeichnen:

```
NEWLOG DSNNAME=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,  
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

Möglicherweise müssen Sie einen Datensatz einfügen, der diese Art von Informationen in den BSDS enthält, da:

- Der Eintrag für den Datensatz wurde gelöscht, wird aber erneut benötigt.
- Sie kopieren den Inhalt eines aktiven Protokolldatensatzes in einen anderen Datensatz.
- Sie führen die Wiederherstellung des BSDS von einer Sicherungskopie aus.

Erhöhen der Größe des aktiven Protokolls

Es gibt zwei Möglichkeiten, diesen Prozess zu erreichen.

1. Wenn der WS-Manager aktiv ist:

- a. Definieren Sie neue größere Protokollatengruppen mit JCL.
 - b. Fügen Sie die neuen Protokollatensätze mit dem MQSC-Befehl DEFINE LOG zum aktiven Warteschlangenmanager hinzu.
 - c. Verwenden Sie den MQSC-Befehl ARCHIVE LOG, um die aktuelle aktive Protokolldatei zu versetzen, um ein neues größeres Protokoll zu verwenden.
 - d. Warten Sie, bis das Archiv der kleineren aktiven Protokolldatei abgeschlossen ist.
 - e. Fahren Sie den WS-Manager mit dem Dienstprogramm CSQJU003 herunter, um die alten kleineren aktiven Protokolle zu entfernen.
 - f. Starten Sie den Warteschlangenmanager erneut.
2. Wenn der WS-Manager inaktiv ist:

- a. Stoppen Sie den Warteschlangenmanager. Dieser Schritt ist erforderlich, weil IBM MQ alle aktiven Protokolldateien für seine exklusive Verwendung zuordnet, wenn er aktiv ist.
- b. Verwenden Sie die Option Zugriffsmethodenservices ALTER mit der Option NEWNAME, um Ihre aktiven Protokolldateien umzubenennen.
- c. Verwenden Sie DEFINE Method Services DEFINE, um größere aktive Protokolldateien zu definieren.

Wenn Sie die alten Dateigruppe wiederverwenden, müssen Sie das Dienstprogramm zum Ändern des Protokollbestands nicht ausführen, um neue Namen in den BSDSs zu erstellen. Die alten Dateinamen und die korrekten RBA-Bereiche befinden sich bereits in den BSDSs.

- d. Verwenden Sie REPRO für Access Method Services, um die alten (umbenannten) Datensätze in ihre entsprechenden neuen Dateien zu kopieren.

Anmerkung: Dieser Schritt kann sehr lange dauern, so dass Ihr Unternehmen für diesen Zeitraum nicht in Aktion sein kann.

- e. Starten Sie den Warteschlangenmanager.

Wenn alle Protokollatensätze die gleiche Größe haben, ist Ihr System operationell konsistenter und effizienter. Wenn die Protokollatensätze nicht dieselbe Größe haben, ist es schwieriger, die Protokolle Ihres Systems zu protokollieren, sodass der Speicherplatz verschwendet werden kann.

Die Verwendung von CSQJUFMT

Führen Sie ein CSQJUFMT-Format nicht aus, wenn Sie die Größe eines aktiven Protokolls erhöhen.

Wenn Sie CSQJUFMT ausführen (um einen Leistungsvorteil zu erhalten, wenn der Warteschlangenmanager zum ersten Mal in das neue aktive Protokoll schreibt), erhalten Sie folgende Nachrichten:

```
IEC070I 203-204,XS95GLX,REPRO02,OUTPUT,B857,SPMG02, 358
IEC070I MG.W.MG4E.LOGCOPY1.DS02,MG.W.MG4E.LOGCOPY1.DS02.DATA,
IDC3302I ACTION ERROR ON MG.W.MG4E.LOGCOPY1.DS02
IDC3351I ** VSAM I/O RETURN CODE IS 28 - RPLFDBWD = X'2908001C'
IDC31467I MAXIMUM ERROR LIMIT REACHED.

IDC0005I NUMBER OF RECORDS PROCESSED WAS 0
```

Stellen Sie außerdem sicher, dass Sie ein neues leeres Protokoll definieren, wenn Sie den Access Method Services REPRO verwenden.

Wenn Sie REPRO verwenden, um die alte (umbenannte) Datei in die entsprechende neue Datei zu kopieren, wird der Standardwert NOREPLACE verwendet.

Dies bedeutet, dass REPRO keinen Datensatz ersetzt, der sich bereits auf dem angegebenen Datensatz befindet. Wenn die Formatierung für die Datei vorgenommen wird, wird der RBA-Wert zurückgesetzt. Das Netzergebnis ist eine Datei, die nach der Formatierung nicht leer ist.

Änderungen für Archivprotokolle

In diesem Thema wird erläutert, wie die Archivprotokolle geändert werden können.

Sie können das Kennwort von, Einträge im BSDS für Archivprotokolle hinzufügen, löschen und ändern. Hier werden nur Beispiele angezeigt. Ersetzen Sie die Dateinamen, die mit den Namen angezeigt werden, die Sie verwenden möchten. Weitere Informationen zum Dienstprogramm finden Sie im Abschnitt [Dienstprogramm zum Ändern des Protokollbestands](#).

- [Archivprotokoll hinzufügen](#)
- [Archivprotokoll löschen](#)
- [Kennwort eines Archivprotokolls ändern](#)

Archivprotokoll hinzufügen

Wenn die Wiederherstellung eines Objekts vom Lesen einer vorhandenen Archivprotokolldatei abhängig ist, muss die BSDS Informationen zu dieser Datei enthalten, damit IBM MQ sie finden kann. Gehen Sie wie folgt vor, um Informationen zu einer vorhandenen Archivprotokolldatei im BSDS zu registrieren:

```
NEWLOG DSNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,  
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

Archivierungsprotokoll löschen

Um eine gesamte Archivierungsprotokolldatei auf einem oder mehreren Datenträgern zu löschen, verwenden Sie:

```
DELETE DSNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

Kennwort eines Archivprotokolls ändern

Wenn Sie das Kennwort für eine vorhandene Archivprotokolldatei ändern, müssen Sie auch die Informationen im BSDS ändern.

1. Listen Sie das BSDS mit Hilfe des Dienstprogramms zum Drucken der Protokollzuordnung auf.
2. Löschen Sie den Eintrag für die Archivprotokolldatei mit dem geänderten Kennwort unter Verwendung der DELETE-Funktion des Dienstprogramms CSQJU003 (siehe Abschnitt [Das Dienstprogramm zum Ändern des Protokollbestands](#)).
3. Benennen Sie die Datei wie für eine neue Archivprotokolldatei. Verwenden Sie die Funktion NEWLOG des Dienstprogramms CSQJU003 (siehe Thema [Das Dienstprogramm zum Ändern des Protokollbestands](#)) und geben Sie das neue Kennwort, das Starten und Beenden von RBAs und die Datenträgerseriennummern (die in der Ausgabe des Dienstprogramms für Druckprotokollzuordnungen zu finden sind, siehe [Dienstprogramm für die Druckprotokollzuordnung](#)).

Gehen Sie wie folgt vor, um das Kennwort für neue Archivierungsprotokolldateien zu ändern:

```
ARCHIVE PASSWORD= password
```

Gehen Sie wie folgt vor, um das Einlegen von Kennwörtern in neue Archivprotokolldateien zu stoppen

```
ARCHIVE NOPASSWD
```

Anmerkung: Verwenden Sie die Dienstprogrammfunktion ARCHIVE nur, wenn Sie keinen externen Sicherheitsmanager haben.

High Level Qualifier (HLQ) für die Protokolle und BSDS ändern

Verwenden Sie dieses Thema, um die Prozedur zu verstehen, die erforderlich ist, um das übergeordnete Qualifikationsmerkmal (HLQ) zu ändern.

Vorbereitende Schritte

Sie müssen den Warteschlangenmanager in der Regel beenden, bevor Sie die Protokolle oder Datensätze in die neuen Dateien kopieren. Dadurch wird sichergestellt, dass die Daten konsistent sind und während des Neustarts keine Wiederherstellung erforderlich ist.

Informationen zu diesem Vorgang

Diese Übung enthält Informationen zum Ändern des HLQ für die Protokolle und BSDS. Führen Sie dazu die folgenden Schritte aus:

Vorgehensweise

1. Führen Sie das Protokolldruckdienstprogramm CSQJU004 aus, um die Protokolldatensatzinformationen aufzuzeichnen. Diese Informationen werden später benötigt.
2. Sie können entweder:
 - a) Führen Sie die DSS-Sicherung und -Zurückschreibung mit einer Umbenennung im Protokoll- und BSDS-Datei aus, die umbenannt werden sollen, oder
 - b) Mit dem Befehl AMS DEFINE und REPRO die HLQ-Dateien erstellen und die Daten aus den alten Dateien kopieren.
3. Ändern Sie die MSTR- und CHIN-Prozeduren so, dass sie auf die neuen Datensätze zeigen.
4. Löschen Sie die alten Protokollinformationen in der neuen Kopie des BSDS mit CSQJU003.
5. Definieren Sie die neuen Protokolldatensätze mit der Funktion NEWLOG von CSQJU003 in den neuen BSDS.
Behalten Sie alle Informationen zu jedem Protokoll bei, abgesehen von der HLQ.
6. Die neuen BSDS sollten die gleichen Informationen widerspiegeln, die für die alten Protokolle im alten BSDS aufgezeichnet wurden.
Das HLQ sollte das einzige sein, das sich geändert hat.

Nächste Schritte

Vergleichen Sie die CSQJU004-Ausgabe für das alte und das neue BSDS, um sicherzustellen, dass sie EXACTLY gleich (mit Ausnahme der HLQs) sehen, bevor Sie den WS-Manager starten.

Anmerkung: Bei der Ausführung dieser Operationen ist Vorsicht geboten. Falsche Aktionen können zu nicht wiederherstellbaren Situationen führen. Überprüfen Sie die Ausgabe des Befehls PRINT LOG MAP UTILITY, und stellen Sie sicher, dass alle für die Wiederherstellung oder den Neustart erforderlichen Informationen enthalten sind.

BSDS wiederherstellen

Wenn IBM MQ im BSDS-Dualmodus betrieben wird und dann eine BSDS beschädigt wird und IBM MQ in den BSDS-Einzelmodus zwingt, läuft der Betrieb von IBM MQ problemlos weiter (bis zum nächsten Neustart).

Gehen Sie wie folgt vor, um die Umgebung in den dualen BSDS-

1. Mit den Zugriffsmethodenservices können Sie die beschädigten BSDS umbenennen oder löschen und ein neues BSDS mit demselben Namen wie das beschädigte BSDS definieren. Beispielsteueranweisungen finden Sie in Job CSQ4BREC in thlqual.SCSQPROC.

2. Geben Sie den IBM MQ-Befehl RECOVER BSDS aus, um eine Kopie der gültigen BSDS in der neu zugeordneten Datei zu erstellen und den BSDS-Dualmodus wiederherzustellen.

Wenn IBM MQ im BSDS-Einzelmodus betrieben wird und die BSDS beschädigt wird oder wenn IBM MQ im BSDS-Dualmodus betrieben wird und beide BSDSs beschädigt werden, wird der Warteschlangenmanager gestoppt und erst dann erneut gestartet, wenn die BSDS-Dateien repariert sind. In diesem Fall geschieht Folgendes:

1. Suchen Sie die BSDS, die der zuletzt archiviertesten Archivprotokolldatei zugeordnet sind. Der Datensatzname des letzten Archivprotokolls wird im Jobprotokoll im letzten Vorkommen der Nachricht CSQJ003I angezeigt, die angibt, dass die Auslastungsverarbeitung erfolgreich abgeschlossen wurde. In Vorbereitung auf den Rest dieses Verfahrens, ist es eine gute Praxis, ein Protokoll aller erfolgreichen Archive, die von dieser Nachricht notiert werden:

- Wenn sich Archivprotokolle auf DASD befinden, wird das BSDS auf einer beliebigen verfügbaren DASD-Einheit zugeordnet. Der BSDS-Name ist wie der Name des entsprechenden Archivprotokolldatensatzes. Ändern Sie nur den ersten Buchstaben des letzten Qualifikationsmerkmals von A nach B, wie in diesem Beispiel:

Archivprotokollname

CSQ.ARCHLOG1. **A** 0000001

BSDS-Kopienname

CSQ.ARCHLOG1. **B** 0000001

- Wenn sich Archivprotokolle auf Band befinden, ist der BSDS der erste Datensatz des ersten Datenträgers des Archivprotokolls. Die BSDS wird auf späteren Datenträgern nicht wiederholt.
2. Wenn die aktuellste Archivprotokolldatei keine Kopie des BSDS enthält (z. B. aufgrund eines Fehlers beim Ausladen), suchen Sie eine frühere Kopie des BSDS aus der früheren Ausladeverarbeitung.
 3. Benennen Sie *beschädigte* BSDSs mit dem Befehl ALTER für den Zugriff auf Methodenservices mit der Option NEWNAME um. Wenn Sie beschädigte BSDS löschen möchten, verwenden Sie den Befehl DELETE für Zugriffsmethodenservices. Verwenden Sie für jede beschädigte BSDS die Zugriffsmethodenservices, um eine neue BSDS als Ersatzdatei zu definieren. Der Job CSQ4BREC in thlqual.SCSQPROC enthält Zugriffssteuerungsanweisungen für den Zugriff auf Methodenservices, um ein neues BSDS zu definieren.
 4. Verwenden Sie den Befehl REPRO von Access Method Services, um die BSDS aus dem Archivprotokoll in eine der Ersatz-BSDS zu kopieren, die Sie in Schritt „3“ auf Seite 400 definiert haben. Kopieren Sie keine Daten in die zweite Ersatz-BSDS, denn dies erfolgt in Schritt „5“ auf Seite 402 vor.

- a. Drucken Sie den Inhalt der Ersatz-BSDS.

Verwenden Sie das Dienstprogramm für die Druckprotokollzuordnung (CSQJU004), um den Inhalt der Ersatz-BSDS zu drucken. Auf diese Weise können Sie den Inhalt der Ersatz-BSDS überprüfen, bevor Sie Ihre Wiederherstellungsarbeiten fortsetzen.

- b. Aktualisieren Sie den Bestand des Archivprotokolldatensatzes in der Ersatz-BSDS.

Überprüfen Sie die Ausgabe des Dienstprogramms für die Druckprotokollzuordnung, und prüfen Sie, ob die Ersatz-BSDS keinen Datensatz des Archivprotokolls enthält, aus dem das BSDS kopiert wurde. Wenn es sich bei dem Ersatz-BSDS um eine alte Kopie handelt, enthält der Bestand möglicherweise nicht alle Archivierungsprotokolldatensätze, die in letzter Zeit erstellt wurden. Der BSDS-Bestand der Archivierungsprotokolldatensätze muss aktualisiert werden, um den aktuellen Subsystembestand widerzuspiegeln.

Verwenden Sie die Anweisung NEWLOG des Änderungsprotokollbestands (CSQJU003) NEWLOG, um die Ersatz-BSDS zu aktualisieren, indem Sie einen Datensatz des Archivprotokolls hinzufügen, aus dem das BSDS kopiert wurde. Wenn die Archivprotokolldatei kennwortgeschützt ist, verwenden Sie die Option PASSWORD der Funktion NEWLOG. Wenn die Archivprotokolldatei katalogisiert ist, müssen Sie außerdem sicherstellen, dass die Option CATALOG der Funktion NEWLOG ordnungsgemäß auf CATALOG=YES gesetzt ist. Verwenden Sie die Anweisung NEWLOG, um alle zusätzlichen Archivprotokolldateien hinzuzufügen, die später erstellt wurden als die BSDS-Kopie.

- c. Aktualisieren Sie die Kennwörter in der Ersatz-BSDS.

Das BSDS enthält Kennwörter für die Archivierungsprotokolldatensätze und für die aktiven Protokolldateien. Um sicherzustellen, dass die Kennwörter in den Ersatz-BSDS die aktuellen Kennwörter widerspiegeln, die von Ihrer Installation verwendet werden, verwenden Sie die Dienstprogramm-funktion ARCHIVE des Änderungsprotokolls mit der Option PASSWORD.

d. Aktualisieren Sie das Inventar der aktiven Protokolldatei in der Ersatz-BSDS.

Unter ungewöhnlichen Umständen kann Ihre Installation seit dem Kopieren des BSDS möglicherweise aktive Protokolldatengruppen hinzugefügt, gelöscht oder umbenannt haben. In diesem Fall gibt der Ersatz-BSDS nicht die tatsächliche Anzahl oder Namen der aktiven Protokolldateien an, die Ihre Installation derzeit verwendet.

Wenn Sie eine aktive Protokolldatei aus dem Ersatz-BSDS-Protokollbestand löschen müssen, verwenden Sie die Funktion DELETE für das Dienstprogramm zum Ändern des Protokollbestands.

Wenn Sie eine aktive Protokolldatei zum Ersatz-BSDS-Protokollbestand hinzufügen müssen, verwenden Sie die Funktion NEWLOG des Änderungsprotokollbestands. Stellen Sie sicher, dass der RBA-Bereich in der NEWLOG-Funktion korrekt angegeben ist. Wenn die aktive Protokolldatei kennwortgeschützt ist, verwenden Sie die Option PASSWORD.

Wenn Sie eine aktive Protokolldatei umbenennen müssen, die im Ersatz-BSDS-Protokollbestand festgelegt ist, verwenden Sie die Funktion DELETE für das Änderungsprotokoll, gefolgt von der Funktion NEWLOG. Stellen Sie sicher, dass der RBA-Bereich in der NEWLOG-Funktion korrekt angegeben ist. Wenn die aktive Protokolldatei kennwortgeschützt ist, verwenden Sie die Option PASSWORD.

e. Aktualisieren Sie die RBA-Bereiche des aktiven Protokolls in der Ersatz-BSDS.

Später, wenn der Warteschlangenmanager erneut gestartet wird, vergleicht er die RBAs der aktiven Protokolldatensätze, die im BSDS aufgelistet sind, mit den RBAs, die in den tatsächlichen aktiven Protokolldateien gefunden wurden. Wenn die RBAs nicht übereinstimmen, wird der WS-Manager nicht erneut gestartet. Das Problem wird vergrößert, wenn eine alte Kopie des BSDS verwendet wird. Beheben Sie diesen Fehler, indem Sie das Dienstprogramm zum Ändern des Protokollbestands (CSQJU003) verwenden, um die im BSDS gefundenen RBAs mithilfe der RBAs in den tatsächlichen aktiven Protokolldateien anzupassen. Gehen Sie dazu wie folgt vor:

- Verwenden Sie das Dienstprogramm zum Drucken von Protokollsätzen (CSQ1LOGP), um einen Zusammenfassungsbericht der aktiven Protokolldatei zu drucken. Hier sehen Sie die Start- und die End-RBAs.
- Vergleichen Sie die tatsächlichen RBA-Bereiche mit den RBA-Bereichen, die Sie gerade gedruckt haben, wenn die RBAs aller aktiven Protokolldatensätze bekannt sind.

Wenn die RBA-Bereiche für alle aktiven Protokolldatensätze gleich sind, können Sie mit dem nächsten Wiederherstellungsschritt fortfahren, ohne dass weitere Arbeiten ausgeführt werden müssen.

Wenn die RBA-Bereiche nicht identisch sind, passen Sie die Werte im BSDS so an, dass sie die tatsächlichen Werte widerspiegeln. Verwenden Sie für jede aktive Protokolldatei, die den RBA-Bereich angepasst haben muss, die Funktion DELETE function des Änderungsprotokolls DELETE, um die aktive Protokolldatei aus dem Bestand in der Ersatz-BSDS zu löschen. Anschließend die Funktion NEWLOG verwenden, um die aktive Protokolldatei erneut zu definieren, die auf BSDS gesetzt ist. Wenn die aktiven Protokolldatensätze kennwortgeschützt sind, verwenden Sie die Option PASSWORD der Funktion NEWLOG.

f. Wenn für jede Kopie des aktiven Protokolls nur zwei aktive Protokolldateien angegeben werden, kann IBM MQ beim Neustart des Warteschlangenmanagers Schwierigkeiten haben. Das Problem kann auftreten, wenn eine der aktiven Protokolldateien voll ist und nicht ausgelagert wurde, während die zweite aktive Protokolldatei kurz vor dem Füllen steht. Fügen Sie in diesem Fall für jede Kopie des aktiven Protokolls eine neue aktive Protokolldatei hinzu, und definieren Sie jede neue aktive Protokolldatei im Ersatzprotokoll des BSDS-Protokolls.

Mit dem Befehl DEFINE für Zugriffsmethodenservices (Access Method Services) eine neue aktive Protokolldatei für jede Kopie des aktiven Protokolls definieren und die Funktion NEWLOG des Änderungsprotokolls NEWLOG verwenden, um die neuen aktiven Protokolldateien in der Ersatz-BSDS

zu definieren. Sie müssen die RBA-Bereiche in der NEWLOG-Anweisung nicht angeben. Wenn die aktiven Protokolldateien jedoch kennwortgeschützt sind, verwenden Sie die Option PASSWORD der Funktion NEWLOG. Beispielsteueranweisungen zum Ausführen dieser Task finden Sie in Job CSQ4LREC in thlqual.SCSQPROC.

5. Kopieren Sie das aktualisierte BSDS in die zweite neue BSDS-Datei. Die BSDSs sind jetzt identisch. Verwenden Sie das Dienstprogramm für die Druckprotokollzuordnung (CSQJU004), um den Inhalt der zweiten Ersatz-BSDS an diesem Punkt zu drucken.
6. Im Abschnitt Probleme mit dem aktiven Protokoll finden Sie Informationen darüber, was zu tun ist, wenn Sie die aktuelle aktive Protokolldatei verloren haben.
7. Starten Sie den Warteschlangenmanager unter Verwendung der neu erstellten BSDS erneut. IBM MQ legt die aktuelle RBA fest und welche aktiven Protokolle archiviert werden müssen.

Seitengruppen verwalten

In diesem Thema wird erläutert, wie die Seitengruppen verwaltet werden, die einem Warteschlangenmanager zugeordnet sind.

In diesem Abschnitt wird beschrieben, wie die Seitengruppen, die einem Warteschlangenmanager zugeordnet sind, hinzugefügt, kopiert und in der Regel verwaltet werden. Er enthält folgende Abschnitte:

- „Vorgehensweise beim Ändern des übergeordneten Qualifikationsmerkmals (HLQ) für die Seitengruppen“ auf Seite 402
- „Vorgehensweise zum Hinzufügen einer Seitengruppe zu einem Warteschlangenmanager“ auf Seite 403
- „Was zu tun ist, wenn eine Ihrer Seitengruppen voll wird“ auf Seite 403
- „Wie Lasten auf Seitengruppen ausgeglichen werden“ auf Seite 404
- Wie Sie die Größe einer Seitengruppe erhöhen
- „Vorgehensweise zum Reduzieren einer Seitengruppe“ auf Seite 408
- „So führen Sie eine Seitengruppe wieder ein“ auf Seite 408
- „Seitengruppen sichern und wiederherstellen“ auf Seite 409
- „So löschen Sie Seitengruppen“ auf Seite 413
- „Vorgehensweise zum Sichern und Zurückschreiben von Warteschlangen mit CSQUTIL“ auf Seite 413

Eine Beschreibung der Seitengruppen, Speicherklassen, Puffer und Pufferpools sowie einige der anwendungstechnischen Überlegungen finden Sie unter Seitengruppen .

Vorgehensweise beim Ändern des übergeordneten Qualifikationsmerkmals (HLQ) für die Seitengruppen

In dieser Übung wird beschrieben, wie die HLQ für die Seitengruppen geändert wird. Führen Sie die folgenden Schritte aus, um diese Task auszuführen:

1. Definieren Sie die neuen HLQ-Seitengruppen.
2. Wenn die Größenordnung mit den alten Seitengruppen identisch ist, kopieren Sie die vorhandene Seitengruppe mit REPRO auf die leeren neuen HLQ-Seitengruppen. Wenn Sie die Größe der Seitengruppen erhöhen, verwenden Sie die Funktion FORMAT von CSQUTIL, um die Zielseitengruppe zu formatieren. Weitere Informationen finden Sie im Abschnitt Seitengruppen formatieren (FORMAT) .
3. Verwenden Sie die Funktion COPYPAGE von CSQUTIL, um alle Nachrichten von der Quellenseitengruppe auf die Zielseitengruppe zu kopieren. Weitere Informationen hierzu finden Sie im Abschnitt Seitengruppe erweitern (COPYPAGE) .
4. Ändern Sie die DD-Anweisung CSQP00xx in der WS-Managerprozedur so, dass sie auf die neuen HLQ-Seitengruppen verweist.

Starten Sie den Warteschlangenmanager erneut, und überprüfen Sie die Änderungen an den Seitengruppen.

Vorgehensweise zum Hinzufügen einer Seitengruppe zu einem Warteschlangenmanager

In dieser Beschreibung wird vorausgesetzt, dass bereits ein Warteschlangenmanager vorhanden ist. Möglicherweise müssen Sie eine Seitengruppe hinzufügen, wenn der WS-Manager beispielsweise neue Anwendungen mit neuen Warteschlangen bewältigen muss.

Gehen Sie wie folgt vor, um eine neue Seitengruppe hinzuzufügen:

1. Definieren und formatieren Sie die neue Seitengruppe. Sie können die Beispiel-JCL in `thlqual.SCSQPROC (CSQ4PAGE)` als Basis verwenden. Weitere Informationen finden Sie im Abschnitt [Seitengruppen formatieren \(FORMAT\)](#).

Es ist wichtig, dass keine Seitengruppen formatiert werden, die im Gebrauch sind, es sei denn, dies ist die Absicht, die Sie beabsichtigen. Ist dies der Fall, verwenden Sie die Option `FORCE` der Dienstprogrammfunktion `FORMAT`.

2. Verwenden Sie den Befehl `DEFINE PSID` mit der Option `DSN`, um die Seitengruppe einem Pufferpool zuzuordnen.
3. Fügen Sie die entsprechenden Speicherklassendefinitionen für Ihre Seitengruppe hinzu, indem Sie `DEFINE STGCLASS`-Befehle absetzen.
4. Optional können Sie wie folgt dokumentieren, wie der Warteschlangenmanager konfiguriert ist:
 - a. Fügen Sie die neue Seitengruppe der gestarteten Taskprozedur für Ihren Warteschlangenmanager hinzu.
 - b. Fügen Sie eine Definition für die neue Seite hinzu, die auf Ihrem `CSQINP1`-Initialisierungsdatensatz festgelegt ist.
 - c. Fügen Sie Ihrem `CSQ4INYP`-Initialisierungsdatei-Member eine Definition für die neue Speicherklasse hinzu.

Weitere Informationen zu den Befehlen `DEFINE PSID` und `DEFINE STGCLASS` finden Sie in [DEFINE PSID](#) und [DEFINE STGCLASS](#).

Was zu tun ist, wenn eine Ihrer Seitengruppen voll wird

Informationen zur Verwendung von Seitengruppen finden Sie mithilfe des IBM MQ-Befehls `DISPLAY USAGE`. Beispiel:

```
DISPLAY USAGE PSID(03)
```

zeigt den aktuellen Status der Seitengruppe 03 an. Hier erfahren Sie, wie viele freie Seiten diese Seitengruppe hat.

Wenn Sie sekundäre Speicherbereiche für Ihre Seitengruppen definiert haben, werden sie bei jedem Auffüllen dynamisch erweitert. Schließlich werden alle sekundären Speicherbereiche verwendet, oder es ist kein weiterer Plattenspeicherplatz verfügbar. Wenn dies geschieht, empfängt eine Anwendung den Rückkehrcode `MQRC_STORAGE_MEDIUM_FULL`.

Wenn eine Anwendung einen Rückkehrcode von `MQRC_STORAGE_MEDIUM_FULL` von einem MQI-Aufruf empfängt, ist dies ein eindeutiger Hinweis darauf, dass auf der Seitengruppe nicht genügend Speicherplatz vorhanden ist. Bleibt das Problem bestehen oder ist es wahrscheinlich, dass es wieder auftritt, müssen Sie etwas tun, um es zu lösen.

Sie können dieses Problem auf mehrere Arten angehen:

- Sie können die Last zwischen Seitengruppen ausgleichen, indem Sie Warteschlangen von einer Seitengruppe in eine andere verschieben.
- Erweitern Sie die Seitengruppe. Entsprechende Anweisungen finden Sie im Abschnitt [„Wie Sie die Größe einer Seitengruppe erhöhen“](#) auf Seite 406.
- Definieren Sie die Seitengruppe neu, so dass sie über 4 GB bis zu einer maximalen Größe von 64 GB erweitert werden kann. Anweisungen hierzu finden Sie im Abschnitt [Eine Seitengruppe mit einer Größe von mehr als 4 GB definieren](#).

Wie Lasten auf Seitengruppen ausgeglichen werden

Der Lastausgleich auf Seitengruppen bedeutet, dass die Nachrichten, die einer oder mehreren Warteschlangen zugeordnet sind, von einer Seitengruppe in eine andere, weniger verwendete, Seitengruppe verschoben werden. Verwenden Sie diese Technik, wenn es nicht sinnvoll ist, die Seitengruppe zu erweitern.

Mithilfe der entsprechenden IBM MQ-Befehle können Sie ermitteln, welche Warteschlangen eine Seitengruppe verwenden. Wenn Sie beispielsweise herausfinden möchten, welche Warteschlangen der Seitengruppe 02 zugeordnet sind, ermitteln Sie zuerst, welche Speicherklassen der Seitengruppe 02 zugeordnet sind, indem Sie den folgenden Befehl verwenden:

```
DISPLAY STGCLASS(*) PSID(02)
```

Verwenden Sie anschließend den folgenden Befehl, um zu ermitteln, welche Warteschlangen welche Speicherklasse verwenden:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

Nicht gemeinsam genutzte Warteschlange verschieben

Mit dem MQSC-Befehl MOVE QLOCAL (eine Beschreibung finden Sie unter [MOVE QLOCAL](#)) können Sie Warteschlangen zusammen mit den in ihnen enthaltenen Nachrichten in eine andere Seitengruppe verschieben. Wenn Sie die Warteschlange oder die Warteschlangen angegeben haben, die Sie in eine neue Seitengruppe verschieben möchten, führen Sie diese Prozedur für jede dieser Warteschlangen aus:

1. Stellen Sie sicher, dass die Warteschlange, die Sie verschieben möchten, nicht von Anwendungen verwendet wird (d. a. IPPROCS- und OPPROCS-Werte aus dem Befehl DISPLAY QSTATUS sind null) und dass sie keine nicht festgeschriebenen Nachrichten enthält (der UNCOM-Wert aus dem Befehl DISPLAY QSTATUS ist NO).

Anmerkung: Der einzige Weg, um sicherzustellen, dass dieser Status weiterhin besteht, ist es, die Sicherheitsberechtigung der Warteschlange vorübergehend zu ändern. Weitere Informationen finden Sie im Abschnitt [Profile für die Warteschlangensicherheit](#).

Wenn dies nicht möglich ist, können die späteren Schritte in dieser Prozedur fehlschlagen, wenn Anwendungen die Warteschlange trotz vorsorgliche Schritte wie z. B. PUT (DISABLED) verwenden. Nachrichten können jedoch bei dieser Prozedur nie verloren gehen.

2. Verhindern Sie, dass Anwendungen Nachrichten in die Warteschlange einreihen, indem Sie die Warteschlangendefinition ändern, um MQPUT zu inaktivieren. Ändern Sie die Warteschlangendefinition in PUT (DISABLED).
3. Definieren Sie eine temporäre Warteschlange mit denselben Attributen wie die Warteschlange, die mit dem Befehl verschoben wird:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

Anmerkung: Wenn diese temporäre Warteschlange bereits von einer früheren Ausführung vorhanden ist, löschen Sie sie, bevor Sie die Definition ausführen.

4. Verschieben Sie die Nachrichten mit dem folgenden Befehl in die temporäre Warteschlange:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_Queue)
```

5. Löschen Sie die Warteschlange, die Sie verschieben möchten, mit dem folgenden Befehl:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Definieren Sie eine neue Speicherklasse, die der erforderlichen Seitengruppe zugeordnet ist, z. B.:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Fügen Sie die neue Speicherklassendefinition zu den CSQINP2-Dateien hinzu, die für den nächsten Neustart des Warteschlangenmanagers bereit sind.

7. Definieren Sie die Warteschlange, die Sie verschieben, erneut, indem Sie das Attribut für die Speicherklasse ändern:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_Queue) STGCLASS(NEW)
```

Wenn die Warteschlange neu definiert wird, basiert sie auf der temporären Warteschlange, die in Schritt „3“ auf Seite 404 erstellt wurde.

8. Verschieben Sie die Nachrichten mit dem folgenden Befehl zurück in die neue Warteschlange:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. Die in Schritt „3“ auf Seite 404 erstellte Warteschlange wird nicht mehr benötigt. Verwenden Sie den folgenden Befehl, um ihn zu löschen:

```
DELETE QL(TEMP_Queue)
```

10. Wenn die Warteschlange, die versetzt wird, in den CSQINP2-Dateien definiert wurde, ändern Sie das Attribut STGCLASS des entsprechenden Befehls DEFINE QLOCAL in den CSQINP2-Dateien. Fügen Sie das Schlüsselwort REPLACE hinzu, so dass die vorhandene Warteschlangendefinition ersetzt wird.

[Abbildung 46 auf Seite 406](#) zeigt einen Auszug aus einem Lastausgleichsjob.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE STGCLASS(NEW) PSID(2)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*

```

Abbildung 46. Auszug aus einem Lastausgleichsjob für eine Seitengruppe

Wie Sie die Größe einer Seitengruppe erhöhen

Sie können anfänglich eine Seitengruppe anlegen, die größer als 4 GB ist. Weitere Informationen hierzu finden Sie unter [Seitengruppe so definieren, dass sie größer als 4 GB ist](#).

Eine Seitengruppe kann so definiert werden, dass sie automatisch eingeblendet wird, wenn sie durch Angabe von EXPAND (SYSTEM) oder EXPAND (USER) voll wird. Wenn Ihre Seitengruppe mit EXPAND (NONE) definiert wurde, können Sie sie auf eine der beiden folgenden Arten erweitern:

- Ändern Sie seine Definition, um eine automatische Erweiterung zu ermöglichen. Weitere Informationen hierzu finden Sie unter [Seitengruppe so ändern, dass eine automatische Erweiterung zugelassen wird](#).
- Erstellen Sie eine neue, größere Seitengruppe, und kopieren Sie die Nachrichten aus der alten Seitengruppe auf die neue. Weitere Informationen hierzu finden Sie unter [Nachrichten in eine neue, größere Seitengruppe verschieben](#).

Definieren einer Seitengruppe, die größer als 4 GB ist

IBM MQ kann eine Seitengruppe mit einer Größe von bis zu 64 GB verwenden, sofern die Datei mit "erweiterter Adressierbarkeit" für VSAM definiert ist. Die erweiterte Adressierbarkeit ist ein Attribut, das durch eine SMS-Datenklasse übertragen wird. In dem Beispiel, das in der folgenden Beispiel-JCL dargestellt ist, wird die Verwaltungsklasse 'EXTENDED' für SMS mit 'Erweiterter Adressierbarkeit' definiert. Wenn Ihre vorhandene Seitengruppe derzeit nicht als erweiterte Adressierbarkeit definiert ist, verwenden Sie die folgende Methode, um eine Datei auf eine erweiterte Adressierbarkeitsdatei zu migrieren.

1. Stoppen Sie den Warteschlangenmanager.
2. Verwenden Sie Access Method Services, um die vorhandene Seitengruppe umzubenennen.
3. Definieren Sie eine Zielseitengruppe, die die gleiche Größe wie die vorhandene Seitengruppe hat, jedoch mit DATACLAS (EXTENDED).

Anmerkung: Dateien mit erweiterter Formatierung müssen SMS-verwaltet sein. Dies sind die Mechanismen für das Anfordern erweiterter Formate für VSAM-Dateigruppen:

- Verwenden einer Datenklasse mit dem DSNTYPE-Wert EXT und dem Unterparameter R oder P, um eine Angabe erforderlich oder bevorzugt zu machen.
- Coding DSNTYPE=EXTREQ (erweitertes Format ist erforderlich) oder DSNTYPE=EXTPREF (erweitertes Format wird bevorzugt) in der Datendefinitionsanweisung.
- Codieren Sie den Parameter LIKE= in der Datendefinitionsanweisung, um auf eine vorhandene Datei mit erweiterter Formatierung zu verweisen.

Weitere Informationen finden Sie unter [Einschränkungen beim Definieren von Dateien für das erweiterte Format](#).

4. Verwenden Sie die Funktion COPYPAGE von CSQUTIL, um alle Nachrichten von der Quellenseitengruppe auf die Zielseitengruppe zu kopieren. Weitere Informationen hierzu finden Sie im Abschnitt [Seitengruppe erweitern \(COPYPAGE\)](#).
5. Starten Sie den Warteschlangenmanager erneut.
6. Ändern Sie die Seitengruppe so, dass die Systemerweiterung verwendet wird, damit sie über ihre aktuelle Zuordnung hinaus wächst.

Die folgende JCL zeigt Beispielbefehle für den Zugriff auf Methodenservices:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER 'VICY.CSQ1.PAGE01' -
NEWNAME('VICY.CSQ1.PAGE01.OLD')
ALTER 'VICY.CSQ1.PAGE01.DATA' -
NEWNAME('VICY.CSQ1.PAGE01.DATA.OLD')
DEFINE CLUSTER (NAME('VICY.CSQ1.PAGE01') -
MODEL('VICY.CSQ1.PAGE01.OLD') -
DATACLAS(EXTENDED))
/*
```

Ändern einer Seitengruppe, um eine automatische Erweiterung zu ermöglichen

Verwenden Sie den Befehl ALTER PSID mit den Optionen EXPAND (USER) oder EXPAND (SYSTEM). Allgemeine Informationen zum Erweitern von Seitengruppen finden Sie unter [ALTER PSID](#) und [Expanding a page set \(COPYPAGE\)](#).

Nachrichten in eine neue, größere Seitengruppe verschieben

Dieses Verfahren umfasst das Stoppen und erneute Starten des Warteschlangenmanagers. Dadurch werden alle nicht persistenten Nachrichten, die sich nicht in gemeinsam genutzten Warteschlangen befinden, bei der Neustartzeit gelöscht. Wenn nicht persistente Nachrichten vorhanden sind, die nicht gelöscht werden sollen, verwenden Sie stattdessen den Lastausgleich. Weitere Informationen finden Sie im Abschnitt „[Wie Lasten auf Seitengruppen ausgeglichen werden](#)“ auf Seite 404. In dieser Beschreibung wird die Seitengruppe, die Sie erweitern möchten, als *Quelle* -Seitengruppe bezeichnet; die neue, größere Seitengruppe wird als *Zieladresse* -Seitengruppe bezeichnet.

Führen Sie folgende Schritte aus:

1. Stoppen Sie den Warteschlangenmanager.
2. Definieren Sie die Zielseitengruppe, und stellen Sie sicher, dass sie größer als die Quellenseitenmenge ist, mit einem größeren Wert für den sekundären Speicherbereich.
3. Verwenden Sie die FORMAT-Funktion von CSQUTIL, um die Zielseitengruppe zu formatieren. Weitere Informationen finden Sie im Abschnitt [Seitengruppen formatieren \(FORMAT\)](#).
4. Verwenden Sie die Funktion COPYPAGE von CSQUTIL, um alle Nachrichten von der Quellenseitengruppe auf die Zielseitengruppe zu kopieren. Weitere Informationen hierzu finden Sie im Abschnitt [Seitengruppe erweitern \(COPYPAGE\)](#).
5. Starten Sie den Warteschlangenmanager unter Verwendung der Zielseitengruppe erneut, indem Sie einen der folgenden Schritte ausführen:
 - Ändern Sie die gestartete Taskprozedur des WS-Managers so, dass sie auf die Zielseitengruppe verweist.
 - Mit den Zugriffsmethodenservices können Sie die Quellenseitengruppe löschen und anschließend die Zielseitengruppe umbenennen, so dass sie denselben Namen wie die Quellenseitengruppe erhält.

Achtung:

Stellen Sie sicher, dass Sie die erforderlichen Sicherungskopien erstellt haben, bevor Sie eine IBM MQ-Seitengruppe löschen.

Vorgehensweise zum Reduzieren einer Seitengruppe

Verhindern Sie, dass alle Benutzer außer dem IBM MQ-Administrator den Warteschlangenmanager verwenden. Zum Beispiel durch Ändern der Zugriffssicherheitseinstellungen.

Wenn Sie eine große Seitengruppe haben, die in der Regel leer ist (wie im Befehl DISPLAY USAGE angezeigt), können Sie die Größe des Seitensatzes verkleinern. Die Vorgehensweise hierzu umfasst die Verwendung der Funktionen COPY, FORMAT und LOAD von CSQUTIL (siehe [IBM MQ-Dienstprogramm](#)). Diese Prozedur funktioniert nicht für die Seitengruppe Null (0), da es nicht praktikabel ist, die Größe dieser Seitengruppe zu verkleinern. Dies ist nur durch eine Neuinitialisierung des Warteschlangenmanagers möglich (siehe [„Erneutes Initialisieren eines Warteschlangenmanagers“](#) auf Seite 435). Die Voraussetzung für diese Prozedur ist es, alle Benutzer aus dem System zu entfernen, so dass alle UOWs abgeschlossen sind und die Seitengruppen konsistent sind.

1. Verwenden Sie den Befehl STOP QMGR mit dem Attribut QUIESCE oder FORCE , um den WS-Manager zu stoppen.
2. Führen Sie die SCOPY-Funktion von CSQUTIL mit der Option PSID aus, um alle Nachrichtendaten aus der großen Seitengruppe zu kopieren und in einem sequenziellen Datensatz zu speichern.
3. Definieren Sie eine neue kleinere Seitengruppe, die die große Seitengruppe ersetzen soll.
4. Führen Sie die Funktion FORMAT TYPE(NEW) von CSQUTIL für die Seitengruppe aus, die Sie in Schritt „3“ auf Seite 408 erstellt haben.
5. Führen Sie einen Neustart des Warteschlangenmanagers durch, der die in Schritt „3“ auf Seite 408 erstellte Seitengruppe verwendet.
6. Führen Sie die Funktion LOAD von CSQUTIL aus, um alle Nachrichten, die in Schritt „2“ auf Seite 408 gespeichert wurden, wieder zu laden.
7. Allen Benutzern den Zugriff auf den WS-Manager erlauben.
8. Löschen Sie die alte große Seitengruppe.

So führen Sie eine Seitengruppe wieder ein

In bestimmten Szenarios ist es sinnvoll, eine alte Seitengruppe wieder in den WS-Manager zu setzen. Wenn keine bestimmte Aktion ausgeführt wird, wenn die alte Seitengruppe online gesetzt wird, erkennt der Warteschlangenmanager, dass die in der Seitengruppe selbst und in den Prüfpunktensätzen gespeicherte RBA für die Seitengruppe alt ist, und startet daher automatisch die Datenträgerwiederherstellung der Seitengruppe, um sie auf dem neuesten Stand zu bringen.

Eine solche Datenträgerwiederherstellung kann nur beim Neustart des Warteschlangenmanagers ausgeführt werden, und es wird wahrscheinlich eine beträchtliche Zeit in Anspruch nehmen, insbesondere dann, wenn die auf Band gehaltenen Archivprotokolle gelesen werden müssen. In diesem Fall ist die Seitengruppe jedoch in der Zwischenzeit offline gesetzt worden, sodass das Protokoll keine Informationen enthält, die für die Wiederherstellung der Seitengruppe relevant sind.

Es stehen die folgenden drei Auswahlmöglichkeiten zur Verfügung:

Zulassen, dass die vollständige Datenträgerwiederherstellung ausgeführt wird.

1. Stoppen Sie den Warteschlangenmanager.
2. Stellen Sie sicher, dass für die Seite, die in der Prozedur für gestartete Tasks für den Warteschlangenmanager und im Initialisierungsdatensatz CSQINP1 festgelegt ist, Definitionen verfügbar sind.
3. Starten Sie den Warteschlangenmanager erneut.

Alle Nachrichten auf der Seitengruppe zulassen, die gelöscht werden sollen.

Diese Auswahl ist nützlich, wenn eine Seitengruppe seit langer Zeit offline ist (z. B. einige Monate), und es wurde jetzt beschlossen, sie für einen anderen Zweck wiederzuverwenden.

1. Formatieren Sie die Seitengruppe mit Hilfe der Funktion FORMAT von CSQUTIL mit der Option TYPE (NEW).
2. Fügen Sie Definitionen für die Seitengruppe hinzu, die sowohl für den Warteschlangenmanager als auch für den Initialisierungsdatensatz CSQINP1 festgelegt wurde.
3. Starten Sie den Warteschlangenmanager erneut.

Bei Verwendung der Option TYPE (NEW) für die Formatierung wird der aktuelle Inhalt der Seitengruppe gelöscht und der WS-Manager angewiesen, alle historischen Informationen im Prüfpunkt der Seitengruppe zu ignorieren.

Bringen Sie die Seitengruppe online, um den Datenträgerwiederherstellungsprozess zu vermeiden.

Verwenden Sie diese Technik nur, wenn Sie sicher sind, dass die Seitengruppe seit einem sauberen Systemabschluss des WS-Managers offline war. Diese Auswahl ist am besten geeignet, wenn die Seitengruppe für einen kurzen Zeitraum offline war, in der Regel aufgrund von betriebsfähigen Problemen, wie z. B. einer Sicherung, die ausgeführt wird, während der WS-Manager gestartet wird.

1. Formatieren Sie die Seitengruppe mit Hilfe der Funktion FORMAT von CSQUTIL mit der Option TYPE (REPLACE).
2. Fügen Sie die Seitengruppe mit dem Befehl DEFINE PSID mit der Option DSN dynamisch wieder in den Warteschlangenmanager ein, oder lassen Sie sie bei einem Neustart des Warteschlangenmanagers hinzugefügt werden.

Bei Verwendung der Option TYPE (REPLACE) für die Formatierung wird geprüft, ob die Seitengruppe vom WS-Manager sauber geschlossen wurde, und markiert sie so, dass die Datenträgerwiederherstellung nicht ausgeführt wird. Es werden keine anderen Änderungen am Inhalt der Seitengruppe vorgenommen.

Seitengruppen sichern und wiederherstellen

Es stehen verschiedene Mechanismen für die Sicherung und Wiederherstellung zur Verfügung. Verwenden Sie dieses Thema, um diese Mechanismen zu verstehen.

In diesem Abschnitt werden die folgenden Themen beschrieben:

- [„Wiederherstellungspunkt für nicht gemeinsam genutzte Ressourcen erstellen“ auf Seite 409](#)
- [„Seitengruppen sichern“ auf Seite 411](#)
- [„Seitengruppen wiederherstellen“ auf Seite 411](#)
- [So löschen Sie Seitengruppen](#)

Informationen zum Erstellen eines Wiederherstellungspunkts für gemeinsam genutzte Ressourcen finden Sie unter [„Gemeinsam genutzte Warteschlangen wiederherstellen“ auf Seite 419](#).

Wiederherstellungspunkt für nicht gemeinsam genutzte Ressourcen erstellen

IBM MQ kann Objekte und nicht gemeinsam genutzte, persistente Nachrichten in ihrem aktuellen Status wiederherstellen, wenn beide:

1. Kopien von Seitengruppen von einem früheren Punkt vorhanden sind.
2. Alle IBM MQ-Protokolle sind verfügbar, um eine Wiederherstellung von diesem Punkt aus durchzuführen.

Diese stellen einen Zeitpunkt der Wiederherstellung für nicht gemeinsam genutzte Ressourcen dar.

Sowohl Objekte als auch Nachrichten werden auf Seitengruppen gehalten. Es können mehrere Objekte und Nachrichten aus verschiedenen Warteschlangen auf derselben Seitengruppe vorhanden sein. Für Wiederherstellungszwecke können Objekte und Nachrichten nicht isoliert gesichert werden, so dass eine Seitengruppe als Ganzes gesichert werden muss, um die korrekte Wiederherstellung der Daten sicherzustellen.

Das Wiederherstellungsprotokoll von IBM MQ enthält einen Datensatz mit allen persistenten Nachrichten und Änderungen an Objekten. Wenn IBM MQ fehlschlägt (z. B. aufgrund eines E/A-Fehlers in einer Seitengruppe), können Sie die Seitengruppe wiederherstellen, indem Sie die Sicherungskopie zurückschreiben und den Warteschlangenmanager erneut starten. IBM MQ wendet die Protokolländerungen auf die Seitengruppe ab dem Zeitpunkt der Sicherungskopie an.

Es gibt zwei Möglichkeiten, einen Wiederherstellungspunkt zu erstellen:

Vollständige Sicherung

Stoppen Sie den WS-Manager, der alle Aktualisierungen auf die Seitengruppen erzwingt.

Auf diese Weise können Sie vom Wiederherstellungspunkt aus erneut starten, wobei nur die gesicherten Seitengruppen-Datensätze und die Protokolle von diesem Punkt aus verwendet werden.

Unschärfe Sicherung

Erstellen Sie *Fuzzy* -Sicherungskopien der Seitengruppen, ohne den WS-Manager zu stoppen.

Wenn Sie diese Methode verwenden und die zugehörigen Protokolle später beschädigt werden oder verloren gehen, können Sie die unscharfen Seitengruppe Sicherungskopien nicht mehr verwenden, um die Wiederherstellung zu erhalten. Dies liegt daran, dass die Sicherungskopien der unscharfen Seitengruppe eine inkonsistente Sicht des Status des Warteschlangenmanagers enthalten und von den verfügbaren Protokollen abhängig sind. Wenn die Protokolle nicht verfügbar sind, müssen Sie zur letzten Gruppe von Sicherungsseitenkopien zurückkehren, die ausgeführt wurden, während das Subsystem inaktiv war ([Methode 1](#)), und den Verlust der Daten ab diesem Zeitpunkt akzeptieren.

Methode 1: Gesamtsicherung

Bei dieser Methode wird der WS-Manager heruntergefahren. Dadurch werden alle Aktualisierungen auf die Seitengruppen erzwungen, sodass sich die Seitengruppen in einem konsistenten Zustand befinden.

1. Stoppen Sie alle IBM MQ-Anwendungen, die den Warteschlangenmanager verwenden (ermöglichen Sie deren vorherige Beendigung). Dies kann beispielsweise durch die Änderung der Zugriffsschutz- oder Warteschlangeneinstellungen erfolgen.
2. Wenn alle Aktivitäten abgeschlossen sind, können Sie alle unbestätigkeitsbezogenen Arbeitseinheiten anzeigen und auflösen. (Verwenden Sie die Befehle `DISPLAY CONN` und `RESOLVE INDOUBT`, wie in `DISPLAY CONN` und `RESOLVE INDOUBT` beschrieben.)

Dadurch wird die Seite in einen konsistenten Status versetzt. Wenn dies nicht der Fall ist, sind Ihre Seitengruppen möglicherweise nicht konsistent, und Sie führen effektiv eine unscharfe Sicherung durch.

3. Geben Sie den Befehl `ARCHIVE LOG` aus, um sicherzustellen, dass die neuesten Protokolldaten in die Protokolldatengruppen geschrieben werden.
4. Geben Sie den Befehl `STOP QMGR MODE (QUIESCE)` aus. Notieren Sie den niedrigsten RBA-Wert in den Nachrichten `CSQI024I` oder `CSQI025I` (siehe [CSQI024I](#) und [CSQI025I](#) für weitere Informationen). Sie sollten die Protokolldatensätze von der aus, die durch den RBA-Wert angegeben wird, bis zur aktuellen Protokolldatengruppe beginnen.
5. Erstellen Sie Sicherungskopien aller Seitengruppen des Warteschlangenmanagers (siehe [„Seitengruppen sichern“](#) auf Seite 411).

Methode 2: Unschärfe Sicherung

Bei dieser Methode wird der WS-Manager nicht heruntergefahren. Aus diesem Grund können sich Aktualisierungen während des Sicherungsprozesses in virtuellen Speicherpuffern befinden. Dies bedeutet, dass sich die Seitengruppen nicht in einem konsistenten Zustand befinden und nur für die Wiederherstellung mit den Protokollen verwendet werden können.

1. Geben Sie den Befehl `DISPLAY USAGE TYPE (ALL)` aus und notieren Sie den RBA-Wert in den Nachrichten `CSQI024I` oder `CSQI025I` (siehe [CSQI024I](#) und [CSQI025I](#) für weitere Informationen).
2. Erstellen Sie Sicherungskopien der Seitengruppen (siehe [„Seitengruppen sichern“](#) auf Seite 411).
3. Geben Sie den Befehl `ARCHIVE LOG` aus, um sicherzustellen, dass die neuesten Protokolldaten in die Protokolldatengruppen geschrieben werden. Um den Wiederherstellungspunkt erneut zu

starten, müssen Sie die Protokolldatensätze ausgehend von der Protokolldatengruppe, die durch den RBA-Wert angegeben ist, bis zur aktuellen Protokolldatengruppe beibehalten.

Seitengruppen sichern

Um eine Seitengruppe wiederherzustellen, muss IBM MQ wissen, wie weit zurück im Protokoll es gehen muss. IBM MQ verwaltet eine RBA-Protokollnummer in Seite null jeder Seitengruppe, die als *Wiederherstellungsprotokollfolgenummer* (LSN) bezeichnet wird. Diese Nummer ist die Anfangs-RBA im Protokoll, ab der IBM MQ die Seitengruppe wiederherstellen kann. Wenn Sie eine Seitengruppe sichern, wird diese Nummer ebenfalls kopiert.

Wenn die Kopie später zum Wiederherstellen der Seitengruppe verwendet wird, muss IBM MQ Zugriff auf alle Protokollsätze ab diesem RBA-Wert bis zum aktuellen RBA-Wert haben. Dies bedeutet, dass genügend Protokollsätze vorhanden sein müssen, damit IBM MQ die Wiederherstellung ab der ältesten Sicherungskopie einer Seitengruppe, die aufbewahrt werden soll, durchführen kann.

Verwenden Sie die Funktion `ADRDSSU COPY`, um die Seitengruppen zu kopieren.

Weitere Informationen finden Sie in der Dokumentation zu [COPY DATASET-Befehlssyntax für logisches Dataset](#).

Beispiel:

```
//STEP2 EXEC PGM=ADRDSSU,REGION=6M
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
COPY -
DATASET(INCLUDE(SCENDATA.MQPA.PAGESET.*)) -
RENAMEU(SCENDATA.MQPA.PAGESET.***,SCENDATA.MQPA.BACKUP1.***) -
SPHERE -
REPUNC -
FASTREPLICATION(PREF )-
CANCELERROR -
TOL(ENQF)
/*
//
```

Wenn Sie die Seitengruppe kopieren, während der Warteschlangenmanager ausgeführt wird, müssen Sie ein Kopierdienstprogramm verwenden, das zuerst die Seite null der Seitengruppe kopiert. Wenn Sie dies nicht tun, können Sie die Daten in Ihrer Seitengruppe beschädigen.

Wenn der Prozess der dynamischen Erweiterung einer Seitengruppe unterbrochen wird, z. B., wenn das System verloren geht, können Sie `ADRDSSU` trotzdem verwenden, um eine Sicherung einer Seitengruppe zu erstellen.

Wenn Sie einen `IDCAMS`-Befehl `LISTCAT ENT('page set data set name') ALLOC` für Access Method Services ausführen, wird angezeigt, dass der Wert für `HI-ALLOC-RBA` höher ist als der Wert für `HI-USED-RBA`.

Wenn diese Seitengruppe das nächste Mal gefüllt wird, wird sie, wenn möglich, erneut erweitert, und die Seiten zwischen der hoch verwendeten RBA und der höchstzugeordneten RBA werden zusammen mit einem anderen neuen Extent verwendet.

Objektdefinitionen sichern

Sie sollten auch Kopien Ihrer Objektdefinitionen sichern. Verwenden Sie dazu die Funktion `MAKEDEF` der Funktion `"CSQUTIL COMMAND"` (in [Befehle an IBM MQ \(COMMAND\)](#) ausgegeben beschrieben).

Sichern Sie Ihre Objektdefinitionen immer dann, wenn Sie eine Sicherungskopie des Warteschlangenmanagers erstellen und die aktuelle Version beibehalten.

Seitengruppen wiederherstellen

Wenn der Warteschlangenmanager aufgrund eines Fehlers beendet wurde, kann der Warteschlangenmanager normalerweise erneut gestartet werden, wobei die gesamte Wiederherstellung während des Neustarts ausgeführt wird. Eine solche Wiederherstellung ist jedoch nicht möglich, wenn eine Ihrer Seitengruppen oder Protokolldatengruppen nicht verfügbar ist. Der Grad der Wiederherstellung hängt von der Verfügbarkeit von Sicherungskopien von Seitengruppen und Protokoll Datensätzen ab.

Um einen Wiederanlauf von einem Wiederherstellungspunkt aus zu starten, müssen folgende Schritte erforderlich

- Eine Sicherungskopie der Seitengruppe, die wiederhergestellt werden soll.
- Wenn Sie den Sicherungsprozess "mit grober Übereinstimmung" verwendet haben, der unter „Methode 2: Unschärfe Sicherung“ auf Seite 410 beschrieben wird, die Protokolldatei, die den aufgezeichneten RBA-Wert enthält, die Protokolldatei, die mit dem Befehl ARCHIVE LOG erstellt wurde, und alle Protokolldateien zwischen diesen.
- Wenn Sie die Gesamtsicherung verwendet haben, aber die Protokoll Datensätze nicht nach dem ARCHIVE LOG-Befehl ausgeführt werden, müssen Sie **nicht** die Funktion FORMAT TYPE (REPLACE) des Dienstprogramms CSQUTIL für alle Seitengruppen ausführen.

Wenn Sie eine Seitengruppe in ihrem aktuellen Status wiederherstellen möchten, müssen Sie seit dem Befehl ARCHIVE LOG auch alle Protokoll Datensätze und -datensätze haben.

Es gibt zwei Methoden für die Wiederherstellung einer Seitengruppe. Um eine der beiden Methoden verwenden zu können, muss der WS-Manager gestoppt werden.

Einfache Wiederherstellung

Dies ist die einfachere Methode und ist für die meisten Wiederherstellungssituationen geeignet.

1. Löschen Sie die Seitengruppe, die Sie aus der Sicherung wiederherstellen möchten.
2. Verwenden Sie die Funktion ADRDSSU COPY, um die Seitengruppe aus der Sicherungskopie wiederherzustellen.

Alternativ können Sie die Sicherungskopie in den ursprünglichen Namen umbenennen oder die DD-Anweisung CSQP00xx in Ihrer WS-Managerprozedur ändern, um auf die Sicherungsseite zu verweisen. Wenn Sie dann die Seitengruppe verlieren oder beschädigen, haben Sie keine Sicherungskopie mehr zum Zurückschreiben von.

3. Starten Sie den Warteschlangenmanager erneut.
4. Wenn der WS-Manager erfolgreich erneut gestartet wurde, können Sie Ihre Anwendungen erneut starten.
5. Setzen Sie die normalen Sicherungsprozeduren für die zurückgeschriebene Seite erneut ein.

Erweiterte Wiederherstellung

Diese Methode bietet Leistungsvorteile, wenn eine große Seitengruppe wiederhergestellt werden soll oder wenn seit der letzten Sicherungskopie viel Aktivität auf der Seitengruppe ausgeführt wurde. Sie erfordert jedoch einen manuellen Eingriff als die einfache Methode, die das Risiko eines Fehlers und die Zeit, die für die Wiederherstellung benötigt wird, erhöhen kann.

1. Löschen Sie die Seitengruppe, die Sie aus der Sicherung wiederherstellen möchten, und definieren Sie sie erneut.
2. Kopieren Sie mit ADRDSSU die Sicherungskopie der Seitengruppe in die neue Seitengruppe. Definieren Sie die neue Seitengruppe mit einem sekundären Speicherbereichswert, so dass sie dynamisch erweitert werden kann.

Alternativ können Sie die Sicherungskopie in den ursprünglichen Namen umbenennen oder die DD-Anweisung CSQP00xx in Ihrer WS-Managerprozedur ändern, um auf die Sicherungsseite zu verweisen. Wenn Sie dann die Seitengruppe verlieren oder beschädigen, haben Sie keine Sicherungskopie mehr zum Zurückschreiben von.

3. Ändern Sie die CSQINP1-Definitionen für Ihren Warteschlangenmanager so, dass der Pufferpool, der der Seitengruppe zugeordnet ist, so groß wie möglich wiederhergestellt wird. Wenn Sie den

Pufferpool groß machen, können Sie möglicherweise alle geänderten Seiten im Pufferpool behalten und die Anzahl der Ein-/Ausgaben auf die Seitengruppe reduzieren.

4. Starten Sie den Warteschlangenmanager erneut.
5. Wenn der Warteschlangenmanager erfolgreich erneut gestartet wurde, stoppen Sie ihn (unter Verwendung von Quiesce) und starten Sie ihn anschließend mit Hilfe der normalen Pufferpooldefinition für diese Seitengruppe erneut. Nach erfolgreichem Abschluss dieses zweiten Neustarts können Sie Ihre Anwendungen erneut starten.
6. Setzen Sie die normalen Sicherungsprozeduren für die zurückgeschriebene Seite erneut ein.

Was geschieht, wenn der WS-Manager erneut gestartet wird

Wenn der Warteschlangenmanager erneut gestartet wird, wendet er alle Änderungen an, die an der Seitengruppe vorgenommen wurden, die im Protokoll registriert sind, und beginnt mit dem Neustartpunkt für die Seitengruppe. IBM MQ kann auf diese Weise mehrere Seitengruppen wiederherstellen. Die Seitengruppe wird, falls erforderlich, während der Datenträgerwiederherstellung dynamisch erweitert.

Während des Neustarts bestimmt IBM MQ die Protokoll-RBA, indem es den niedrigsten Wert aus folgenden Werten nimmt:

- Die Protokollfolgennummer für die Wiederherstellung aus dem Prüfpunktprotokollsatz für jede Seitengruppe.
- Recovery-LSN von Seite null in jeder Seitengruppe.
- Der RBA der ältesten unvollständigen Einheit der Wiederherstellung im System zu dem Zeitpunkt, zu dem die Sicherung ausgeführt wurde.

Alle Objektdefinitionen werden auf der Seitengruppe Null gespeichert. Nachrichten können auf jeder verfügbaren Seitengruppe gespeichert werden.

Anmerkung: Der WS-Manager kann nicht erneut gestartet werden, wenn die Seitengruppe Null nicht verfügbar ist.

So löschen Sie Seitengruppen

Sie löschen eine Seitengruppe, indem Sie den Befehl DELETE PSID verwenden; weitere Informationen zu diesem Befehl finden Sie in [DELETE PSID](#).

Eine Seitengruppe, die noch von einer Speicherklasse referenziert wird, kann nicht gelöscht werden. Verwenden Sie DISPLAY STGCLASS, um zu ermitteln, welche Speicherklassen auf eine Seitengruppe verweisen.

Die Datei wird von IBM MQ freigegeben, aber nicht gelöscht. Sie bleibt für die zukünftige Verwendung verfügbar oder kann mithilfe von z/OS-Funktionen gelöscht werden.

Entfernen Sie die Seitengruppe aus der gestarteten Taskprozedur für Ihren Warteschlangenmanager.

Entfernen Sie die Definition der Seitengruppe aus der Initialisierungsdatei CSQINP1.

Vorgehensweise zum Sichern und Zurückschreiben von Warteschlangen mit CSQUTIL

Verwenden Sie dieses Thema als Referenz, um weitere Informationen zum Sichern und Wiederherstellen mit CSQUTIL zu erhalten.

Sie können die Dienstprogrammfunktionen CSQUTIL zum Sichern und Zurückschreiben von Warteschlangen verwenden. Um eine Warteschlange zu sichern, verwenden Sie die Funktion COPY oder SCOPY, um die Nachrichten aus einer Warteschlange in eine Datei zu kopieren. Verwenden Sie die ergänzende Funktion LOAD oder SLOAD, um die Warteschlange wiederherzustellen. Weitere Informationen finden Sie im Abschnitt [IBM MQ-Dienstprogramm](#).

Pufferpools verwalten

Verwenden Sie dieses Thema, wenn Sie Ihre Pufferpools ändern oder löschen wollen.

In diesem Abschnitt wird beschrieben, wie Pufferpools geändert und gelöscht werden. Er enthält folgende Abschnitte:

- „[Wie kann die Anzahl der Puffer in einem Pufferpool geändert werden?](#)“ auf Seite 414
- „[Vorgehensweise zum Löschen eines Pufferpools](#)“ auf Seite 415

Pufferpools werden während der Initialisierung des Warteschlangenmanagers definiert, wobei die Befehle `DEFINE BUFFPOOL`, die von der Initialisierungseingabedatei `CSQINP1` ausgegeben werden, verwendet werden. Ihre Attribute können in Abhängigkeit von den Geschäftsanforderungen geändert werden, während der Warteschlangenmanager ausgeführt wird, wobei die in diesem Thema beschriebenen Prozesse verwendet werden. Der WS-Manager zeichnet die aktuellen Pufferpoolattribute in Prüfpunktprotokollsätzen auf. Diese werden bei einem nachfolgenden Neustart des Warteschlangenmanagers automatisch wiederhergestellt, es sei denn, die Pufferpooldefinition in `CSQINP1` enthält das Attribut `REPLACE`.

Verwenden Sie den Befehl `DISPLAY USAGE`, um die aktuellen Pufferattribute anzuzeigen.

Sie können Pufferpools auch dynamisch mit dem Befehl `DEFINE PSID` mit der DSN-Option definieren.

Wenn Sie Pufferpools dynamisch ändern, müssen Sie auch ihre Definitionen in der Initialisierungsdatei `CSQINP1` aktualisieren.

Eine Beschreibung der Seitengruppen, Speicherklassen, Puffer und Pufferpools sowie einiger Leistungsaspekte, die zu beachten sind, finden Sie unter [Planung für z/OS](#).

Anmerkung: Pufferpools verwenden einen signifikanten Speicher. Wenn Sie die Größe eines Pufferpools erhöhen oder einen neuen Pufferpool definieren, stellen Sie sicher, dass genügend Speicher verfügbar ist. Weitere Informationen finden Sie im Abschnitt [Speichervoraussetzungen für Adressräume](#).

Wie kann die Anzahl der Puffer in einem Pufferpool geändert werden?

Wenn ein Pufferpool zu klein ist, kann die Bedingung in der Nachricht `CSQP020E` auf der Konsole ausgegeben werden. Sie können mit dem Befehl `ALTER BUFFPOOL` wie folgt weitere Puffer zuordnen:

1. Ermitteln Sie, wie viel Speicherplatz für neue Puffer verfügbar ist, indem Sie die Nachrichten `CSQY220I` im Protokoll anzeigen. Der verfügbare Speicherbereich wird in MB angegeben. Da ein Puffer eine Größe von 4 KB hat, ermöglicht es jedem MB verfügbaren Speicherplatz, 256 Puffer zuzuordnen. Zuordnen Sie nicht den gesamten freien Speicherbereich zu Puffern, da einige für andere Tasks erforderlich sind.

Wenn der Pufferpool feste 4-KB-Seiten verwendet, also das Attribut `PAGECLAS FIXED4KB` ist, stellen Sie sicher, dass auf der LPAR genügend Realspeicher zur Verfügung steht.

2. Wenn der zurückgemeldete freie Speicherbereich unzureichend ist, geben Sie mithilfe des Befehls einige Puffer aus einem anderen Pufferpool frei.

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

Dabei ist *buf-pool-id* der Pufferpool, aus dem der Speicherbereich wiederhergestellt werden soll, und *integer* ist die neue Anzahl Puffer, die diesem Pufferpool zugeordnet werden sollen, die kleiner als die ursprüngliche Anzahl Puffer sein muss, die dem Pufferpool zugeordnet sind.

3. Fügen Sie dem Pufferpool Puffer hinzu, die mit dem Befehl erweitert werden sollen.

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

Hierbei steht *buf-pool-id* für den zu erweiterbaren Pufferpool und *integer* für die neue Anzahl Puffer, die diesem Pufferpool zugeordnet werden sollen. Dieser Wert muss größer sein als die ursprüngliche Anzahl Puffer, die ihm zugeordnet sind.

Vorgehensweise zum Löschen eines Pufferpools

Wenn ein Pufferpool nicht mehr von den Seitengruppen verwendet wird, löschen Sie ihn, um den virtuellen Speicher freizugeben, der ihm zugeordnet ist.

Sie löschen einen Pufferpool mit dem Befehl `DELETE BUFFPOOL`. Der Befehl schlägt fehl, wenn alle Seitengruppen diesen Pufferpool verwenden.

Informationen zum Löschen von Seitengruppen finden Sie im Abschnitt [„So löschen Sie Seitengruppen“](#) auf Seite 413.

Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzte Warteschlangen unter z/OS verwalten

IBM MQ kann verschiedene Typen gemeinsam genutzter Ressourcen verwenden, z. B. Gruppen mit gemeinsamer Warteschlange, gemeinsam genutzte Warteschlangen und die Coupling-Facility. Verwenden Sie dieses Thema, um die Prozeduren zu überprüfen, die für die Verwaltung dieser gemeinsam genutzten Ressourcen erforderlich sind.

Dieser Abschnitt enthält Informationen zu den folgenden Themen:

- [„Gruppen mit gemeinsamer Warteschlange verwalten“](#) auf Seite 415
- [„Gemeinsam genutzte Warteschlangen verwalten“](#) auf Seite 419
- [„Gruppenobjekte verwalten“](#) auf Seite 424
- [„Coupling Facility verwalten“](#) auf Seite 424

Gruppen mit gemeinsamer Warteschlange verwalten

Sie können einen Warteschlangenmanager zu einer Gruppe mit gemeinsamer Warteschlange (Queue Sharing Group, QSG) hinzufügen oder daraus entfernen und Sie können die zugehörigen Db2-Tabellen verwalten.

In diesem Abschnitt finden Sie Abschnitte zu den folgenden Tasks:

- [„Gruppe mit gemeinsamer Warteschlange einrichten“](#) auf Seite 415
- [„Warteschlangenmanager zur Gruppe mit gemeinsamer Warteschlange hinzufügen“](#) auf Seite 416
- [„Warteschlangenmanager aus einer Gruppe mit gemeinsamer Warteschlange entfernen“](#) auf Seite 417
- [„Gruppe mit gemeinsamer Warteschlange aus den DB2-Tabellen entfernen“](#) auf Seite 418
- [„Prüfen der Konsistenz von DB2-Definitionen“](#) auf Seite 419

Gruppe mit gemeinsamer Warteschlange einrichten

Jede Gruppe mit gemeinsamer Warteschlange hat einen Namen, der aus bis zu vier Zeichen besteht. Der Name muss in Ihrem Netz eindeutig sein und muss sich von allen WS-Managernamen unterscheiden.

Führen Sie die folgenden Schritte aus, um eine Gruppe mit gemeinsamer Warteschlange zu konfigurieren:

1. Wenn es sich um die erste Gruppe mit gemeinsamer Warteschlange handelt, die die Db2-Gruppe mit gemeinsamer Datennutzung verwendet, [richten Sie die Db2-Umgebung ein](#).
2. [Konfigurieren Sie die Coupling-Facility](#).
3. Fügen Sie die Gruppe mit gemeinsamer Warteschlange den Db2-Tabellen hinzu. Verwenden Sie die Funktion `ADD QSG` des Dienstprogramms für die Gruppe mit gemeinsamer Warteschlange (`CSQ5PQSG`). Dieses Programm wird im Abschnitt [Dienstprogramm für die Gruppe mit gemeinsamer Warteschlange](#) beschrieben. Eine Probe wird in `thlqual.SCSQPROC (CSQ45AQS)` bereitgestellt.
4. Fügen Sie der Gruppe mit gemeinsamer Warteschlange einen Warteschlangenmanager hinzu, indem Sie die Schritte in [„Warteschlangenmanager zur Gruppe mit gemeinsamer Warteschlange hinzufügen“](#) auf Seite 416 ausführen.

5. Definieren Sie Anwendungsstrukturen für IBM MQ mithilfe der Schritte in „Coupling-Facility-Struktur hinzufügen“ auf Seite 425.
6. Falls erforderlich, migrieren Sie nicht gemeinsam genutzte Warteschlangen in gemeinsam genutzte Warteschlangen.
7. Erstellen Sie für die Verfügbarkeit gemeinsam genutzte Kanäle in die Gruppe mit gemeinsamer Warteschlange und aus der Gruppe mit gemeinsamer Warteschlange.
 - Für Verbindungen in die Gruppe mit gemeinsamer Warteschlange:
 - Richten Sie einen VIPA-Socket oder einen Hardware-Router ein, um die Auslastung zwischen den verfügbaren Warteschlangenmanagern in der QSG zu verteilen.
 - Definieren Sie einen Empfängerkanal mit QSGDISP (GROUP), um sicherzustellen, dass die Kanaldefinition auf allen Warteschlangenmanagern in der QSG verfügbar ist.
 - Starten Sie einen Listener mit INDISP (GROUP) auf jedem WS-Manager für MCA-Kanalverbindungen in die QSG. Clientverbindungen in die QSG sollten immer noch eine Verbindung zu einem Empfangsprogramm herstellen, das mit INDISP (QMGR) gestartet wurde.
 - Anwendungen so ändern, dass sie mit dem QSG-Namen und nicht mit einem bestimmten WS-Manager-Namen verbunden werden.
 - Stellen Sie sicher, dass die Regeln für die Kanalauthentifizierung auf allen WS-Managern in der QSG identisch sind, damit Anwendungen eine Verbindung zu jedem WS-Manager in der QSG herstellen können.
 - Für Verbindungen aus der Gruppe mit gemeinsamer Warteschlange:
 - Definieren Sie eine gemeinsam genutzte Übertragungswarteschlange.
 - Definieren Sie den abgehenden Kanal mit QSGDISP (GROUP) und DEFCDISP (SHARED).

Wenn Sie einen vorhandenen Kanal in einen gemeinsam genutzten Kanal konvertieren, müssen Sie möglicherweise den Befehl RESET CHANNEL ausgeben, bevor Sie den Kanal starten, da sich die vom Kanal verwendete Synchronisationswarteschlange geändert hat.

Warteschlangenmanager zur Gruppe mit gemeinsamer Warteschlange hinzufügen

Ein Warteschlangenmanager kann einer vorhandenen Gruppe mit gemeinsamer Warteschlange hinzugefügt werden.

Dabei ist Folgendes zu beachten:

- Die Gruppe mit gemeinsamer Warteschlange muss vorhanden sein, bevor Sie ihr Warteschlangenmanager hinzufügen können.
- Ein Warteschlangenmanager kann nur Mitglied einer Gruppe mit gemeinsamer Warteschlange sein.

Führen Sie die folgenden Schritte aus, um einen WS-Manager zu einer Gruppe mit gemeinsamer Warteschlange hinzuzufügen:

1. Führen Sie die Tasks im Abschnitt ESM-Sicherheitssteuerelemente für die Gruppe mit gemeinsamer Warteschlange implementieren aus, um den entsprechenden Zugriff auf die Benutzer-IDs des Warteschlangenmanagers und des Kanalinitiators zu erteilen.
2. Wenn die Gruppe mit gemeinsamer Warteschlange CF-Strukturen hat, die so konfiguriert sind, dass Daten in SMDS ausgelagert werden, führen Sie die Tasks in SMDS-Umgebung einrichten aus.
3. Stoppen Sie den Warteschlangenmanager.
4. Verwenden Sie die Funktion ADD QMGR des Dienstprogramms für die Gruppe mit gemeinsamer Warteschlange (CSQ5PQSG). Dieses Programm wird im Abschnitt Dienstprogramm für die gemeinsame Nutzung von Gruppen mit gemeinsamer Warteschlange beschrieben. Ein Beispiel wird in thlqual.SCSQPROC (CSQ45AQM) bereitgestellt.
5. Ändern Sie Ihr Systemparametermodul, um Daten der Gruppe mit gemeinsamer Warteschlange hinzuzufügen:

- a. Ändern Sie CSQ6SYSP, um den Parameter QSGDATA anzugeben. Weitere Informationen finden Sie im Abschnitt [CSQ6SYSP](#) verwenden.
 - b. Assemblieren und verknüpfen Sie das Systemparametermodul. Es kann sein, dass Sie einen anderen Namen für das Lademodul verwenden möchten.
 - c. Ändern Sie den Startprozess so, dass er das neue Modul verwendet.
6. Kopieren und passen Sie das Beispielmemberr hlqual.SCSQPROC (CSQ4INSS) an, das die erforderlichen CF-Strukturen und SYSTEM-Warteschlangen definiert. Fügen Sie das angepasste Member der CSQINP2-DD in der Start-JCL des Warteschlangenmanagers hinzu.
 7. Starten Sie den Warteschlangenmanager mit Hilfe des Systemparametermoduls der Gruppe mit gemeinsamer Warteschlange neu.
 8. Optional können Sie die Migration auf Sicherheitsprofile, denen der Name der Gruppe mit gemeinsamer Warteschlange als Präfix vorangestellt ist, anstelle des Namens des Warteschlangenmanagers verwenden.
 9. Wenn gemeinsam genutzte Kanäle für Verbindungen in die QSG verwendet werden, erstellen Sie Kanalauthentifizierungsregeln, die die auf den anderen WS-Managern in der QSG widerspiegeln, damit Anwendungen eine Verbindung zu jedem WS-Manager in der QSG herstellen können.
 10. Wählen Sie optional eines der folgenden Schritte aus, damit Anwendungen, die mit dem Warteschlangenmanager in der QSG verbunden sind, Nachrichten an Warteschlangen stellen können, die von anderen Warteschlangenmanagern in der QSG per Hosting bereitgestellt werden:
 - Aktivieren Sie [Gruppeninterne Warteschlangensteuerung](#) , indem Sie den Befehl ALTER QMGR IGQ (ENABLED) ausgeben.
 - Definieren Sie Übertragungswarteschlangen und Kanäle zu den anderen Warteschlangenmanagern in der QSG. Das Definieren von Übertragungswarteschlangen mit demselben Namen wie die Ziel-WS-Manager vermeidet die Notwendigkeit, ferne Warteschlangen und WS-Manager-Aliasnamen zu definieren.

Anmerkung: Wenn Sie einen Warteschlangenmanager zu einer vorhandenen Gruppe mit gemeinsamer Warteschlange hinzufügen möchten, die Warteschlangenmanager mit früheren Versionen von IBM MQ enthält, müssen Sie zuerst die Koexistenz-PTF für die höchste Version von IBM MQ in der Gruppe auf jeden Warteschlangenmanager der früheren Version in der Gruppe anwenden.

Warteschlangenmanager aus einer Gruppe mit gemeinsamer Warteschlange entfernen

Sie können einen Warteschlangenmanager nur aus einer Gruppe mit gemeinsamer Warteschlange entfernen, wenn die Protokolle des Warteschlangenmanagers nicht von einem anderen Prozess benötigt werden und alle SMDS, deren Eigner der Warteschlangenmanager ist, leer sind.

Die Protokolle werden benötigt, wenn sie Folgendes enthalten:

- Die letzte Sicherung einer der CF-Anwendungsstrukturen, die von der Gruppe mit gemeinsamer Warteschlange verwendet werden.
- Daten, die von einem zukünftigen Wiederherstellungsprozess benötigt werden, d. a. der Warteschlangenmanager hat eine wiederherstellbare Struktur verwendet, da die Zeit, die durch den letzten Wert für das Ausschlussintervall beschrieben wurde, verwendet wurde.

Wenn eine oder beide dieser Punkte oder ein SMDS, dessen Eigner der WS-Manager ist, Nachrichten enthält, kann der Warteschlangenmanager nicht entfernt werden. Verwenden Sie den MQSC-Befehl DISPLAY CFSTATUS mit der Option TYPE (BACKUP), um festzustellen, welche Warteschlangenmanager-Protokolle für einen zukünftigen Wiederherstellungsprozess benötigt werden. (Weitere Informationen zu diesem Befehl finden Sie in [DISPLAY CFSTATUS](#) .)

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager aus einer Gruppe mit gemeinsamer Warteschlange zu entfernen:

1. Stoppen Sie alle Anwendungen, die mit dem Warteschlangenmanager verbunden sind und die Nachrichten in gemeinsam genutzte Warteschlangen stellen.
2. Beheben Sie alle unbestätigten Arbeitseinheiten, an denen dieser Warteschlangenmanager beteiligt ist.
3. Stellen Sie fest, ob Nachrichten in SMDS vorhanden sind, deren Eigner der WS-Manager ist, indem Sie den Befehl DISPLAY USAGE TYPE (SMDS) ausgeben.
4. Wenn für jede Anwendungsstruktur ausgelagerte Nachrichten vorhanden sind, warten Sie, bis diese Nachrichten aus der Warteschlange abgerufen wurden. Die Anzahl der ausgelagerten Nachrichten, die von DISPLAY USAGE TYPE (SMDS) gemeldet werden, sollte null sein, bevor fortgefahren wird.
5. Fahren Sie den WS-Manager mit STOP QMGR MODE (QUIESCE) sauber herunter.
6. Warten Sie auf ein Intervall, das mindestens dem Wert des Parameters EXCLINT entspricht, den Sie im nächsten Schritt mit dem Befehl BACKUP CFSTRUCT angeben.
7. Führen Sie auf einem anderen Warteschlangenmanager eine CF-Struktursicherung für jede wiederherstellbare CF-Struktur aus, indem Sie den MQSC-Befehl BACKUP CFSTRUCT verwenden und einen Wert für EXCLINT angeben, wie im vorherigen Schritt erforderlich.
8. Stellen Sie sicher, dass die Protokolle des WS-Managers nicht benötigt werden, um alle CF-Strukturen wiederherzustellen, indem Sie die Ausgabe des Befehls DISPLAY CFSTATUS (*) TYPE (BACKUP) überprüfen.
9. Verwenden Sie die Funktion REMOVE QMGR des Dienstprogramms CSQ5PQSG, um den Warteschlangenmanager aus der Gruppe mit gemeinsamer Warteschlange zu entfernen. Dieses Programm wird im Abschnitt Dienstprogramm für die Gruppe mit gemeinsamer Warteschlange beschrieben. Ein Beispiel wird in thlqual.SCSQPROC (CSQ45RQM) bereitgestellt.
10. Bevor Sie den Warteschlangenmanager erneut starten, setzen Sie den Systemparameter QSGDATA auf den Standardwert zurück, und erstellen Sie das Systemparametermodul erneut. Informationen zur Anpassung der Systemparameter finden Sie im Abschnitt CSQ6SYSP verwenden.

Beachten Sie, dass Sie beim Entfernen des letzten Warteschlangenmanagers in einer Gruppe mit gemeinsamer Warteschlange die Option FORCE und nicht die Option REMOVE verwenden müssen. Dadurch wird der Warteschlangenmanager aus der Gruppe mit gemeinsamer Warteschlange entfernt, während die Konsistenzprüfungen der Warteschlangenmanagerprotokolle, die für die Wiederherstellung erforderlich sind, nicht ausgeführt werden. Sie sollten diese Operation nur ausführen, wenn Sie die Gruppe mit gemeinsamer Warteschlange löschen.

Gruppe mit gemeinsamer Warteschlange aus den DB2-Tabellen entfernen

Verwenden Sie zum Entfernen einer Gruppe mit gemeinsamer Warteschlange aus den Db2-Tabellen die Funktion REMOVE QSG des Dienstprogramms für die Gruppe mit gemeinsamer Warteschlange (CSQ5PQSG). Dieses Programm wird im Abschnitt Dienstprogramm zur gemeinsamen Nutzung von Gruppen mit gemeinsamer Warteschlange beschrieben. Eine Probe wird in thlqual.SCSQPROC (CSQ45RQS) bereitgestellt.

Sie können eine Gruppe mit gemeinsamer Warteschlange erst dann aus den allgemeinen Db2-Tabellen für die Gruppe mit gemeinsamer Datennutzung entfernen, wenn alle Warteschlangenmanager aus der Gruppe mit gemeinsamer Warteschlange entfernt wurden (siehe „Warteschlangenmanager aus einer Gruppe mit gemeinsamer Warteschlange entfernen“ auf Seite 417).

Wenn der Datensatz für die Gruppe mit gemeinsamer Warteschlange aus der Verwaltungstabelle der Gruppe mit gemeinsamer Warteschlange entfernt wurde, werden alle Objekte und Verwaltungsinformationen, die sich auf diese Gruppe mit gemeinsamer Warteschlange beziehen, aus anderen IBM MQ Db2-Tabellen entfernt. Dies umfasst Informationen zu gemeinsam genutzten Warteschlangen und Gruppenobjekten.

Prüfen der Konsistenz von DB2-Definitionen

Probleme für gemeinsam genutzte Warteschlangen in einer Gruppe mit gemeinsamer Warteschlange können auftreten, wenn die Objektdefinitionen für Db2 aus irgendeinem Grund inkonsistent sind.

Um die Konsistenz der Db2-Objektdefinitionen für Warteschlangenmanager, CF-Strukturen und gemeinsam genutzte Warteschlangen zu überprüfen, verwenden Sie die Funktion VERIFY QSG des Dienstprogramms für die Gruppe mit gemeinsamer Warteschlange (CSQ5PQSG). Dieses Programm wird im Abschnitt [Dienstprogramm für die Gruppe mit gemeinsamer Warteschlange](#) beschrieben.

Gemeinsam genutzte Warteschlangen verwalten

In diesem Thema wird erläutert, wie gemeinsam genutzte Warteschlangen wiederhergestellt, verschoben und migriert werden.

In diesem Abschnitt werden die folgenden Tasks beschrieben:

- [„Gemeinsam genutzte Warteschlangen wiederherstellen“](#) auf Seite 419
- [„Gemeinsam genutzte Warteschlangen verschieben“](#) auf Seite 420
- [„Nicht gemeinsam genutzte Warteschlangen in gemeinsam genutzte Warteschlangen migrieren“](#) auf Seite 422
- [Eine Db2-Verbindung aussetzen](#)

Gemeinsam genutzte Warteschlangen wiederherstellen

IBM MQ kann in folgenden Fällen persistente Nachrichten in gemeinsam genutzten Warteschlangen wiederherstellen:

- Es wurden Backups der CF-Strukturen ausgeführt, die die Nachrichten enthalten.
- Alle Protokolle für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange sind verfügbar, um die Wiederherstellung von dem Zeitpunkt herzustellen, an dem die Backups vorgenommen wurden.
- Db2 ist verfügbar und die Struktursicherungstabelle ist aktueller als die zuletzt verwendete CF-Struktursicherung.

Die Nachrichten in einer gemeinsam genutzten Warteschlange werden in einer Coupling-Facility-Struktur (CF) gespeichert. Persistente Nachrichten können in gemeinsam genutzte Warteschlangen gestellt werden und wie persistente Nachrichten in nicht gemeinsam genutzten Warteschlangen werden sie in das WS-Manager-Protokoll kopiert. Die Befehle MQSC BACKUP CFSTRUCT und RECOVER CFSTRUCT werden bereitgestellt, um die Wiederherstellung einer CF-Struktur im unwahrscheinlichen Fall eines Ausfalls der Coupling Facility zu ermöglichen. Unter solchen Umständen gehen alle nicht persistenten Nachrichten, die in der betroffenen Struktur gespeichert sind, verloren, aber persistente Nachrichten können wiederhergestellt werden. Jede weitere Anwendungsaktivität, die die Struktur verwendet, wird verhindert, bis die Struktur wiederhergestellt wurde.

Um die Wiederherstellung zu aktivieren, müssen Sie die Strukturen der Coupling-Facility-Liste häufig mit dem Befehl MQSC BACKUP CFSTRUCT sichern. Die Nachrichten in der CF-Struktur werden in die aktive Protokolldatei des Warteschlangenmanagers geschrieben, der die Sicherung erstellt. Er wird ein Datensatz der Sicherung in Db2 geschrieben: Name der CF-Struktur, die gesichert wird, Name des Warteschlangenmanagers, der die Sicherung erstellt, RBA-Bereich für diese Sicherung im betreffenden Warteschlangenmanagerprotokoll und die Sicherungszeit. Sichern Sie CF-Listenstrukturen, selbst wenn Sie gemeinsam genutzte Warteschlangen nicht aktiv verwenden, beispielsweise beim Einrichten einer Gruppe mit gemeinsamer Warteschlange für eine spätere Verwendung.

Sie können eine CF-Struktur wiederherstellen, indem Sie den MQSC-Befehl RECOVER CFSTRUCT für den Warteschlangenmanager ausgeben, der die Wiederherstellung ausführen kann. Sie können dafür jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verwenden. Sie können eine einzelne CF-Struktur angeben, die wiederhergestellt werden soll, oder Sie können mehrere CF-Strukturen gleichzeitig wiederherstellen.

Wie bereits erwähnt, ist es wichtig, dass Sie Ihre CF-Listenstrukturen häufig sichern, da andernfalls die Wiederherstellung einer CF-Struktur sehr lange dauern kann. Darüber hinaus kann der Wiederherstellungsprozess nicht abgebrochen werden.

Die Definition einer gemeinsam genutzten Warteschlange wird in einer Db2-Datenbank gespeichert und kann daher bei Bedarf mithilfe von Standarddatenbankprozeduren von Db2 wiederhergestellt werden. Weitere Informationen hierzu finden Sie im Abschnitt [Gemeinsam genutzte Warteschlangen und Gruppen mit gemeinsamer Warteschlange](#).

Gemeinsam genutzte Warteschlangen verschieben

In diesem Abschnitt wird beschrieben, wie der Lastausgleich ausgeführt wird, indem eine gemeinsam genutzte Warteschlange von einer Coupling-Facility-Struktur in eine andere verschoben wird. Darüber hinaus wird beschrieben, wie eine nicht gemeinsam genutzte Warteschlange in eine gemeinsam genutzte Warteschlange verschoben wird und wie eine gemeinsam genutzte Warteschlange in eine nicht gemeinsam genutzte Warteschlange verschoben wird.

Wenn Sie eine Warteschlange verschieben, müssen Sie eine temporäre Warteschlange als Teil der Prozedur definieren. Dies liegt daran, dass jede Warteschlange einen eindeutigen Namen haben muss, sodass Sie nicht zwei Warteschlangen mit demselben Namen haben können, auch wenn die Warteschlangen über unterschiedliche Warteschlangendispositionen verfügen. IBM MQ toleriert zwei Warteschlangen mit demselben Namen (wie in Schritt „2“ auf Seite 420), aber die Warteschlangen können nicht verwendet werden.

- Eine Warteschlange von einer Coupling-Facility-Struktur in eine andere versetzen
- Nicht gemeinsam genutzte Warteschlange in eine gemeinsam genutzte Warteschlange verschieben
- Gemeinsam genutzte Warteschlange in eine nicht gemeinsam genutzte Warteschlange verschieben

Eine Warteschlange von einer Coupling-Facility-Struktur in eine andere versetzen

Verwenden Sie den MQSC-Befehl `MOVE QLOCAL`, um Warteschlangen und ihre Nachrichten von einer CF-Struktur in eine andere zu versetzen. Wenn Sie die Warteschlange oder die Warteschlangen angegeben haben, die Sie in eine neue CF-Struktur verschieben möchten, verwenden Sie die folgende Prozedur, um jede Warteschlange zu verschieben:

1. Stellen Sie sicher, dass die Warteschlange, die Sie verschieben möchten, nicht von anderen Anwendungen verwendet wird, d. h. die Warteschlangenattribute `IPPROCS` und `OPPROCS` haben auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange den Wert null.
2. Verhindern Sie, dass Anwendungen Nachrichten in die Warteschlange einreihen, indem Sie die Warteschlangendefinition ändern, um `MQPUT` zu inaktivieren. Ändern Sie die Warteschlangendefinition in `PUT (DISABLED)`.
3. Definieren Sie eine temporäre Warteschlange mit den gleichen Attributen wie die Warteschlange, die mit dem folgenden Befehl verschoben wird:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

Anmerkung: Wenn diese temporäre Warteschlange von einer vorherigen Ausführung aus vorhanden ist, löschen Sie sie, bevor Sie die Definition ausführen.

4. Verschieben Sie die Nachrichten mit dem folgenden Befehl in die temporäre Warteschlange:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Löschen Sie die Warteschlange, die Sie verschieben möchten, mit dem folgenden Befehl:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Definieren Sie die Warteschlange, die verschoben wird, erneut, und ändern Sie das Attribut CFSTRUCT mit dem folgenden Befehl:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_Queue) CFSTRUCT(NEW) QSGDISP(SHARED)
```

Wenn die Warteschlange neu definiert wird, basiert sie auf der temporären Warteschlange, die in Schritt „3“ auf Seite 420 erstellt wurde.

7. Verschieben Sie die Nachrichten mit dem Befehl zurück in die neue Warteschlange:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

8. Die in Schritt „3“ auf Seite 420 erstellte Warteschlange wird nicht mehr benötigt. Verwenden Sie den folgenden Befehl, um ihn zu löschen:

```
DELETE QL(TEMP_Queue)
```

9. Wenn die Warteschlange, die versetzt wird, in den CSQINP2-Dateien definiert wurde, ändern Sie das Attribut CFSTRUCT des entsprechenden Befehls DEFINE QLOCAL in den CSQINP2-Dateien. Fügen Sie das Schlüsselwort REPLACE hinzu, so dass die vorhandene Warteschlangendefinition ersetzt wird.

Abbildung 47 auf Seite 421 zeigt einen Beispieljob für das Verschieben einer Warteschlange von einer CF-Struktur in eine andere.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
// DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_Queue) PURGE
DEFINE QL(TEMP_Queue) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_Queue)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(TEMP_Queue) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_Queue) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_Queue)
/*
```

Abbildung 47. Beispieljob zum Versetzen einer Warteschlange von einer CF-Struktur in eine andere

Nicht gemeinsam genutzte Warteschlange in eine gemeinsam genutzte Warteschlange verschieben

Die Prozedur zum Verschieben einer nicht gemeinsam genutzten Warteschlange in eine gemeinsam genutzte Warteschlange entspricht der Prozedur zum Verschieben einer Warteschlange von einer CF-Struktur in eine andere (siehe „Eine Warteschlange von einer Coupling-Facility-Struktur in eine

andere versetzen” auf Seite 420). [Abbildung 48 auf Seite 422](#) zeigt einen Beispieljob für diesen Vorgang.

Anmerkung: Beachten Sie, dass Nachrichten in gemeinsam genutzten Warteschlangen bestimmten Einschränkungen in Bezug auf die maximale Nachrichtengröße, die Nachrichtenpersistenz und den Warteschlangenindextyp unterliegen, sodass Sie möglicherweise nicht in der Lage sind, einige nicht gemeinsam genutzte Warteschlangen in eine gemeinsam genutzte Warteschlange zu verschieben.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*
```

Abbildung 48. Beispieljob zum Versetzen einer nicht gemeinsam genutzten Warteschlange in eine gemeinsam genutzte Warteschlange

Gemeinsam genutzte Warteschlange in eine nicht gemeinsam genutzte Warteschlange verschieben

Die Prozedur zum Verschieben einer gemeinsam genutzten Warteschlange in eine nicht gemeinsam genutzte Warteschlange entspricht der Prozedur zum Verschieben einer Warteschlange von einer CF-Struktur in eine andere (siehe „[Eine Warteschlange von einer Coupling-Facility-Struktur in eine andere versetzen](#)” auf Seite 420).

[Abbildung 49 auf Seite 422](#) zeigt einen Beispieljob für diesen Vorgang.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW) QSGDISP(QMGR)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*
```

Abbildung 49. Beispieljob zum Versetzen einer gemeinsam genutzten Warteschlange in eine nicht gemeinsam genutzte Warteschlange

Nicht gemeinsam genutzte Warteschlangen in gemeinsam genutzte Warteschlangen migrieren

Es gibt zwei Phasen, in die nicht gemeinsam genutzte Warteschlangen in gemeinsam genutzte Warteschlangen migriert werden:

- Migration des ersten (oder einzigen) Warteschlangenmanagers in der Gruppe mit gemeinsamer Warteschlange
- Migration aller anderen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange

Migration des ersten (oder einzigen) Warteschlangenmanagers in der Gruppe mit gemeinsamer Warteschlange

Abbildung 48 auf Seite 422 zeigt einen Beispieljob für das Verschieben einer nicht gemeinsam genutzten Warteschlange in eine gemeinsam genutzte Warteschlange dargestellt. Führen Sie diese Schritte für jede Warteschlange aus, die migriert werden muss.

Anmerkung:

1. Nachrichten in gemeinsam genutzten Warteschlangen unterliegen gewissen Einschränkungen hinsichtlich der maximalen Nachrichtengröße, der Nachrichtenpersistenz und des Warteschlangenindextyps, sodass Sie möglicherweise nicht in der Lage sind, einige nicht gemeinsam genutzte Warteschlangen in eine gemeinsam genutzte Warteschlange zu verschieben.
2. Sie müssen den korrekten Indextyp für gemeinsam genutzte Warteschlangen verwenden. Wenn Sie eine Übertragungswarteschlange in eine gemeinsam genutzte Warteschlange migrieren, muss der Indextyp MSGID sein.

Wenn die Warteschlange leer ist oder Sie die Nachrichten nicht behalten müssen, ist die Migration der Warteschlange einfacher. Abbildung 50 auf Seite 423 zeigt einen Beispieljob zur Verwendung unter diesen Bedingungen.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
// DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(QUEUE_TO_MOVE) PUT(ENABLED) GET(ENABLED)
DELETE QL(QUEUE_TO_MOVE)
DEFINE QL(QUEUE_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
DELETE QL(TEMP_QUEUE)
/*
```

Abbildung 50. Beispieljob zum Versetzen einer nicht gemeinsam genutzten Warteschlange ohne Nachrichten in eine gemeinsam genutzte Warteschlange

Migration aller anderen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange

1. Verschieben Sie jede Warteschlange, die nicht denselben Namen wie eine bereits vorhandene gemeinsam genutzte Warteschlange hat, wie in Abbildung 48 auf Seite 422 oder Abbildung 50 auf Seite 423 beschrieben.
2. Verschieben Sie Nachrichten aus Warteschlangen, die denselben Namen wie eine bereits vorhandene gemeinsam genutzte Warteschlange haben, mithilfe der in Abbildung 51 auf Seite 423 gezeigten Befehle in die gemeinsam genutzte Warteschlange.

```
MOVE QLOCAL(QUEUE_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(QUEUE_TO_MOVE)
DELETE QLOCAL(QUEUE_TO_MOVE) QSGDISP(QMGR)
```

Abbildung 51. Nachrichten aus einer nicht gemeinsam genutzten Warteschlange in eine vorhandene gemeinsam genutzte Warteschlange versetzen

Verbindung zu Db2 aussetzen

Wenn Sie Wartungs- oder Servicemaßnahmen für Db2-Tabellen oder -Pakete, die sich auf gemeinsam genutzte Warteschlangen beziehen, durchführen möchten, ohne Ihren Warteschlangenmanager zu stoppen, müssen Sie die Warteschlangenmanager in der Gruppe mit gemeinsamer Datennutzung (DSG) von Db2 trennen.

Gehen Sie dazu wie folgt vor:

1. Verwenden Sie den MQSC-Befehl SUSPEND QMGR FACILITY(Db2).
2. Führen Sie die Bindungen aus.
3. Stellen Sie die Verbindung zu Db2 mit dem MQSC-Befehl RESUME QMGR FACILITY(Db2) wieder her.

Beachten Sie, dass es Einschränkungen bei der Verwendung dieser Befehle gibt.



Achtung: Während die Db2-Verbindung ausgesetzt ist, sind die folgenden Operationen nicht verfügbar. Daher müssen Sie diese Arbeit in einem Zeitraum ausführen, in dem Ihr Unternehmen am wenigsten ausgelastet ist.

- Zugriff auf gemeinsam genutzte Warteschlangenobjekte für die Verwaltung (definieren, löschen, ändern)
- Gemeinsam genutzte Kanäle starten
- Nachrichten in Db2 speichern
- CFSTRUCT sichern oder wiederherstellen

Gruppenobjekte verwalten

In diesem Thema wird erläutert, wie mit Gruppenobjekten gearbeitet wird.

IBM MQ kopiert die Definition eines Gruppenobjekts automatisch in die Seitengruppe null jedes Warteschlangenmanagers, der sie verwendet. Sie können die Kopie der Definition vorübergehend ändern, und IBM MQ ermöglicht Ihnen das Aktualisieren der Seitengruppenkopien aus der Repository-Kopie. IBM MQ versucht immer, die Seitengruppenkopien beim Start von der Repository-Kopie zu aktualisieren (bei Kanalobjekten wird dies bei Neustarts des Kanalinitiators ausgeführt). Dadurch wird sichergestellt, dass die Kopien der Seitengruppe die Version im Repository widerspiegeln, einschließlich aller Änderungen, die beim Inaktiven des Warteschlangenmanagers vorgenommen wurden.

Es gibt Umstände, unter denen die Aktualisierung nicht ausgeführt wird, z. B.:

- Wenn eine Kopie der Warteschlange geöffnet ist, schlägt eine Aktualisierung fehl, die die Verwendung der Warteschlange ändern würde.
- Wenn eine Kopie einer Warteschlange Nachrichten enthält, schlägt eine Aktualisierung fehl, die diese Warteschlange löschen würde.

Unter diesen Umständen wird die Aktualisierung nicht für diese Kopie ausgeführt, sondern wird in den Kopien auf allen anderen Warteschlangenmanagern ausgeführt. Nach dem Hinzufügen, Ändern oder Löschen eines Gruppenobjekts und beim Neustart des Warteschlangenmanagers oder Kanalinitiators können Sie alle Probleme mit dem Kopierobjekt überprüfen und korrigieren.

Coupling Facility verwalten

In diesem Thema wird erläutert, wie Coupling-Facility-Strukturen (CF-Coupling Facility) hinzugefügt oder entfernt werden.

In diesem Abschnitt werden die folgenden Tasks beschrieben:

- „Coupling-Facility-Struktur hinzufügen“ auf Seite 425
- „Entfernen einer Coupling-Facility-Struktur“ auf Seite 425

Coupling-Facility-Struktur hinzufügen

Führen Sie die folgende Prozedur aus, um eine Coupling-Facility-Struktur hinzuzufügen:

1. Definieren Sie die CF-Struktur in der CFRM-Richtliniendatei. Die Informationen zum Einrichten der Coupling-Facility in [Coupling-Facility einrichten](#) beschreiben die Regeln für die Benennung von Coupling-Facility-Strukturen und wie Strukturen in der CFRM-Richtliniendatei definiert werden.
2. Wenn Sie die Struktur so konfigurieren möchten, dass Nachrichtendaten in SMDS-, Zuordnungs- und Vorkonfigurationsdatensätze ausgelagert werden sollen. Weitere Informationen finden Sie im Abschnitt [Gemeinsam genutzte Nachrichtendatei erstellen](#).
3. Definieren Sie die Struktur für IBM MQ mit dem Befehl [DEFINE CFSTRUCT](#).

Entfernen einer Coupling-Facility-Struktur

Gehen Sie wie folgt vor, um eine Coupling-Facility-Struktur zu entfernen:

1. Verwenden Sie den folgenden Befehl, um eine Liste aller Warteschlangen abzurufen, die die Coupling-Facility-Struktur verwenden, die Sie löschen möchten:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Löschen Sie alle Warteschlangen, die die Struktur verwenden.
3. Löschen Sie die CF-Struktur aus IBM MQ mit dem Befehl [DELETE CFSTRUCT](#).
4. Wenn die Struktur so konfiguriert wurde, dass Nachrichtendaten in SMDS ausgelagert werden, löschen Sie die SMDS.
5. Entfernen Sie die Strukturdefinition aus Ihrer CFRM-Richtliniendatei, und führen Sie das Dienstprogramm IXCMIAPU aus. (Dies ist die Umkehrung der Anpassungstask, die die Coupling-Facility konfiguriert hat (siehe [Coupling Facility konfigurieren](#)).

Überwachung der Coupling-Facility-Liste optimieren

Verwenden Sie dieses Thema, um die Überwachung der Coupling Facility-Liste

Die Überwachung der CF-Liste (Coupling Facility) wird verwendet, um den Status von Listenstrukturen zu überwachen, die gemeinsam genutzte IBM MQ-Warteschlangen enthalten. Wenn eine Nachricht zu einer gemeinsam genutzten Warteschlange hinzugefügt wird und die Tiefe der Warteschlange von null auf ungleich null übergeht, benachrichtigt die CF alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange. Wenn die WS-Manager benachrichtigt werden, können sie eine Reihe von Aktionen ausführen, einschließlich der Benachrichtigung von Auslösermonitoren, die TRIGGER (FIRST) verwenden, oder Anwendungen, die eine Abwartestatus-Wartezeit ausführen.

Standardmäßig benachrichtigt die CF alle WS-Manager in der Gruppe mit gemeinsamer Warteschlange gleichzeitig. In bestimmten Konfigurationen kann dies zu Problemen führen, z. B.:

- Skewed-Auslastungsverteilung, bei der ein großer Prozentsatz von Nachrichten an einen bestimmten Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange gesendet wird, oft der Warteschlangenmanager, der auf der schnellsten logischen Partition (LPAR) ausgeführt wird, oder die der CF am nächsten ist, oder
- Eine große Anzahl fehlgeschlagener Abrufe, die zu einer vergeugenen CPU-Zeit führen.

z/OS V2R3 führt ein neues CFRM-Attribut (Coupling Facility Resource Manager) mit dem Namen **KEYRNOTIFYDELAY** ein, das mit Listenstrukturen verwendet werden kann, die gemeinsam genutzte Warteschlangen enthalten (d. h. Anwendungsstrukturen, nicht die Verwaltungsstruktur), und mit dem bei bestimmten Workloads die Auswirkungen ungleicher Lastverteilungen oder leerer MQGET-Aufrufe minimiert werden können.

KEYRNOTIFYDELAY kann nur auf Strukturen in einem CF gesetzt werden, die bei CFLEVEL 22 oder höher ausgeführt werden.

Sein Wert muss ein bis sieben Dezimalziffern in einem Bereich von 0 bis 1.000.000 Mikrosekunden sein. Wenn der Wert ungleich null und die Tiefe einer Warteschlange von null auf ungleich null wechselt, wählt die CF einen einzelnen WS-Manager aus der Gruppe mit gemeinsamer Warteschlange aus und benachrichtigt diesen Warteschlangenmanager vor allen anderen Warteschlangenmanagern in der Gruppe.

Der WS-Manager wird in einer Umlaufdatei ausgewählt. Wenn der ausgewählte Warteschlangenmanager die Nachricht nicht innerhalb des von **KEYRNOTIFYDELAY** beschriebenen Zeitintervalls verarbeitet, werden auch alle anderen WS-Manager in der Gruppe mit gemeinsamer Warteschlange benachrichtigt.

Mehr Informationen über **KEYRNOTIFYDELAY** ist hier erhältlich: [Verstehen Keyrange Monitoring Benachrichtigungsverzug](#).

Beachten Sie, dass es zwei ähnliche CFRM-Attribute mit dem Namen **LISTNOTIFYDELAY** und **SUBNOTIFYDELAY** gibt. Keines dieser Attribute hat messbare Auswirkungen auf die IBM MQ-Workload.

Wiederherstellung und Neustart unter z/OS

In diesem Abschnitt werden die Wiederherstellungs- und Neustartmechanismen von IBM MQ beschrieben.

erneut starten IBM MQ

Nach Beendigung eines Warteschlangenmanagers werden verschiedene Neustartprozeduren benötigt, je nachdem, wie der WS-Manager beendet wurde. In diesem Thema können Sie die verschiedenen Neustartprozeduren kennen, die Sie verwenden können.

Dieser Abschnitt enthält Informationen zum erneuten Starten des Warteschlangenmanagers unter den folgenden Umständen:

- [„Nach einem normalen Systemabschluss erneut starten“ auf Seite 426](#)
- [„Neustarten nach einer abnormalen Beendigung“ auf Seite 426](#)
- [„Neustarten, wenn Sie Ihre Seitengruppen verloren haben“ auf Seite 427](#)
- [„Sie können erneut starten, wenn Sie Ihre Protokolldatensätze verloren haben.“ auf Seite 427](#)
- [Neustart nach einem Verlust der CF-Strukturen durchführen](#)

Nach einem normalen Systemabschluss erneut starten

Wenn der Warteschlangenmanager mit dem Befehl STOP QMGR gestoppt wurde, beendet das System seine Arbeit ordnungsgemäß und nimmt einen Beendigungsprüfpunkt vor, bevor er gestoppt wird. Wenn Sie den Warteschlangenmanager erneut starten, verwendet er Informationen aus dem Systemprüfpunkt und dem Wiederherstellungsprotokoll, um den Systemstatus bei Beendigung zu ermitteln.

Geben Sie zum erneuten Starten des Warteschlangenmanagers den Befehl START QMGR wie in [„Warteschlangenmanager unter z/OS starten und stoppen“ auf Seite 351](#) beschrieben aus.

Neustarten nach einer abnormalen Beendigung

IBM MQ erkennt automatisch, ob ein Neustart auf einen normalen Systemabschluss oder eine abnormale Beendigung folgt.

Das Starten des Warteschlangenmanagers nach seiner Beendigung unterscheidet sich vom Start nach dem Absetzen des Befehls STOP QMGR nicht. Wenn der Warteschlangenmanager abnormal beendet wird, wird er beendet, ohne seine Arbeit beenden zu können oder einen Beendigungsprüfpunkt zu verwenden.

Geben Sie zum erneuten Starten des Warteschlangenmanagers den Befehl START QMGR wie in [„Warteschlangenmanager unter z/OS starten und stoppen“ auf Seite 351](#) beschrieben aus. Wenn Sie einen

WS-Manager nach einer abnormalen Beendigung erneut starten, aktualisiert er seine Kenntnisse über seinen Status bei Beendigung mit Hilfe von Informationen im Protokoll und benachrichtigt Sie über den Status verschiedener Tasks.

Normalerweise löst der Neustartprozess alle inkonsistenten Status auf. In einigen Fällen müssen Sie jedoch bestimmte Schritte unternehmen, um Inkonsistenzen zu beheben. Weitere Informationen hierzu finden Sie unter [„Manuelle Recovering-Arbeitseinheiten“](#) auf Seite 442.

Neustarten, wenn Sie Ihre Seitengruppen verloren haben

Wenn Sie Ihre Seitengruppen verloren haben, müssen Sie sie aus Ihren Sicherungskopien zurückschreiben, bevor Sie den WS-Manager erneut starten können. Weitere Informationen hierzu finden Sie unter [„Seitengruppen sichern und wiederherstellen“](#) auf Seite 409.

Der WS-Manager kann unter diesen Umständen aufgrund der für die Datenträgerwiederherstellung erforderlichen Zeit sehr lange erneut gestartet werden.

Sie können erneut starten, wenn Sie Ihre Protokolldatensätze verloren haben.

Wenn nach dem Stoppen eines Warteschlangenmanagers (mit dem Befehl STOP QMGR) beide Kopien des Protokolls verloren gehen oder beschädigt sind, können Sie den Warteschlangenmanager erneut starten, sofern Sie über eine konsistente Gruppe von Seitengruppen verfügen (die mit [Methode 1: Gesamtsicherung](#) erstellt wurden).

Gehen Sie wie folgt vor:

1. Definieren Sie neue Seitengruppen, die jeder vorhandenen Seitengruppe in Ihrem WS-Manager entsprechen. Weitere Informationen zur Definition von Seitengruppen finden Sie unter [Task 15: Seitengruppen definieren](#).
- Stellen Sie sicher, dass jede neue Seitengruppe größer ist als die entsprechende Quellenseitengruppe.
2. Verwenden Sie die FORMAT-Funktion von CSQUTIL, um die Zielseitengruppe zu formatieren. Weitere Informationen finden Sie im Abschnitt [Seitengruppen formatieren](#).
3. Verwenden Sie die Funktion RESETPAGE von CSQUTIL, um die vorhandenen Seitengruppen zu kopieren oder sie in Position zu setzen, und setzen Sie den RBA-Protokollbereich auf jeder Seite zurück. Weitere Informationen zu dieser Funktion finden Sie im Abschnitt [Seitengruppe kopieren und das Protokoll zurücksetzen](#).
4. Protokolldatensätze des Warteschlangenmanagers und BSDS mit CSQJU003 neu definieren (siehe [Dienstprogramm zum Ändern des Protokollbestands](#))
5. Starten Sie den Warteschlangenmanager unter Verwendung der neuen Seitengruppen erneut. Führen Sie dazu einen der folgenden Schritte aus:
 - Ändern Sie die gestartete Taskprozedur des WS-Managers so, dass sie auf die neuen Seitengruppen verweist. Weitere Informationen finden Sie unter [Task 6: Prozeduren für den IBM MQ-Warteschlangenmanager erstellen](#).
 - Mit den Zugriffsmethodenservices können Sie die alten Seitengruppen löschen und anschließend die neuen Seitengruppen umbenennen, so dass sie dieselben Namen wie die alten Seitengruppen erhalten.

Achtung: Stellen Sie sicher, dass Sie die erforderlichen Sicherungskopien erstellt haben, bevor Sie eine IBM MQ-Seitengruppe löschen.

Wenn der Warteschlangenmanager ein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, werden die Objektdefinitionen GROUP und SHARED normalerweise nicht durch verlorene oder beschädigte Protokolle beeinträchtigt. Wenn jedoch Nachrichten in einer gemeinsam genutzten Warteschlange an einer Arbeitseinheit beteiligt sind, die durch die verlorenen oder beschädigten Protokolle abgedeckt wurde, ist die Auswirkung auf solche nicht festgeschriebenen Nachrichten unvorhersehbar.

Anmerkung: Wenn Protokolle beschädigt sind und der Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, können gemeinsam genutzte persistente Nachrichten möglicherweise nicht wiederhergestellt werden. Geben Sie sofort den Befehl `BACKUP CFSTRUCT` auf einem anderen aktiven Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange für alle CF-Strukturen mit dem Attribut `RECOVER(YES)` aus.

Neustarten, wenn Sie Ihre CF-Strukturen verloren haben

Sie müssen nicht erneut starten, wenn Sie Ihre CF-Strukturen verloren haben, da der Warteschlangenmanager nicht beendet wird.

Wiederherstellung am alternativen Standort unter z/OS

Sie können einen einzelnen Warteschlangenmanager oder eine Gruppe mit gemeinsamer Warteschlange wiederherstellen oder eine Plattenspiegelung ausführen.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- [Einen einzelnen Warteschlangenmanager an einem alternativen Standort wiederherstellen](#)
- [Gruppe mit gemeinsamer Warteschlange wiederherstellen](#)
 - [Wiederherstellung der CF-Struktur](#)
 - [Gruppe mit gemeinsamer Warteschlange am Hauptstandort sichern](#)
 - [Gruppe mit gemeinsamer Warteschlange am alternativen Standort wiederherstellen](#)
- [Plattenspiegelung verwenden](#)

Wiederherstellen eines einzelnen Warteschlangenmanagers an einer alternativen Site

Bei einem vollständigen Ausfall eines IBM MQ-Rechenzentrums können Sie eine Wiederherstellung auf einem anderen Warteschlangenmanager oder einer anderen Gruppe mit gemeinsamer Warteschlange an einem Wiederherstellungsstandort vornehmen. (Unter „[Gruppe mit gemeinsamer Warteschlange am alternativen Standort wiederherstellen](#)“ auf Seite 432 finden Sie Informationen zur Wiederherstellungsprozedur am alternativen Standort für eine Gruppe mit gemeinsamer Warteschlange.)

Wenn Sie einen anderen Warteschlangenmanager auf einer Wiederherstellungssite wiederherstellen möchten, müssen Sie die Seitengruppen und die Protokolle regelmäßig sichern. Wie bei allen Datenwiederherstellungsoperationen sind die Ziele der Wiederherstellung nach einem Katastrophenfall so gering wie möglich, da die Daten, die Workloadverarbeitung (Aktualisierungen) und die Zeit, wie möglich, verloren gehen.

An der Wiederherstellungssite:

- Die Recoverywarteschlangenmanager **müssen** die gleichen Namen wie die verloren gegangenen Warteschlangenmanager haben.
- Das Systemparametermodul (z. B. `CSQZPARM`), das in jedem Wiederherstellungswarteschlangenmanager verwendet wird, muss die gleichen Parameter wie der entsprechende verlorene Warteschlangenmanager enthalten.

Wenn Sie dies getan haben, stellen Sie alle Warteschlangenmanager wieder her, wie in der folgenden Prozedur beschrieben. Dies kann verwendet werden, um die Wiederherstellung nach einem Katastrophenfall an der Wiederherstellungssite für einen einzelnen Warteschlangenmanager durchzuführen. Es wird davon ausgegangen, dass alle verfügbar sind:

- Kopien der Archivprotokolle und BSDSs, die durch normale Ausführung am primären Standort erstellt wurden (die aktiven Protokolle gehen zusammen mit dem Warteschlangenmanager auf der primären Site verloren).
- Kopiert die Seitengruppen vom Warteschlangenmanager am primären Standort, die im selben Alter oder älter sind als die zuletzt verfügbaren Archivierungskopien.

Sie können die doppelte Protokollierung für die aktiven Protokolldateien und die Archivprotokolle verwenden. In diesem Fall müssen Sie die BSDS-Aktualisierungen auf beide Kopien anwenden:

1. Definieren Sie neue Seitengruppen-Datensätze und laden Sie sie mit den Daten in den Kopien der Seitengruppen von der primären Site.
2. Definieren Sie neue aktive Protokolldateien.
3. Definieren Sie eine neue BSDS-Datei, und verwenden Sie REPRO für Access Method Services, um die *neuesten* archivierten BSDS in diese Datei zu kopieren.
4. Verwenden Sie das Druckprotokollzuordnungsdienstprogramm CSQJU004, um Informationen aus diesem aktuellen BSDS zu drucken. Zu dem Zeitpunkt, zu dem diese BSDS archiviert wurde, ist das zuletzt archivierte Protokoll, das Sie gerade archiviert haben, nur als aktives Protokoll abgeschnitten worden und wird nicht als archiviertes Protokoll angezeigt. Notieren Sie die STARTRBA- und ENDRBA-Datei dieses Protokolls.
5. Registrieren Sie diese neueste Archivprotokolldatei mithilfe des Dienstprogramms zum Ändern des Protokollbestands (CSQJU003) in dem BSDS, den Sie soeben wiederhergestellt haben. Verwenden Sie dazu die Werte für STARTRBA und ENDRBA, die Sie in Schritt „4“ auf Seite 429 aufgezeichnet haben.
6. Mit der Option DELETE von CSQJU003 werden alle aktiven Protokollinformationen aus dem BSDS entfernt.
7. Verwenden Sie die Option NEWLOG von CSQJU003, um aktive Protokolle zum BSDS hinzuzufügen. Geben Sie STARTRBA oder ENDRBA nicht an.
8. Verwenden Sie CSQJU003, um dem BSDS einen Wiederanlaufsteuersatz hinzuzufügen. Geben Sie CRESTART CREATE, ENDRBA=highrba an, wobei highrba für den hohen RBA des zuletzt verfügbaren Archivprotokolls steht (in Schritt „4“ auf Seite 429 gefunden), plus 1.

Das BSDS beschreibt jetzt alle aktiven Protokolle als leer, alle archivierten Protokolle, die Sie zur Verfügung haben, und keine Prüfpunkte, die über das Ende der Protokolle hinausgehen.

9. Starten Sie den WS-Manager mit dem Befehl START QMGR erneut. Während der Initialisierung wird eine Bedienerantwortnachricht wie die folgende ausgegeben:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Geben Sie Y ein, um den WS-Manager zu starten. Der Warteschlangenmanager wird gestartet und erfasst Daten bis zu ENDRBA, die in der Anweisung CRESTART angegeben sind.

Informationen zur Verwendung von CSQJU003 und CSQJU004 finden Sie im Abschnitt [IBM MQ-Dienstprogramme verwenden](#).

Das folgende Beispiel zeigt Beispieleingabeaufforderungen für CSQJU003 für die Schritte 6, 7 und 8:

```
* Step 6  
DELETE DSNAME=MQM2.LOGCOPY1.DS01  
DELETE DSNAME=MQM2.LOGCOPY1.DS02  
DELETE DSNAME=MQM2.LOGCOPY1.DS03  
DELETE DSNAME=MQM2.LOGCOPY1.DS04  
DELETE DSNAME=MQM2.LOGCOPY2.DS01  
DELETE DSNAME=MQM2.LOGCOPY2.DS02  
DELETE DSNAME=MQM2.LOGCOPY2.DS03  
DELETE DSNAME=MQM2.LOGCOPY2.DS04  
  
* Step 7  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS01,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS02,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS03,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS04,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS01,COPY2  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS02,COPY2  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS03,COPY2  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS04,COPY2  
  
* Step 8  
CRESTART CREATE,ENDRBA=063000
```

Die Punkte, die Sie beim Neustart des Kanalinitiators am Wiederherstellungsstandort beachten müssen, sind dieselben wie die, mit denen Sie bei der Verwendung von ARM zum erneuten Starten des Kanalinitiators auf einem anderen z/OS-Image konfrontiert sind. Weitere Informationen finden Sie unter „[ARM in einem IBM MQ-Netz verwenden](#)“ auf Seite 439. Ihre Wiederherstellungsstrategie sollte auch die Wiederherstellung der IBM MQ -Produktbibliotheken und der Anwendungsprogrammumgebungen umfassen, die IBM MQ verwenden (z. B. CICS).

Andere Funktionen des Änderungsprotokollbestands-Dienstprogramms (CSQJU003) können auch in Szenarios zur Wiederherstellung nach einem Katastrophenfall verwendet werden. Die Funktion HIGHRBA ermöglicht die Aktualisierung der höchsten RBA-geschriebenen und höchsten RBA-Auslastungswerte innerhalb der Bootstrap-Datei. Die Funktion CHECKPT ermöglicht das Hinzufügen neuer Prüfpunktwarteschlangendatensätze oder das Löschen vorhandener Prüfpunktwarteschlangendatensätze in der BSDS.

Achtung: Diese Funktionen können sich auf die Integrität Ihrer IBM MQ-Daten auswirken. Verwenden Sie diese Funktionen nur in Disaster-Recovery-Szenarios unter Anleitung des IBM Kundendiensts.

Fast Copy-Verfahren

Wenn Kopien aller Seitengruppen und Protokolle erstellt werden, während der Warteschlangenmanager blockiert ist, ist die Kopie eine konsistente Gruppe, die verwendet werden kann, um den WS-Manager an einer alternativen Site erneut zu starten. Sie ermöglichen in der Regel einen deutlich schnelleren Neustart des Warteschlangenmanagers, da nur wenig Datenträgerwiederherstellung ausgeführt werden kann.

Verwenden Sie den Befehl SUSPEND QMGR LOG, um den WS-Manager einzufrieren. Mit diesem Befehl werden Pufferpools in die Seitengruppen geschrieben, mit einem Prüfpunkt gestoppt und jede weitere Schreibaktivität des Protokolls gestoppt. Sobald die Protokollschreibaktivität ausgesetzt wurde, wird der Warteschlangenmanager wirksam blockiert, bis Sie einen RESUME QMGR LOG-Befehl absetzen. Während der WS-Manager blockiert ist, können die Seitengruppen und Protokolle kopiert werden.

Durch die Verwendung von Kopierwerkzeugen wie FLASHCOPY oder SNAPSHOT, um die Seitengruppen und Protokolle schnell zu kopieren, kann die Zeit, in der der Warteschlangenmanager blockiert ist, auf ein Minimum reduziert werden.

Innerhalb einer Gruppe mit gemeinsamer Warteschlange kann der Befehl SUSPEND QMGR LOG jedoch unter Umständen nicht die beste Lösung sein. Um wirksam zu sein, müssen die Kopien der Protokolle denselben Zeitpunkt für die Wiederherstellung enthalten. Das bedeutet, dass der Befehl SUSPEND QMGR LOG auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange gleichzeitig ausgegeben werden muss und somit die gesamte Gruppe mit gemeinsamer Warteschlange für einige Zeit blockiert wird.

Gruppe mit gemeinsamer Warteschlange wiederherstellen

Bei einem Ausfall an einem Hauptstandort können Sie eine Gruppe mit gemeinsamer Warteschlange mithilfe der Sicherungsdaten vom Hauptstandort an einem fernen Standort erneut starten. Für die Wiederherstellung einer Gruppe mit gemeinsamer Warteschlange müssen Sie die Wiederherstellung auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange und mit anderen Ressourcen, insbesondere mit Db2, koordinieren. In diesem Abschnitt werden diese Tasks im Detail beschrieben.

- [Wiederherstellung der CF-Struktur](#)
- [Gruppe mit gemeinsamer Warteschlange am Hauptstandort sichern](#)
- [Gruppe mit gemeinsamer Warteschlange am alternativen Standort wiederherstellen](#)

Wiederherstellung der CF-Struktur

Die Datenträgerwiederherstellung einer CF-Struktur, die zum Speichern persistenter Nachrichten in einer gemeinsam genutzten Warteschlange verwendet wird, stützt sich auf eine Sicherung der Datenträger, die durch die Anwendung protokollierter Aktualisierungen vorwärts wiederhergestellt werden können. Sichern Sie Ihre CF-Strukturen regelmäßig mit dem Befehl MQSC BACKUP CFSTRUCT. AL-

le Aktualisierungen an gemeinsam genutzten Warteschlangen (MQGET und MQPUT s) werden in das Protokoll des Warteschlangenmanagers geschrieben, in dem die Aktualisierung ausgeführt wird. Um eine Datenträgerwiederherstellung einer CF-Struktur durchzuführen, müssen Sie protokollierte Aktualisierungen für diese Sicherung aus den Protokollen aller WS-Manager anwenden, die diese CF-Struktur verwendet haben. Wenn Sie den MQSC-Befehl RECOVER CFSTRUCT verwenden, führt IBM MQ die Protokolle automatisch von den relevanten Warteschlangenmanagern zusammen und wendet die Aktualisierungen auf die aktuellste Sicherung an.

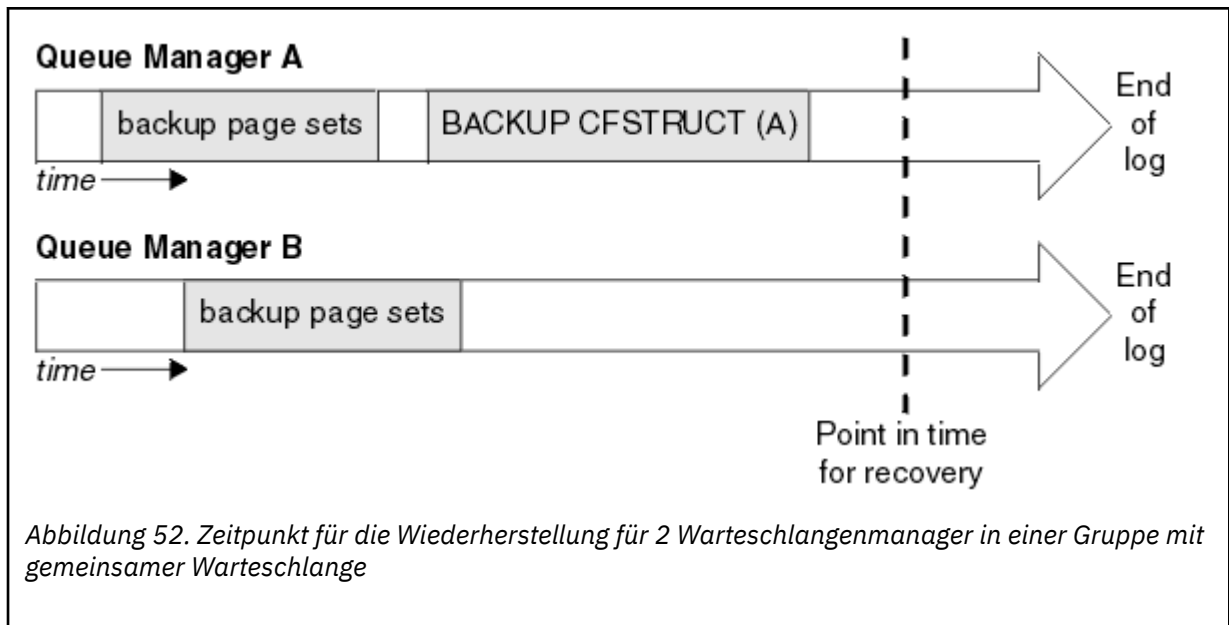
Die Sicherung der CF-Struktur wird in das Protokoll des Warteschlangenmanagers geschrieben, der den Befehl BACKUP CFSTRUCT verarbeitet hat, so dass keine zusätzlichen Datensätze erfasst und an die alternative Site transportiert werden.

Gruppe mit gemeinsamer Warteschlange auf der Hauptseite sichern

Auf der Hauptseite müssen Sie regelmäßig eine konsistente Gruppe von Sicherungen einrichten, die im Falle eines Ausfalls verwendet werden können, um die Gruppe mit gemeinsamer Warteschlange an einem alternativen Standort erneut zu erstellen. Für einen einzelnen WS-Manager kann die Wiederherstellung bis zu einem beliebigen Zeitpunkt erfolgen, in der Regel das Ende der Protokolle, die am fernen Standort verfügbar sind. Wenn persistente Nachrichten jedoch in einer gemeinsam genutzten Warteschlange gespeichert wurden, müssen alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange zusammengeführt werden, damit gemeinsam genutzte Warteschlangen wiederhergestellt werden können, da jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange möglicherweise Aktualisierungen (MQPUT oder MQGET) in der Warteschlange ausgeführt hat.

Für die Wiederherstellung einer Gruppe mit gemeinsamer Warteschlange müssen Sie einen Zeitpunkt festlegen, der sich innerhalb des Protokollbereichs der Protokolldaten aller Warteschlangenmanager befindet. Da Sie jedoch nur **vorwärts** Wiederherstellungsdatenträger aus dem Protokoll wiederherstellen können, muss dieser Zeitpunkt nach dem Absetzen des Befehls BACKUP CFSTRUCT und nach der Ausführung von Seitengruppe-Sicherungen angegeben werden. (Der Zeitpunkt für die Wiederherstellung kann in der Regel dem Ende eines Geschäftstags oder einer Woche entsprechen.)

Das folgende Diagramm zeigt die Zeitlinien für zwei Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange. Für jeden Warteschlangenmanager werden Fuzzy-Sicherungen von Seitengruppen erstellt (siehe [Methode 2: Fuzzy-Sicherung](#)). Auf WS-Manager A wird der Befehl BACKUP CFSTRUCT ausgegeben. Anschließend wird auf jedem Warteschlangenmanager ein ARCHIVE LOG-Befehl ausgegeben, um das aktive Protokoll abzuschneiden und auf Datenträger aus dem Warteschlangenmanager zu kopieren, die an den alternativen Standort transportiert werden können. Das Ende des Protokolls gibt den Zeitpunkt an, an dem der Befehl ARCHIVE LOG ausgegeben wurde, und markiert damit den Umfang der Protokolldaten, die normalerweise am alternativen Standort verfügbar sind. Der Zeitpunkt für die Wiederherstellung muss zwischen dem Ende der Seitengruppe- oder CF-Struktursicherungen und dem frühesten Ende des Protokolls liegen, das an der alternativen Site verfügbar ist.



IBM MQ zeichnet Informationen, die den CF-Struktursicherungen zugehörig sind, in einer Tabelle in Db2 auf. Abhängig von Ihren Anforderungen möchten Sie möglicherweise den Zeitpunkt für die Wiederherstellung von IBM MQ mit der für Db2 koordinieren, oder es reicht möglicherweise aus, eine Kopie der Tabelle IBM MQ CSQ.ADMIN_B_STRBACKUP zu erstellen, nachdem die Befehle BACKUP CFSTRUCT beendet wurden.

Gehen Sie wie folgt vor, um eine Wiederherstellung vorzubereiten

1. Erstellen Sie Sicherungskopien der Seitengruppen für jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange.
2. Geben Sie den Befehl BACKUP CFSTRUCT für jede CF-Struktur mit dem Attribut RECOVER (YES) aus. Sie können diese Befehle aus einem einzelnen Warteschlangenmanager oder aus verschiedenen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange ausgeben, um die Auslastung gleichmäßig zu verteilen.
3. Wenn alle Sicherungen abgeschlossen sind, geben Sie den Befehl ARCHIVE LOG aus, um das aktive Protokoll zu wechseln und Kopien der Protokolle und BSDS der einzelnen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange zu erstellen.
4. Transportieren Sie die Sicherungen der Seitengruppen, die archivierten Protokolle, die archivierten BSDS aller Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange und Ihre ausgewählten Db2-Sicherungsinformationen an einem anderen Standort.

Gruppe mit gemeinsamer Warteschlange am alternativen Standort wiederherstellen

Vor der Wiederherstellung der Gruppe mit gemeinsamer Warteschlange müssen Sie die Umgebung vorbereiten:

1. Wenn Sie bei der Installation der Gruppe mit gemeinsamer Warteschlange alte Informationen in Ihrer Coupling Facility aus der Praxis gestartet haben, müssen Sie diesen Schritt zuerst bereinigen:

Anmerkung: Wenn keine alten Informationen in der Coupling-Facility vorhanden sind, können Sie diesen Schritt überspringen.

- a. Geben Sie den folgenden z/OS-Befehl ein, um die CF-Strukturen für diese Gruppe mit gemeinsamer Warteschlange anzuzeigen:

```
D XCF,STRUCTURE,STRNAME= qsgname
```


- b. Verwenden Sie für alle Strukturen, die mit dem Namen der Gruppe mit gemeinsamer Warteschlange starten, den z/OS-Befehl SETXCF FORCE CONNECTION, um die Verbindung zu diesen Strukturen zu erzwingen:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Löschen Sie alle CF-Strukturen mit dem folgenden Befehl für jede Struktur:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Stellen Sie Db2-Systeme und Gruppen mit gemeinsamer Datennutzung wieder her.
3. Stellen Sie die Tabelle CSQ.ADMIN_B_STRBACKUP so wieder ein, dass sie Informationen zu den letzten Struktursicherungen enthält, die am Primärstandort erstellt wurden.

Anmerkung: Es ist wichtig, dass die STRBACKUP-Tabelle die neuesten Informationen zur Struktursicherung enthält. Für Informationen zur Sicherung der älteren Struktur sind möglicherweise Datensätze erforderlich, die Sie aufgrund der Informationen, die durch den aktuellen Befehl DISPLAY USAGE TYPE (DATASET) angegeben wurden, gelöscht haben. Dies würde bedeuten, dass die wiederhergestellte CF-Struktur keine genauen Informationen enthält.

4. Führen Sie den Befehl ADD QMGR des Dienstprogramms CSQ5PQSG für jeden WS-Manager in der Gruppe mit gemeinsamer Warteschlange aus. Dadurch wird der XCF-Gruppeneintrag für jeden WS-Manager wiederhergestellt.

Wenn Sie das Dienstprogramm in diesem Szenario ausführen, sind die folgenden Nachrichten normal:

```
CSQU566I Unable to get attributes for admin structure, CF not found  
or not allocated  
CSQU546E Unable to add QMGR queue_manager_name entry,  
already exists in DB2 table CSQ.ADMIN_B_QMGR  
CSQU148I CSQ5PQSG Utility completed, return code=4
```

So stellen Sie Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange wieder her:

1. Definieren Sie neue Seitengruppen-Datensätze und laden Sie sie mit den Daten in den Kopien der Seitengruppen von der primären Site.
2. Definieren Sie neue aktive Protokolldateien.
3. Definieren Sie eine neue BSDS-Datei, und verwenden Sie REPRO für Access Method Services, um die *neuesten* archivierten BSDS in diese Datei zu kopieren.
4. Verwenden Sie das Druckprotokollzuordnungsdienstprogramm CSQJU004, um Informationen aus diesem aktuellen BSDS zu drucken. Zu dem Zeitpunkt, zu dem diese BSDS archiviert wurde, ist das zuletzt archivierte Protokoll, das Sie gerade archiviert haben, nur als aktives Protokoll abgeschnitten worden und wird nicht als archiviertes Protokoll angezeigt. Notieren Sie die Werte für STARTRBA, STARTLRN, ENDRBA und ENDLRSN in diesem Protokoll.
5. Registrieren Sie diese neueste Archivprotokolldatei mithilfe des Dienstprogramms zum Ändern des Protokollbestands (CSQJU003) in der BSDS, die Sie soeben wiederhergestellt haben. Verwenden Sie dazu die in Schritt „4“ auf Seite 433 aufgezeichneten Werte.
6. Mit der Option DELETE von CSQJU003 werden alle aktiven Protokollinformationen aus dem BSDS entfernt.
7. Verwenden Sie die Option NEWLOG von CSQJU003, um aktive Protokolle zum BSDS hinzuzufügen. Geben Sie STARTRBA oder ENDRBA nicht an.
8. Berechnen Sie *recoverylrsn* für die Gruppe mit gemeinsamer Warteschlange. *recoverylrsn* ist der niedrigste der ENDLRSNs in allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange (wie in Schritt „4“ auf Seite 433 aufgezeichnet) minus 1. In der Gruppe mit gemeinsamer Warteschlange befinden sich beispielsweise zwei Warteschlangenmanager und der Wert von ENDLRSN für einen dieser Warteschlangenmanager ist B713 3C72 22C5 und für den anderen B713 3D45 2123. Dadurch ergibt sich für *recoverylrsn* der Wert B713 3C72 22C4.

9. Verwenden Sie CSQJU003, um dem BSDS einen Wiederanlaufsteuersatz hinzuzufügen. Angeben:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

Dabei ist *recoverylrsn* der Wert, den Sie in Schritt „8“ auf Seite 433 aufgezeichnet haben.

Das BSDS beschreibt jetzt alle aktiven Protokolle als leer, alle archivierten Protokolle, die Sie zur Verfügung haben, und keine Prüfpunkte, die über das Ende der Protokolle hinausgehen.

Sie müssen den CRESTART-Datensatz für jeden WS-Manager in der Gruppe mit gemeinsamer Warteschlange dem BSDS hinzufügen.

10. Starten Sie jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange mit dem Befehl START QMGR. Während der Initialisierung wird eine Bedienerantwortnachricht wie die folgende ausgegeben:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Antworten Sie Y, um den Warteschlangenmanager zu starten. Der Warteschlangenmanager wird gestartet und erfasst Daten bis zu ENDRBA, die in der Anweisung CRESTART angegeben sind.

In IBM WebSphere MQ 7.0.1 und höher kann der erste gestartete Warteschlangenmanager die Partitionen der Verwaltungsstruktur für andere Mitglieder der Gruppe mit gemeinsamer Warteschlange sowie seine eigenen Mitglieder erneut erstellen und muss nicht mehr jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange zu diesem Zeitpunkt erneut starten.

11. Wenn die Verwaltungsstrukturdaten für alle WS-Manager erneut erstellt wurden, geben Sie für jede CF-Anwendungsstruktur den Befehl RECOVER CFSTRUCT aus.

Wenn Sie den Befehl RECOVER CFSTRUCT für alle Strukturen auf einem einzelnen Warteschlangenmanager absetzen, wird der Protokollzusammenführungsprozess nur einmal ausgeführt. Daher ist es schneller, als den Befehl auf einem anderen Warteschlangenmanager für jede CF-Struktur auszugeben, wenn jeder WS-Manager den Protokollzusammenführungsschritt ausführen muss.

Wenn die Verarbeitung eines bedingten Neustarts in einer Gruppe mit gemeinsamer Warteschlange vorgenommen wird, überprüfen die Warteschlangenmanager von IBM WebSphere MQ 7.0.1 und höher, die einen Peer-Administrator erneut erstellen, ob die Bootstrap-Dateien von Peers die gleichen CRESTART-LRSN wie sie selbst enthalten. Dadurch wird die Integrität der neu erstellten Verwaltungsstruktur sichergestellt. Daher ist es wichtig, andere Peers in der QSG neu zu starten, damit sie ihre eigenen CRESTART-Informationen vor dem nächsten unbedingten Neustart eines beliebigen Mitglieds der Gruppe verarbeiten können.

Plattenspiegelung verwenden

Viele Installationen verwenden jetzt Plattenspiegelungstechnologien, z. B. IBM Metro Mirror (früher PPRC), um synchrone Kopien von Datensätzen an einem alternativen Standort zu erstellen. In solchen Situationen sind viele der Schritte nicht mehr erforderlich, da die IBM MQ-Seitengruppen und -Protokolle am alternativen Standort effektiv mit denen am Hauptstandort identisch sind. Bei der Verwendung solcher Technologien können die Schritte für den Neustart einer Gruppe mit gemeinsamer Warteschlange an einem alternativen Standort folgendermaßen zusammengefasst werden:

- Löschen Sie die IBM MQ-CF-Strukturen am alternativen Standort. (Diese enthalten oft Restinformationen aus einer vorherigen Übung zur Wiederherstellung nach einem Katastrophenfall).
- Stellen Sie Db2-Systeme und alle Tabellen in der Datenbank wieder her, die von der IBM MQ-Gruppe mit gemeinsamer Warteschlange verwendet werden.
- Starten Sie die Warteschlangenmanager erneut. Vor IBM WebSphere MQ 7.0.1 muss jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange erneut gestartet werden, da jeder Warteschlangenmanager seine eigene Partition der Verwaltungsstruktur beim Neustart wiederherstellt.

Nachdem jeder Warteschlangenmanager erneut gestartet wurde, können die nicht in der Home-LPAR nicht mehr heruntergefahren werden. In IBM WebSphere MQ 7.0.1 und höher erstellt der erste gestartete Warteschlangenmanager die Partitionen der Verwaltungsstruktur für andere Mitglieder der Gruppe mit gemeinsamer Warteschlange sowie seine eigenen Mitglieder erneut und muss nicht mehr jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange erneut starten.

- Nachdem die Verwaltungsstruktur wiederhergestellt wurde, stellen Sie die Anwendungsstrukturen wieder her.

Erneutes Initialisieren eines Warteschlangenmanagers

Wenn der WS-Manager abnormal beendet wurde, können Sie ihn möglicherweise nicht erneut starten. Dies kann daran liegen, dass Ihre Seitengruppen oder Protokolle verloren gegangen sind, abgeschnitten oder beschädigt sind. Wenn dies der Fall ist, müssen Sie möglicherweise den Warteschlangenmanager erneut initialisieren (einen Kaltstart ausführen).

Achtung

Führen Sie nur einen Kaltstart durch, wenn Sie den WS-Manager nicht anders erneut starten können.

Wenn Sie einen Kaltstart ausführen, können Sie Ihren Warteschlangenmanager und Ihre Objektdefinitionen wiederherstellen. Sie können die Nachrichtendaten **nicht** wiederherstellen. Stellen Sie sicher, dass keine der anderen in diesem Thema beschriebenen Neustartszenarios für Sie arbeiten, bevor Sie dies tun.

Nach dem Neustart sind alle IBM MQ-Objekte definiert und können verwendet werden, aber es sind keine Nachrichtendaten vorhanden.

Anmerkung: Sie können einen Warteschlangenmanager nicht erneut initialisieren, während er Teil eines Clusters ist. Sie müssen zuerst den WS-Manager aus dem Cluster entfernen (mit RESET CLUSTER-Befehlen auf den anderen WS-Managern im Cluster), dann reinitialisieren und schließlich als neuen Warteschlangenmanager wieder in den Cluster einleiten.

Dies liegt daran, dass die WS-Manager-Kennung (QMID) während der Neuinitialisierung geändert wird, so dass jedes Clusterobjekt mit der alten WS-Manager-ID aus dem Cluster entfernt werden muss.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- [Erneutes Initialisieren eines Warteschlangenmanagers, der sich nicht in einer Gruppe mit gemeinsamer Warteschlange befindet](#)
- [Erneutes Initialisieren eines Warteschlangenmanagers in einer Gruppe mit gemeinsamer Warteschlange](#)

Erneutes Initialisieren eines Warteschlangenmanagers, der sich nicht in einer Gruppe mit gemeinsamer Warteschlange befindet

Gehen Sie wie folgt vor, um einen WS-Manager erneut zu initialisieren:

1. Bereiten Sie die Objektdefinitionsanweisungen vor, die beim Neustart des Warteschlangenmanagers verwendet werden sollen. Führen Sie dazu entweder die folgenden Schritte aus:
 - Wenn Seitengruppe 0 verfügbar ist, verwenden Sie die Funktion CSQUTIL SDEFS (siehe Liste mit IBM MQ-Befehlen erstellen). Sie müssen Definitionen für alle Objektarten abrufen (Authentifizierungsinformationsobjekte, CF-Strukturen, Kanäle, Namenslisten, Prozesse, Warteschlangen und Speicherklassen).
 - Wenn die Seitengruppe 0 nicht verfügbar ist, verwenden Sie die Definitionen aus der letzten Sicherung Ihrer Objektdefinitionen.
2. Führen Sie eine erneute Definition Ihrer Warteschlangenmanagerdateien durch (tun Sie dies erst, nachdem Sie Schritt „1“ auf Seite 435 ausgeführt haben).

Weitere Informationen finden Sie im Abschnitt [Bootstrap- und Protokoll datengruppen erstellen und Seitengruppen definieren](#).

3. Starten Sie den Warteschlangenmanager unter Verwendung der neu definierten und initialisierten Protokollgruppendaten, BSDS und Seitengruppen erneut. Verwenden Sie die Eingabeanweisungen für Objektdefinitionen, die Sie in Schritt „1“ auf Seite 435 erstellt haben, als Eingabe in der CSQINP2-Initialisierungseingabedatei.

Erneutes Initialisieren eines Warteschlangenmanagers in einer Gruppe mit gemeinsamer Warteschlange

In einer Gruppe mit gemeinsamer Warteschlange ist die Neuinitialisierung eines Warteschlangenmanagers komplexer. Es kann erforderlich sein, einen oder mehrere Warteschlangenmanager aufgrund von Seitengruppen- oder Protokollproblemen erneut zu initialisieren, aber es kann auch Probleme mit Db2 oder der Coupling-Facility geben, mit Sie beheben müssen. Aus diesem Grund gibt es eine Reihe von Alternativen:

Kaltstart

Durch das erneute Initialisieren der gesamten Gruppe mit gemeinsamer Warteschlange werden alle Coupling-Facility-Strukturen erzwungen, alle Objektdefinitionen für die Gruppe mit gemeinsamer Warteschlange aus Db2 gelöscht, die Protokolle und BSDS gelöscht oder erneut definiert und Seitengruppen für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange formatiert.

Gemeinsam genutzte Definitionen beibehalten

Die Protokolle und BSDS, Formatseitengruppen für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange löschen oder neu definieren und alle Coupling-Facilities-Strukturen erzwingen. Beim Neustart werden alle Nachrichten gelöscht. Die Warteschlangenmanager erstellen COPY-Objekte erneut, die den GROUP-Objekten entsprechen, die noch in der Db2-Datenbank vorhanden sind. Alle gemeinsam genutzten Warteschlangen sind noch vorhanden und können verwendet werden.

Einzelwarteschlangenmanager wurde reinitialisiert

Die Protokolle und BSDS-Dateien sowie die Formatseitengruppen für den einzelnen Warteschlangenmanager löschen oder neu definieren (dadurch werden alle privaten Objekte und Nachrichten gelöscht). Beim Neustart erstellt der Warteschlangenmanager COPY-Objekte erneut, die den GROUP-Objekten entsprechen, die noch in der Db2-Datenbank vorhanden sind. Alle gemeinsam genutzten Warteschlangen sind weiterhin vorhanden, ebenso wie die Nachrichten zu ihnen, und können verwendet werden.

Point-in-Time-Recovery einer Gruppe mit gemeinsamer Warteschlange

Dies ist das alternative Szenario zur Wiederherstellung nach einem Katastrophenfall.

Gemeinsam genutzte Objekte werden bis zu dem Zeitpunkt wiederhergestellt, der von der Db2-Wiederherstellung erreicht wird (siehe [Ausfall eines Db2-Systems](#)). Jeder WS-Manager kann bis zu einem Zeitpunkt wiederhergestellt werden, der aus den Sicherungskopien, die an der alternativen Site verfügbar sind, erreichbar ist.

Persistente Nachrichten können in Gruppen für gemeinsame Nutzung verwendet und mithilfe des Befehls MQSC RECOVER CFSTRUCT wiederhergestellt werden. Beachten Sie, dass dieser Befehl die Zeit des Ausfalls wiedergibt. Es gibt jedoch keine Wiederherstellung nicht persistenter Nachrichten der gemeinsam genutzten Warteschlange. Sie gehen verloren, wenn Sie Sicherungskopien nicht unabhängig mit der Funktion COPY des Dienstprogramms CSQUTIL erstellt haben.

Es muss nicht versucht werden, jeden Warteschlangenmanager zu dem selben Zeitpunkt wiederherzustellen, da es keine gegenseitigen Abhängigkeiten zwischen den lokalen Objekten auf verschiedenen Warteschlangenmanagern gibt (die ja eigentlich wiederhergestellt werden). Bei der Resynchronisation des Warteschlangenmanagers mit Db2 werden beim Neustart je nach Bedarf COPY-Objekte auf einem Warteschlangenmanager (auf Warteschlangenmanagerbasis) erstellt oder gelöscht.

z/OS Automatic Restart Manager (ARM) verwenden

In diesem Thema wird erläutert, wie Sie mit ARM die Warteschlangenmanager automatisch erneut starten können.

Dieser Abschnitt enthält Informationen zu den folgenden Themen:

- [„Was ist die ARM?“ auf Seite 437](#)
- [„ARM-Richtlinien“ auf Seite 437](#)
- [„ARM in einem IBM MQ-Netz verwenden“ auf Seite 439](#)

Was ist die ARM?

Der z/OS Automatic Restart Manager (ARM) ist eine z/OS-Wiederherstellungsfunktion, die die Verfügbarkeit Ihrer Warteschlangenmanager verbessern kann. Wenn ein Job oder eine Task fehlschlägt oder das System, auf dem die Task ausgeführt wird, fehlschlägt, kann ARM den Job oder die Task ohne Bediener-eingriff erneut starten.

Wenn ein Warteschlangenmanager oder Kanalinitiator fehlgeschlagen ist, wird er von ARM in demselben z/OS-Image neu gestartet. Wenn z/OS und damit eine ganze Gruppe zusammengehöriger Subsysteme und Anwendungen fehlgeschlagen sind, kann ARM alle fehlgeschlagenen Systeme automatisch in einer vordefinierten Reihenfolge auf einem anderen z/OS-Image innerhalb des Sysplex erneut starten. Dies wird als *systemübergreifender Neustart* bezeichnet.

Starten Sie den Kanalinitiator nur unter außergewöhnlichen Umständen erneut. Wenn der Warteschlangenmanager von ARM erneut gestartet wird, starten Sie den Kanalinitiator aus der Initialisierungsdatei CSQINP2 erneut (siehe [„ARM in einem IBM MQ-Netz verwenden“ auf Seite 439](#)).

Mit ARM können Sie einen Warteschlangenmanager in einem anderen z/OS-Image innerhalb des Sysplex neu starten, falls z/OS fehlschlägt. Die Netzauswirkungen eines von IBM MQ-ARM-Neustarts auf einem anderen z/OS-Image werden im Abschnitt [„ARM in einem IBM MQ-Netz verwenden“ auf Seite 439](#) beschrieben.

So aktivieren Sie den automatischen Neustart:

- Konfigurieren Sie eine ARM-Koppeldatei.
- Definieren Sie in einer *ARM-Richtlinie* die automatischen Neustartaktionen, die z/OS ausführen soll.
- Starten Sie die ARM-Richtlinie.

Außerdem muss sich IBM MQ beim Start bei ARM registrieren (dies geschieht automatisch).

Anmerkung: Wenn Warteschlangenmanager auf verschiedenen z/OS-Images automatisch erneut gestartet werden sollen, müssen Sie jeden Warteschlangenmanager als Subsystem in jedem z/OS-Image definieren, auf dem dieser Warteschlangenmanager erneut gestartet werden könnte, wobei der aus vier Zeichen bestehende Subsystemname im gesamten Sysplex eindeutig sein muss.

ARM-Paar-Dateigruppen

Stellen Sie sicher, dass Sie die für ARM erforderlichen Datenmengen definieren und dass sie online und aktiv sind, bevor Sie einen Warteschlangenmanager starten, für den Sie die ARM-Unterstützung benötigen. Die automatische IBM MQ-ARM-Registrierung schlägt fehl, wenn die Koppeldateien beim Start des Warteschlangenmanagers nicht verfügbar sind. In dieser Situation geht IBM MQ davon aus, dass das Fehlen der Koppeldatei bedeutet, dass Sie keine ARM-Unterstützung wünschen, und die Initialisierung wird fortgesetzt.

Informationen zu ARM-Koppeldateien finden Sie in [z/OS MVS Setting up a Sysplex](#).

ARM-Richtlinien

Die Richtlinien für den automatischen Neustart (Automatic Restart Manager) sind benutzerdefinierte Regeln, die ARM-Funktionen steuern, die die Neustarts eines Warteschlangenmanagers steuern können.

ARM-Funktionen werden durch eine benutzerdefinierte *ARM-Richtlinie* gesteuert. Jedes z/OS-Image, auf dem eine Warteschlangenmanagerinstanz ausgeführt wird, die von ARM erneut gestartet werden soll, muss mit einer ARM-Koppeldatei verbunden sein, die mit einer aktiven ARM-Richtlinie definiert ist.

IBM stellt eine ARM-Standardrichtlinie bereit. Sie können neue Richtlinien definieren oder die Richtlinienstandardwerte überschreiben, indem Sie das *Dienstprogramm für Verwaltungsdaten* (IXCMIAPU) verwenden, das mit z/OS bereitgestellt wird. *z/OS MVS Setting up a Sysplex* beschreibt dieses Dienstprogramm und enthält vollständige Details zur Definition einer ARM-Richtlinie.

Abbildung 53 auf Seite 438 zeigt ein Beispiel einer ARM-Richtlinie. Mit dieser Beispielrichtlinie wird jeder Warteschlangenmanager in einem Sysplex erneut gestartet, wenn entweder der Warteschlangenmanager fehlgeschlagen ist oder ein ganzes System fehlgeschlagen ist.

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
RESTART_GROUP(DEFAULT)
ELEMENT(*)
RESTART_ATTEMPTS(0) /* Jobs not to be restarted by ARM */
RESTART_GROUP(GROUP1)
ELEMENT(SYSQMGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Abbildung 53. ARM-Beispielrichtlinie

Weitere Informationen finden Sie unter:

- [ARM-Richtlinie definieren](#)
- [ARM-Richtlinie aktivieren](#)
- [Bei ARM registrieren](#)

ARM-Richtlinie definieren

Richten Sie die ARM-Richtlinie wie folgt ein:

- Definieren Sie RESTART_GROUPS für jede Warteschlangenmanagerinstanz, die auch alle CICS- oder IMS-Subsysteme enthält, die eine Verbindung zu dieser Warteschlangenmanagerinstanz herstellen. Wenn Sie eine Namenskonvention für das Subsystem verwenden, können Sie die '?' verwenden.' und '*' -Platzhalterzeichen in Ihren Elementnamen enthalten, um RESTART_GROUPS mit minimalem Definitionsaufwand zu definieren.
- Geben Sie TERMTYPE(ELEMTerm) für Ihre Kanalinitiatoren an, um anzugeben, dass sie nur dann erneut gestartet werden, wenn der Kanalinitiator fehlgeschlagen ist und das z/OS-Image nicht fehlgeschlagen ist.
- Geben Sie TERMTYPE(ALLTERM) für Ihre Warteschlangenmanager an, um anzugeben, dass sie erneut gestartet werden, wenn entweder der Warteschlangenmanager fehlgeschlagen ist oder das z/OS-Image fehlgeschlagen ist.
- Geben Sie RESTART_METHOD (BOTH, PERSIST) für WS-Manager und Kanalinitiatoren an. Dadurch wird ARM angewiesen, die Verwendung der JCL, die während des letzten Starts gespeichert wurde (nach der Auflösung von Systemsymbolen), erneut zu starten. ARM wird angewiesen, dies unabhängig davon zu tun, ob das einzelne Element fehlgeschlagen ist oder ob das z/OS-Image fehlgeschlagen ist.
- Akzeptieren Sie die Standardwerte für alle anderen ARM-Richtlinienoptionen.

ARM-Richtlinie aktivieren

Geben Sie folgenden z/OS-Befehl aus, um Ihre Managementrichtlinie für den automatischen Neustart zu starten:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

Wenn die Richtlinie gestartet wird, verwenden alle Systeme, die mit der ARM-Koppeldatei verbunden sind, die gleiche aktive Richtlinie.

Verwenden Sie den Befehl SETXCF STOP, um die automatischen Neustarts zu inaktivieren.

Mit ARM registrieren

IBM MQ registriert sich beim Start des Warteschlangenmanagers automatisch als *ARM-Element* (abhängig von der ARM-Verfügbarkeit). Sie nimmt die Registrierung während der Beendigung der Systemabschlussphase zurück, sofern dies nicht angefordert wird.

Beim Start bestimmt der Warteschlangenmanager, ob ARM verfügbar ist. Ist dies der Fall, registriert sich IBM MQ mit dem Namen SYMQMGR *ssid*, wobei *ssid* für den vierstelligen Warteschlangenmanagernamen steht und SYMQMGR der Elementtyp ist.

Mit den Befehlen STOP QMGR MODE (QUIESCE) und STOP QMGR MODE (FORCE) wird die Registrierung des WS-Managers von ARM zurückgenommen (wenn er beim Start beim ARM registriert wurde). Dadurch wird verhindert, dass ARM diesen WS-Manager erneut startet. Mit dem Befehl STOP QMGR MODE (RESTART) wird die Registrierung des WS-Managers von ARM nicht zurückgenommen, so dass er für einen sofortigen automatischen Neustart berechtigt ist.

Jeder Kanalinitiatoradressraum bestimmt, ob ARM verfügbar ist, und meldet sich bei der Registrierung des Elementnamens SYMQCH *ssid* an, wobei *ssid* für den Namen des Warteschlangenmanagers steht und SYMQCH der Elementtyp ist.

Der Kanalinitiator wird immer von ARM zurückgenommen, wenn er normal gestoppt wird, und bleibt nur dann registriert, wenn er abnormal beendet wird. Der Kanalinitiator wird immer zurückgenommen, wenn der WS-Manager ausfällt.

ARM in einem IBM MQ-Netz verwenden

Sie können Ihren Warteschlangenmanager so konfigurieren, dass die Kanalinitiatoren und die zugeordneten Empfangsprogramme automatisch gestartet werden, wenn der Warteschlangenmanager erneut gestartet wird.

Gehen Sie wie folgt vor, um einen vollständig automatischen Neustart des Warteschlangenmanagers in demselben z/OS-Image sowohl für LU 6.2- als auch TCP/IP-Kommunikationsprotokolle sicherzustellen:

- Starten Sie die Empfangsprogramme automatisch, indem Sie den entsprechenden Befehl START LISTENER zum CSQINPX-Datensatz hinzufügen.
- Starten Sie den Kanalinitiator automatisch, indem Sie den entsprechenden Befehl START CHINIT zur Datei CSQINP2 hinzufügen.

Informationen zum erneuten Starten eines Warteschlangenmanagers mit TCP/IP oder LU6.2 finden Sie unter.

- [„Neustart auf einem anderen z/OS-Image mit TCP/IP“](#) auf Seite 439
- [„Neustart auf einem anderen z/OS-Image mit LU 6.2“](#) auf Seite 441

Weitere Informationen zu den Dateien CSQINP2 und CSQINPX finden Sie in [Task 13: Initialisierungseingabedatensätze anpassen](#).

Neustart auf einem anderen z/OS-Image mit TCP/IP

Wenn Sie TCP/IP als Kommunikationsprotokoll und virtuelle IP-Adressen verwenden, können Sie diese so konfigurieren, dass sie auf anderen z/OS-Images wiederhergestellt werden, sodass Kanäle, die eine Verbindung zu diesem Warteschlangenmanager herstellen, ohne Änderungen erneut verbunden werden

können. Andernfalls können Sie die TCP/IP-Adresse nach dem Verschieben eines Warteschlangenmanagers in ein anderes z/OS-Image nur dann erneut zuordnen, wenn Sie Cluster verwenden oder eine Verbindung zu einer Gruppe mit gemeinsamer Warteschlange über den logischen Gruppennamen für das dynamische DNS (Domain Name System) des Workload-Managers herstellen.

- Bei Verwendung von Clustering
- Bei Herstellung einer Verbindung zu einer Gruppe mit gemeinsamer Warteschlange

Bei Verwendung von Clustering

z/OS ARM reagiert auf einen Systemfehler, indem der Warteschlangenmanager auf einem anderen z/OS-Image im selben Sysplex erneut gestartet wird. Dieses System hat eine andere TCP/IP-Adresse als das ursprüngliche z/OS-Image. Im Folgenden wird erläutert, wie Sie mithilfe von IBM MQ-Clustern die TCP/IP-Adresse eines Warteschlangenmanagers erneut zuordnen können, nachdem er durch einen ARM-Neustart in ein anderes z/OS-Image verschoben wurde.

Wenn ein Client-WS-Manager den Warteschlangenmanagerfehler (als Kanalfehler) feststellt, antwortet er, indem er geeignete Nachrichten in seiner Clusterübertragungswarteschlange auf einen anderen Server-WS-Manager, der eine andere Instanz der Zielclusterwarteschlange enthält, erneut zuweist. Sie kann jedoch keine Nachrichten erneut zuordnen, die durch Affinitätsbeschränkungen an den ursprünglichen Server gebunden sind, oder Nachrichten, die sich im Zweifel befinden, weil der Warteschlangenmanager des Servers während der Verarbeitung des Stapelverarbeitungs-Stapeljobs fehlgeschlagen ist. Gehen Sie wie folgt vor, um diese Nachrichten zu verarbeiten:

1. Ordnen Sie jedem z/OS-Warteschlangenmanager einen anderen Clusterempfängerkanalnamen und einen anderen TCP/IP-Port zu. Jeder Warteschlangenmanager benötigt einen anderen Port, sodass zwei Systeme einen einzelnen TCP/IP-Stack auf einem z/OS-Image gemeinsam nutzen können. Einer davon ist der Warteschlangenmanager, der ursprünglich auf diesem z/OS-Image ausgeführt wird, und der andere ist der Warteschlangenmanager, den ARM auf diesem z/OS-Image nach einem Systemausfall erneut starten wird. Konfigurieren Sie jeden Port auf jedem z/OS-Image, sodass ARM jeden Warteschlangenmanager auf jedem z/OS-Image erneut starten kann.
2. Erstellen Sie eine andere Kanalinitiatorbefehlseingabedatei (CSQINPX) für jede Kombination aus Warteschlangenmanager und z/OS-Image, auf die beim Start des Kanalinitiators verwiesen werden soll.

Jede CSQINPX-Datei muss einen für diesen Warteschlangenmanager spezifischen START LISTENER PORT(port)-Befehl enthalten sowie einen ALTER CHANNEL-Befehl für einen Clusterempfängerkanal, der für diese Kombination aus Warteschlangenmanager und z/OS-Image spezifisch ist. Der Befehl ALTER CHANNEL muss den Verbindungsnamen auf den TCP/IP-Namen des z/OS-Images setzen, auf dem der Neustart erfolgt. Sie muss die Portnummer, die für den neu gestarteten Warteschlangenmanager spezifisch ist, als Teil des Verbindungsnamens enthalten.

Die Start-JCL jedes Warteschlangenmanagers kann einen festen Dateinamen für diese CSQINPX-Datei haben und jedes z/OS-Image muss über eine andere Version jeder CSQINPX-Datei auf einem nicht gemeinsam genutzten DASD-Datenträger verfügen.

Wenn ein ARM-Neustart ausgeführt wird, macht IBM MQ die geänderte Kanaldefinition für das Cluster-Repository bekannt, das sie wiederum an alle Clientwarteschlangenmanager veröffentlicht, die ein Interesse an dem Serverwarteschlangenmanager bekundet haben.

Der Client-WS-Manager behandelt den Ausfall des Server-WS-Managers als Kanalfehler und versucht, den fehlgeschlagenen Kanal erneut zu starten. Wenn der Client-WS-Manager den neuen Serververbindungsname erlernt, verbindet der Kanalneustart den Client-WS-Manager mit dem neu gestarteten Server-WS-Manager. Der Client-WS-Manager kann dann seine Nachrichten resynchronisieren, alle unbestätigten Nachrichten in der Übertragungswarteschlange des Clientwarteschlangenmanagers auflösen und die normale Verarbeitung fortgesetzt werden.

Beim Herstellen einer Verbindung zu einer Gruppe mit gemeinsamer Warteschlange

Wenn Sie eine Verbindung zu einer Gruppe mit gemeinsamer Warteschlange über den logischen Gruppennamen des dynamischen DNS für eine TCP/IP-Adresse herstellen, gibt der Verbindungsname in

Ihrer Kanaldefinition den logischen Gruppennamen Ihrer Gruppe mit gemeinsamer Warteschlange an und nicht den Hostnamen oder die IP-Adresse einer physischen Maschine. Beim Start dieses Kanals wird eine Verbindung zum dynamischen DNS und anschließend zu einem der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange hergestellt. Dieser Prozess wird in Kommunikation für IBM MQ for z/OS mit Gruppen mit gemeinsamer Warteschlange einrichten beschrieben.

Im unwahrscheinlichen Fall eines Imagefehlers tritt einer der folgenden Fehler auf:

- Die Warteschlangenmanager auf dem fehlerhaften Image werden aus dem dynamischen DNS, das auf Ihrem Sysplex ausgeführt wird, entfernt. Der Kanal antwortet auf den Verbindungsfehler durch Eingabe des Status RE TRYING und stellt dann eine Verbindung zu dem dynamischen DNS her, das auf dem Sysplex ausgeführt wird. Das dynamische DNS ordnet die eingehende Anforderung einem der übrigen Mitglieder der Gruppe mit gemeinsamer Warteschlange zu, das noch immer in den verbleibenden Images ausgeführt wird.
- Wenn kein anderer Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange aktiv ist und ARM den Warteschlangenmanager und den Kanalinitiator erneut in einem anderen Image startet, wird der Gruppenlistener mit dem dynamischen DNS von diesem neuen Image registriert. Dies bedeutet, dass der Name der logischen Gruppe (aus dem Feld für den Verbindungsnamen des Kanals) eine Verbindung zum dynamischen DNS herstellt und dann mit demselben Warteschlangenmanager verbunden ist, der jetzt auf einem anderen Image ausgeführt wird. Für die Kanaldefinition war keine Änderung erforderlich.

Damit diese Art der Wiederherstellung ausgeführt werden kann, müssen die folgenden Punkte beachtet werden:

- Unter z/OS wird der dynamische DNS auf einem der z/OS-Images im Sysplex ausgeführt. Wenn dieses Image fehlschlagen würde, muss der dynamische DNS so konfiguriert werden, dass im Sysplex ein sekundärer Namensserver aktiv ist, der als Alternative zum primären Namensserver agiert. Informationen zu primären und sekundären dynamischen DNS-Servern finden Sie im Handbuch *OS/390 SecureWay CS IP Configuration*.
- Der TCP/IP-Gruppenlistener wurde möglicherweise auf einer bestimmten IP-Adresse gestartet, die in diesem z/OS-Image nicht verfügbar ist. Ist dies der Fall, muss das Empfangsprogramm unter Umständen in einer anderen IP-Adresse auf dem neuen Image gestartet werden. Wenn Sie virtuelle IP-Adressen verwenden, können Sie diese so konfigurieren, dass sie auf anderen z/OS-Images wiederhergestellt werden, sodass keine Änderung des Befehls START LISTENER erforderlich ist.

Neustart auf einem anderen z/OS-Image mit LU 6.2

Wenn Sie nur LU 6.2-Kommunikationsprotokolle verwenden, führen Sie die folgende Prozedur aus, um die erneute Herstellung der Netzverbindung nach dem automatischen Neustart eines Warteschlangenmanagers auf einem anderen z/OS-Image im Sysplex zu ermöglichen:

- Definieren Sie jeden WS-Manager innerhalb des Sysplex mit einem eindeutigen Subsystemnamen.
- Definieren Sie jeden Kanalinitiator innerhalb des Sysplex mit einem eindeutigen LUNAME. Diese Angabe wird sowohl in den WS-Managerattributen als auch im Befehl START LISTENER angegeben.

Anmerkung: Der LUNAME benennt einen Eintrag in der APPC-Nebentabelle, der dies wiederum dem tatsächlichen LUNAME zuordnet.

- Richten Sie eine gemeinsam genutzte APPC-Nebentabelle ein, auf die von jedem z/OS-Image im Sysplex verwiesen wird. Dieser Eintrag muss einen Eintrag für jeden LUNAME des Kanalinitiators enthalten. Informationen zu diesem Thema finden Sie unter *z/OS MVS Planning: APPC/MVS Management*.
- Richten Sie ein Member APPCPM xx von SYS1.PARMLIB für jeden Kanalinitiator innerhalb des Sysplex ein, um eine LUADD-Datei zu enthalten, um den APPC-seitigen Tabelleneintrag für diesen Kanalinitiator zu aktivieren. Diese Member sollten von allen z/OS-Images gemeinsam genutzt werden. Das entsprechende SYS1.PARMLIB-Member wird durch einen z/OS-Befehl SET APPC=xx aktiviert, der automatisch während des ARM-Neustarts des Warteschlangenmanagers (und seines Kanalinitiators) auf einem anderen z/OS-Image ausgegeben wird, so wie im folgenden Text beschrieben.

- Verwenden Sie das WS-Manager-Attribut LU62ARM, um das Suffix xx dieses SYS1.PARMLIB-Members für jeden Kanalinitiator anzugeben. Dies bewirkt, dass der Kanalinitiator den erforderlichen z/OS-Befehl SET APPC=xx ausgibt, um dessen LUNAME zu aktivieren.

Definieren Sie Ihre ARM-Richtlinie so, dass sie den Kanalinitiator nur dann erneut startet, wenn er fehlschlägt, während sein z/OS-Image aktiv bleibt; die Benutzer-ID, die dem XCFAS-Adressraum zugeordnet ist, muss berechtigt sein, den IBM MQ-Befehl START CHINIT auszugeben. Starten Sie den Kanalinitiator nicht automatisch erneut, wenn sein z/OS-Image ebenfalls fehlschlägt. Verwenden Sie stattdessen Befehle in den Dateien CSQINP2 und CSQINPX, um den Kanalinitiator und die Listener zu starten.

Manuelle Recovering-Arbeitseinheiten

Sie können Arbeitseinheiten von CICS, IMS, RRS oder andere Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange manuell wiederherstellen. Sie können WS-Manager-Befehle verwenden, um den Status der Arbeitseinheiten anzuzeigen, die jeder Verbindung zum Warteschlangenmanager zugeordnet sind.

Dieses Thema enthält Informationen zu den folgenden Themen:

- [„Verbindungen und Threads anzeigen“ auf Seite 442](#)
- [„CICS-Arbeitseinheiten mit Wiederherstellung manuell wiederherstellen“ auf Seite 443](#)
- [„IMS-Arbeitseinheiten mit Wiederherstellung manuell wiederherstellen“ auf Seite 446](#)
- [„Wiederherstellung von RRS-Arbeitseinheiten manuell wiederherstellen“ auf Seite 448](#)
- [„Wiederherstellungseinheiten auf einem anderen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange wiederherstellen“ auf Seite 449](#)

Verbindungen und Threads anzeigen

Sie können den Befehl `DISPLAY CONN` verwenden, um Informationen zu Verbindungen zu Warteschlangenmanagern und den zugehörigen Arbeitseinheiten abzurufen. Sie können aktive Arbeitseinheiten anzeigen, um zu sehen, was gerade passiert, oder zu sehen, was beendet werden muss, damit der Warteschlangenmanager heruntergefahren werden kann, und Sie können nicht aufgelöste Arbeitseinheiten anzeigen, um die Wiederherstellung zu erleichtern.

Aktive Arbeitseinheiten

Um nur aktive Arbeitseinheiten anzuzeigen, verwenden Sie

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

Unaufgelöste Arbeitseinheiten

Eine unaufgelöste Arbeitseinheit, die auch als "unbestän. Thread" bezeichnet wird, ist eine Einheit, die sich im zweiten Arbeitsgang der zweiphasigeCommit-Operation befindet. Für einen solchen Thread werden in IBM MQ Ressourcen bereitgehalten. Um unaufgelöste Arbeitseinheiten anzuzeigen, verwenden Sie

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

Es ist ein externer Eingriff erforderlich, um den Status von nicht aufgelösten Arbeitseinheiten aufzulösen. Dies kann nur mit dem Starten des Wiederherstellungskordinators (CICS, IMS oder RRS) oder auch mit mehr verbunden sein, wie in den folgenden Abschnitten beschrieben.

In diesem Abschnitt erfahren Sie, was beim Neustart des CICS-Adapters passiert und wie Sie unaufgelöste Arbeitseinheiten mit Wiederherstellung, die sich ergeben, handhaben sollten.

Was beim Neustart des CICS-Adapters passiert

Wenn eine Verbindung unterbrochen wird, muss der Adapter während des *Reconnect-Prozesses* eine *Neustartphase* durchlaufen. Die Neustartphase resynchronisiert die Ressourcen. Die Resynchronisation zwischen CICS und IBM MQ ermöglicht es, unbestätigte Arbeitseinheiten zu identifizieren und aufzulösen.

Die Resynchronisation kann wie folgt verursacht werden:

- Eine explizite Anforderung von der verteilten Warteschlangenkomponente
- Eine implizite Anforderung, wenn eine Verbindung zu IBM MQ hergestellt wird

Wenn die Resynchronisation durch eine Verbindung zu IBM MQ verursacht wird, ist die Abfolge der Ereignisse wie folgt:

1. Der Verbindungsprozess ruft eine Liste mit IDs von unbestätigten UOW (Units of Work = Arbeitseinheiten) von IBM MQ ab.
2. Die UOW-IDs werden auf der Konsole in CSQC313I-Nachrichten angezeigt.
3. Die UOW-IDs werden an CICS übergeben.
4. CICS leitet für jede unbestätigte UOW-ID eine Resynchronisationstask (CRSY) ein.
5. Das Ergebnis der Task für jede unbestätigte UOW wird in der Konsole angezeigt.

Sie müssen die Nachrichten überprüfen, die während des Verbindungsprozesses angezeigt werden:

CSQC313I

Zeigt an, dass eine UOW im Zweifel ist.

CSQC400I

Bezeichnet die Arbeitseinheit (UOW) und wird von einer dieser Nachrichten gefolgt:

- CSQC402I oder CSQC403I zeigt, dass die Arbeitseinheit (UOW) erfolgreich aufgelöst wurde (festgeschrieben oder zurückgesetzt).
- CSQC404E, CSQC405E, CSQC406E oder CSQC407E zeigt an, dass die UOW nicht aufgelöst wurde.

CSQC409I

Zeigt an, dass alle UOWs erfolgreich aufgelöst wurden.

CSQC408I

Zeigt an, dass nicht alle UOWs erfolgreich aufgelöst wurden.

CSQC314I

Warnt, dass UOW-IDs, die mit einem Stern (*) hervorgehoben sind, nicht automatisch aufgelöst werden. Diese UOWs müssen explizit durch die verteilte Warteschlangenkomponente aufgelöst werden, wenn sie erneut gestartet wird.

[Abbildung 54 auf Seite 444](#) zeigt eine Beispielgruppe von Neustartnachrichten, die in der z/OS-Konsole angezeigt werden.

```

CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCCON CONNECT received from TERMID=PB62 TRANID=CKCN
+CSQC303I VICIC1 CSQCCON CSQCSERV loaded. Entry point is 850E8918
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F0E2178D25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F055B2AC25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6EFFD60D425 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F07AB56D22 is in doubt
+CSQC307I VICIC1 CSQCCON Successful connection to subsystem VC2
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAD18) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAA10) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BA708) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAE88) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAB80) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA878) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA570) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA268) connect
successful
+CSQC403I VICIC1 CSQCTRU Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRU UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRU Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRU UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRU Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRU UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRU Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRU UOWID=VICIC1.A6E5A6EFFD60D425
+CSQC409I VICIC1 CSQCTRU Resynchronization completed successfully

```

Abbildung 54. Beispielneustartnachrichten

Die Gesamtzahl der CSQC313I-Nachrichten muss der Gesamtzahl der Nachrichten CSQC402I plus CSQC403I entsprechen. Wenn die Gesamtsummen nicht gleich sind, gibt es UOWs, die der Verbindungsprozess nicht lösen kann. Diese UOWs, die nicht aufgelöst werden können, werden durch Probleme mit CICS (z. B. einem Kaltstart) oder mit IBM MQ oder durch die Verteilung von Warteschlangensteuerung verursacht. Wenn diese Probleme behoben wurden, können Sie eine andere Resynchronisation einleiten, indem Sie die Verbindung abschalten und anschließend wieder anschließen.

Alternativ können Sie jede ausstehende UOW mit Hilfe des Befehls RESOLVE INDOUBT und der in Nachricht CSQC400I angezeigten UOW-ID auflösen. Anschließend müssen Sie eine Verbindungstrennung und -wiederherstellung durchführen, um die *Deskriptoren der Arbeitseinheit mit Wiederherstellung* in CICS zu bereinigen. Sie müssen das richtige Ergebnis der UOW kennen, um UOWs manuell aufzulösen.

Alle Nachrichten, die nicht aufgelösten UOWs zugeordnet sind, werden von IBM MQ gesperrt, und es können keine Batch-, TSO- oder CICS -Tasks auf sie zugreifen.

Wenn CICS fehlschlägt und ein Wiederanlauf nach Systemabsturz erforderlich ist, ändern Sie *nicht* die GENERIC-APPLID des CICS-Systems. Denn sonst kann nach der Wiederherstellung der Verbindung zu IBM MQ die Datenintegrität mit IBM MQ nicht garantiert werden. Das liegt daran, dass IBM MQ die neue Instanz von CICS wie ein anderes CICS behandelt (weil die APPLID anders ist). Die Auflösung der unbestätigten Arbeitseinheit basiert dann auf dem falschen CICS-Protokoll.

Wie CICS-Arbeitseinheiten mit Wiederherstellung manuell aufgelöst werden

Wenn der Adapter abnormal beendet wird, erstellen CICS und IBM MQ unbestätigte Listen entweder dynamisch oder während des Neustarts, je nachdem, welches Subsystem die abnormale Beendigung verursacht hat.

Anmerkung: Wenn Sie Arbeitseinheiten mit dem Beispielprogramm DFH\$INDB anzeigen, werden Sie möglicherweise feststellen, dass IBM MQ-UOWs nicht immer korrekt angezeigt werden.

Wenn CICS eine Verbindung zu IBM MQ herstellt, gibt es möglicherweise eine oder mehrere Arbeitseinheiten mit Wiederherstellung, die nicht aufgelöst wurden.

Eine der folgenden Nachrichten wird an die Konsole gesendet:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

Details zur Bedeutung dieser Nachrichten finden Sie unter [Nachrichten zu CICS-Adapter und -Bridge](#).

CICS speichert Details zu den Arbeitseinheiten mit Wiederherstellung, die während des Verbindungstarts nicht aufgelöst wurden. Ein Eintrag wird gelöscht, wenn er nicht mehr in der von IBM MQ angezeigten Liste angezeigt wird.

Arbeitseinheiten mit Wiederherstellung, die nicht von CICS aufgelöst werden können, müssen mithilfe von IBM MQ-Befehlen manuell aufgelöst werden. Dieses manuelle Verfahren wird selten in einer Installation verwendet, da es nur dann erforderlich ist, wenn Betriebsfehler oder Softwareprobleme die automatische Auflösung verhindert haben. *Alle Inkonsistenzen, die bei der Auflösung unbestätigter Arbeitseinheiten gefunden werden, müssen untersucht werden.*

Gehen Sie wie folgt vor, um die Arbeitseinheiten mit

1. Rufen Sie mit folgendem Befehl eine Liste der Arbeitseinheiten mit Wiederherstellung von IBM MQ ab:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

Sie erhalten die folgende Nachricht:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC85772CBE3E0001)
EXTCONN(C3E2D8C3C7D9F0F940404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-04)
UOWLOGTI(10.17.44)
UOWSTDA(2005-02-04)
UOWSTTI(10.17.44)
UOWSTATE(UNRESOLVED)
NID(IYRCSQ1 .BC8571519B60222D)
EXTURID(BC8571519B60222D)
QMURID(0000002BDA50)
URTYPE(CICS)
USERID(MQTEST)
APPLTAG(IYRCSQ1)
ASID(0000)
APPLTYPE(CICS)
TRANSID(GP02)
TASKNO(0000096)
END CONN DETAILS
```

Bei CICS-Verbindungen besteht die NID aus der CICS-Anwendungs-ID und einer eindeutigen Nummer, die von CICS zu dem Zeitpunkt bereitgestellt wird, an dem die Synchronisationspunktprotokolleinträge geschrieben werden. Diese eindeutige Nummer wird in Datensätzen gespeichert, die zum Zeitpunkt der Synchronisationspunktverarbeitung sowohl in das CICS-Systemprotokoll als auch in das IBM MQ-Protokoll geschrieben werden. Dieser Wert wird in CICS als *Wiederherstellungstoken* bezeichnet.

2. Durchsuchen Sie das CICS-Protokoll nach Einträgen, die sich auf eine bestimmte Arbeitseinheit mit Wiederherstellung beziehen.

Suchen Sie nach einem PREPARE-Datensatz für die taskbezogene Installation, bei der das Feld für das Wiederherstellungstokenfeld (JCSRMTKN) dem Wert entspricht, der von der Netzwerk-ID abgerufen wurde. Die Netz-ID wird von IBM MQ in der Ausgabe des Befehls DISPLAY CONN bereitgestellt.

Der PREPARE-Datensatz im CICS-Protokoll für die Arbeitseinheiten mit Wiederherstellung stellt die CICS-Tasknummer bereit. Alle anderen Einträge im Protokoll für diese CICS-Task können anhand dieser Nummer gefunden werden.

Sie können beim Durchsuchen des Protokolls das CICS-Journaldruckdienstprogramm DFHJUP verwenden. Ausführliche Informationen zur Verwendung dieses Programms finden Sie im *CICS Operations and Utilities Guide*.

3. Durchsuchen Sie das IBM MQ-Protokoll nach Datensätzen, die sich auf eine bestimmte Arbeitseinheit mit Wiederherstellung beziehen. Verwenden Sie dann die URID aus diesem Datensatz, um die übrigen Protokollsätze für diese Arbeitseinheit mit Wiederherstellung abzurufen.

Beachten Sie beim Durchsuchen des IBM MQ-Protokolls, dass die IBM MQ-Startnachricht CSQJ001I die Start-RBA für diese Sitzung bereitstellt.

Das Programm zum Drucken von Protokollsätzen (CSQ1LOGP) kann zu diesem Zweck verwendet werden.

4. Wenn es sein muss, führen Sie die Auflösung unbestätigter Arbeitseinheiten in IBM MQ durch.

IBM MQ kann angewiesen werden, die Wiederherstellungsaktion für eine Arbeitseinheit mit Wiederherstellung unter Verwendung des IBM MQ-Befehls [RESOLVE INDOUBT](#) auszuführen.

Wenn Sie alle Threads wiederherstellen möchten, die einem bestimmten *connection-name* zugeordnet sind, verwenden Sie die Option NID (*).

Der Befehl erstellt eine der folgenden Nachrichten, die zeigen, ob der Thread festgeschrieben oder zurückgesetzt wird:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED
```

Wenn Sie eine Auflösung unbestätigter Arbeitseinheiten durchführen, wissen CICS und der Adapter nichts von den Befehlen an IBM MQ zum Festschreiben oder Zurückschreiben von Arbeitseinheiten mit Wiederherstellung, da nur IBM MQ-Ressourcen betroffen sind. In CICS werden jedoch Details zu den unbestätigten Threads gespeichert, die von IBM MQ nicht aufgelöst werden konnten. Diese Informationen werden gelöscht, wenn entweder die bereitgestellte Liste leer ist oder wenn sie keine Arbeitseinheit mit Wiederherstellung enthält, für die CICS Details gespeichert hat.

IMS-Arbeitseinheiten mit Wiederherstellung manuell wiederherstellen

In diesem Abschnitt erfahren Sie, was beim Neustart des IMS-Adapters passiert und wie Sie unaufgelöste Arbeitseinheiten mit Wiederherstellung, die sich ergeben, handhaben sollten.

Was beim Neustart des IMS-Adapters passiert

Immer wenn die Verbindung zu IBM MQ entweder nach einem Neustart des Warteschlangenmanagers oder nach einem IMS-Befehl /START SUBSYS wiederhergestellt wird, leitet IMS den folgenden Resynchronisationsprozess ein:

1. IMS übergibt die Liste der IDs von UOWs (Unit of Work = Arbeitseinheit), die es für unbestätigt hält, an den IBM MQ IMS-Adapter, und zwar nacheinander mit dem Auflösungsparameter für Festschreiben (Commit) oder Zurücksetzen (Backout).
2. Der IMS-Adapter übergibt die Auflösungsanforderung an IBM MQ und meldet das Ergebnis zurück an IMS.

3. Nachdem alle IMS-Auflösungsanforderungen verarbeitet wurden, ruft der IMS-Adapter von IBM MQ eine Liste aller UOWs ab, die IBM MQ nach wie vor für unbestätigt hält und die vom IMS-System initiiert wurden. Diese werden in der Nachricht CSQQ008I an das IMS-Hauptterminal gemeldet.

Anmerkung: Solange eine UOW unbestätigt ist, wird jede zugehörige IBM MQ-Nachricht von IBM MQ gesperrt und ist für keine Anwendung verfügbar.

Wie IMS-Arbeitseinheiten mit Wiederherstellung manuell aufgelöst werden

Wenn IMS eine Verbindung zu IBM MQ herstellt, hat IBM MQ möglicherweise eine oder mehrere Arbeitseinheiten mit Wiederherstellung, die nicht aufgelöst wurden.

Wenn IBM MQ unbestätigte Arbeitseinheiten mit Wiederherstellung hat, die IMS nicht aufgelöst hat, wird folgende Nachricht am IMS-Hauptterminal ausgegeben:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

Wenn diese Nachricht ausgegeben wird, wurde entweder ein Kaltstart von IMS durchgeführt oder es wurde mit einem unvollständigen Protokollband gestartet. Diese Nachricht kann auch ausgegeben werden, wenn IBM MQ oder IMS aufgrund eines Softwarefehlers oder eines anderen Subsystemfehlers abnormal beendet wird.

Nach Empfang der Nachricht CSQQ008I:

- Die Verbindung bleibt aktiv.
- IMS-Anwendungen können weiter auf IBM MQ-Ressourcen zugreifen.
- Einige IBM MQ-Ressourcen bleiben gesperrt.

Wenn der unbestätigte Thread nicht aufgelöst wird, kann es sein, dass sich IMS-Nachrichtenwarteschlangen aufbauen. Sobald die IMS-Warteschlangen voll sind, wird IMS beendet. Sie müssen sich dieser potenziellen Schwierigkeit bewusst sein und Sie müssen IMS überwachen, bis die unbestätigten Arbeitseinheiten mit Wiederherstellung vollständig aufgelöst sind.

Wiederherstellungsverfahren

Führen Sie die folgenden Schritte aus, um die IMS-Arbeitseinheiten wiederherzustellen:

1. Erzwingen Sie das Schließen des IMS-Protokolls mit /SWI OLDS und archivieren Sie anschließend das IMS-Protokoll. Verwenden Sie das Dienstprogramm DFSERA10, um die Datensätze aus dem vorherigen IMS-Protokollband zu drucken. Geben Sie X'3730'-Protokollsätze ein, und geben Sie eine Anforderung für einen phase-2-Commit-Request an, und geben Sie X'38'-Protokollsätze an, Notieren Sie die angeforderte Aktion für die letzte Transaktion in jeder abhängigen Region.
2. Führen Sie den DL/I-Stapeljob aus, um jedes beteiligte PSB zu sichern, das keinen Festschreibungspunkt erreicht hat. Der Prozess kann einige Zeit in Anspruch nehmen, da Transaktionen immer noch verarbeitet werden. Es kann auch eine Reihe von Datensätzen sperren, die sich auf den Rest der Verarbeitung und die übrigen Nachrichtenwarteschlangen auswirken könnten.
3. Erstellen Sie mit folgendem Befehl eine Liste der unbestätigten Arbeitseinheiten mit Wiederherstellung aus IBM MQ:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

Sie erhalten die folgende Nachricht:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3E2C5C3F240404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
```

```

)
UOWLOGDA(2005-02-15)
UOWLOGTI(16.39.43)
UOWSTDA(2005-02-15)
UOWSTTI(16.39.43)
UOWSTATE(UNRESOLVED)
NID(IM8F .BC45A794D3810344)
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0000)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)

```

Bei IMS besteht die NID aus dem IMS-Verbindungsnamen und einer eindeutigen Nummer, die von IMS bereitgestellt wird. Der Wert wird in IMS als *Wiederherstellungstoken* bezeichnet. Weitere Informationen finden Sie im *IMS Customization Guide*.

4. Vergleichen Sie die NIDs (IMS-ID plus OASN in Hexadezimalzeichen) in den DISPLAY THREAD-Nachrichten mit den OASNs (4 Byte dezimal), die in der Ausgabe von DFSERA10 angezeigt werden. Entscheiden Sie, ob ein Commit oder ein Backout durchgeführt werden
5. Führen Sie die Auflösung der unbestätigten Arbeitseinheit in IBM MQ wie folgt mit dem Befehl RESOLVE INDOUBT durch:

```

RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )

```

Wenn Sie alle Threads wiederherstellen möchten, die *connection-name* zugeordnet sind, verwenden Sie die Option NID (*). Der Befehl führt zu einer der folgenden Nachrichten, um anzugeben, ob der Thread festgeschrieben oder zurückgesetzt wurde:

```

CSQV414I  THREAD network-id COMMIT SCHEDULED
CSQV415I  THREAD network-id BACKOUT SCHEDULED

```

Wenn Sie eine Auflösung unbestätigter Arbeitseinheiten durchführen, wissen IMS und der Adapter nichts von den Befehlen an IBM MQ zum Festschreiben oder Zurückschreiben von unbestätigten Arbeitseinheiten mit Wiederherstellung, da nur IBM MQ-Ressourcen betroffen sind.

Wiederherstellung von RRS-Arbeitseinheiten manuell wiederherstellen

In diesem Thema wird erläutert, wie Sie feststellen können, ob unbestätigte RRS-Einheiten für die Wiederherstellung vorhanden sind und wie diese Einheiten der Wiederherstellung manuell aufgelöst werden können.

Wenn RRS eine Verbindung zu IBM MQ herstellt, hat IBM MQ möglicherweise eine oder mehrere Arbeitseinheiten mit Wiederherstellung, die nicht aufgelöst wurden. Wenn in IBM MQ unbestätigte Wiederherstellungseinheiten vorhanden sind, die RRS nicht aufgelöst hat, wird eine der folgenden Nachrichten an der z/OS -Konsole ausgegeben:

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Sowohl IBM MQ als auch RRS stellen Tools, mit denen Informationen zu unbestätigten Arbeitseinheiten mit Wiederherstellung angezeigt werden können, sowie Verfahren zu deren manueller Auflösung bereit.

Verwenden Sie in IBM MQ den Befehl DISPLAY CONN, um Informationen zu unbestätigten IBM MQ-Threads anzuzeigen. Die Ausgabe des Befehls enthält die IDs der RRS-Arbeitseinheiten mit Wiederher-

stellung für die IBM MQ-Threads, die RRS als Koordinator haben. Dies kann verwendet werden, um das Ergebnis der Arbeitseinheit mit Wiederherstellung zu ermitteln.

Verwenden Sie den Befehl RESOLVE INDOUBT, um den unbestätigten IBM MQ-Thread manuell aufzulösen. Dieser Befehl kann verwendet werden, um die Wiederherstellungseinheit festzuschreiben oder zu sichern, nachdem Sie die richtige Entscheidung getroffen haben.

z/OS **Wiederherstellungseinheiten auf einem anderen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange wiederherstellen**

In diesem Abschnitt finden Sie Informationen, wie Sie Wiederherstellungseinheiten in anderen Warteschlangenmanagern in einer Gruppe mit gemeinsamer Warteschlange ermitteln und manuell wiederherstellen können.

Wenn ein Warteschlangenmanager, der Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, fehlschlägt und nicht erneut gestartet werden kann, können andere Warteschlangenmanager in der Gruppe eine Peerwiederherstellung vornehmen und den Vorgang ab dort fortsetzen. Der Warteschlangenmanager hat jedoch möglicherweise unbestätigte Arbeitseinheiten mit Wiederherstellung, die nicht durch Peer-Wiederherstellung aufgelöst werden können, da die endgültige Disposition dieser Arbeitseinheit nur für den fehlgeschlagenen Warteschlangenmanager bekannt ist. Diese Arbeitseinheiten mit Wiederherstellung werden aufgelöst, wenn der Warteschlangenmanager schließlich erneut gestartet wird, aber bis dahin bleiben sie unbestätigt.

Dies bedeutet, dass bestimmte Ressourcen (z. B. Nachrichten) gesperrt werden können, so dass sie anderen Warteschlangenmanagern in der Gruppe nicht zur Verfügung stehen. In dieser Situation können Sie den Befehl DISPLAY THREAD verwenden, um diese Arbeitseinheiten auf dem inaktiven Warteschlangenmanager anzuzeigen. Wenn Sie diese Wiederherstellungseinheiten manuell auflösen wollen, um die Nachrichten anderen Warteschlangenmanagern in der Gruppe zur Verfügung zu stellen, können Sie den Befehl RESOLVE INDOUBT verwenden.

Wenn Sie den Befehl DISPLAY THREAD ausgeben, um die unbestätigten Arbeitseinheiten anzuzeigen, können Sie das Schlüsselwort QMNAME verwenden, um den Namen des inaktiven Warteschlangenmanagers anzugeben. Wenn Sie z. B. den folgenden Befehl ausgeben:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

Sie erhalten die folgenden Nachrichten:

```
CSQV436I +CSQ1 INDOUBT THREADS FOR QM01 -
NAME   THREAD-XREF   URID NID
USER1  0000000000000000000000000000 CSQ:0001.0
USER2  0000000000000000000000000000 CSQ:0002.0
DISPLAY THREAD REPORT COMPLETE
```

Wenn der angegebene Warteschlangenmanager aktiv ist, gibt IBM MQ keine Informationen zu unbestätigten Threads zurück, sondern gibt folgende Nachricht aus:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Verwenden Sie den Befehl IBM MQ-Befehl RESOLVE INDOUBT, um die unbestätigten Threads manuell aufzulösen. Verwenden Sie das Schlüsselwort QMNAME, um den Namen des inaktiven Warteschlangenmanagers im Befehl anzugeben.

Dieser Befehl kann verwendet werden, um die Arbeitseinheit mit Wiederherstellung festzuschreiben oder zu sichern. Der Befehl löst nur den gemeinsam genutzten Teil der Arbeitseinheit mit Wiederherstellung

auf; lokale Nachrichten sind nicht betroffen und bleiben gesperrt, bis der Warteschlangenmanager erneut gestartet wird oder er die Verbindung zu CICS, IMS oder RRS-Stapel wiederherstellt.

z/OS IBM MQ und IMS

IBM MQ stellt zwei Komponenten als Schnittstelle zu IMS bereit, den IBM MQ - IMS-Adapter und die IBM MQ - IMS-Bridge. Diese Komponenten werden in der Regel als IMS-Adapter und als IMS-Bridge bezeichnet.

z/OS IMS-Adapter betreiben

In diesem Abschnitt wird erläutert, wie der IMS -Adapter ausgeführt wird, der IBM MQ mit IMS -Systemen verbindet.

Anmerkung: Der IMS-Adapter verfügt nicht über Operationen und Steuerkonsolen.

Dieses Thema enthält die folgenden Abschnitte:

- [„IMS-Verbindungen steuern“ auf Seite 450](#)
- [„Verbindung von der IMS-Steuerregion herstellen“ auf Seite 450](#)
- [„Unbestätigte Arbeitseinheiten mit Wiederherstellung anzeigen“ auf Seite 453](#)
- [„IMS-abhängige Regionsverbindungen steuern“ auf Seite 454](#)
- [„trennen vonIMS“ auf Seite 457](#)
- [„IMS-Auslösemonitor steuern“ auf Seite 457](#)

z/OS IMS-Verbindungen steuern

Dieser Abschnitt enthält Informationen zu den IMS-Bedienerbefehlen, mit denen die Verbindung zu IBM MQ gesteuert und überwacht wird.

IMS stellt folgende Bedienerbefehle bereit, um die Verbindung zu IBM MQ zu steuern und zu überwachen:

/CHANGE SUBSYS

Löscht eine unbestätigte Arbeitseinheit mit Wiederherstellung aus IMS.

/DISPLAY OASN SUBSYS

Zeigt ausstehende Wiederherstellungselemente an.

/DISPLAY SUBSYS

Zeigt den Verbindungsstatus und die Threadaktivität an.

/START SUBSYS

Verbindet die IMS-Steuerregion mit einem Warteschlangenmanager.

/STOP SUBSYS

Trennt die Verbindung von IMS mit einem Warteschlangenmanager.

/TRACE

Controls the IMS trace.

Weitere Informationen zu diesen Befehlen finden Sie im Handbuch *IMS/ESA Operator's Reference* für die von Ihnen verwendete Version von IMS.

IMS-Befehlsantworten werden an das Terminal gesendet, von dem der Befehl ausgegeben wurde. Die Berechtigung zum Ausgeben von IMS-Befehlen basiert auf der IMS-Sicherheit.

z/OS Verbindung von der IMS-Steuerregion herstellen

In diesem Abschnitt werden die verfügbaren Verfahren zum Herstellen einer Verbindung von IMS zu IBM MQ beschrieben.

IMS stellt eine Verbindung von seiner Steuerregion zu jedem Warteschlangenmanager her, der IMS verwendet. IMS muss aktiviert sein, damit die Verbindung auf eine der folgenden Arten hergestellt werden kann:

- Automatisch während eines der folgenden Schritte
 - Eine Kaltstartinitialisierung.
 - Ein Warmstart von IMS, falls die IBM MQ-Verbindung aktiv war, als IMS beendet wurde.
- Als Antwort auf den IMS-Befehl:

```
/START SUBSYS sysid
```

Dabei steht *sysid* für den Namen des Warteschlangenmanagers.

Der Befehl kann unabhängig davon, ob der WS-Manager aktiv ist, abgesetzt werden.

Die Verbindung wird erst hergestellt, wenn der erste MQ-API-Aufruf an den WS-Manager erfolgt. Bis zu diesem Zeitpunkt zeigt der IMS-Befehl /DIS SUBSYS den Status 'NOT CONN' (nicht verbunden) an.

Die Reihenfolge, in der Sie IMS und den Warteschlangenmanager starten, ist nicht von Bedeutung.

IMS kann die Verbindung zum Warteschlangenmanager nicht automatisch erneut aktivieren, wenn der Warteschlangenmanager mit dem Befehl STOP QMGR, dem IMS-Befehl /STOP SUBSYS oder durch eine abnormale Beendigung gestoppt wird. Deshalb müssen Sie die Verbindung mit dem IMS-Befehl /START SUBSYS herstellen.

Adapter initialisieren und Verbindung zum WS-Manager herstellen

Der Adapter ist eine Gruppe von Modulen, die in die IMS-Steuerregion und abhängige Regionen geladen werden. Dies geschieht mithilfe der External Subsystem Attach Facility von IMS.

Diese Prozedur initialisiert den Adapter und stellt eine Verbindung zum WS-Manager her:

1. Lesen Sie das Subsystem-Member (SSM) aus IMS.PROCLIB. Das ausgewählte SSM ist ein IMS-EXEC-Parameter. Das Member enthält einen Eintrag für jeden Warteschlangenmanager, zu dem IMS eine Verbindung herstellen kann. Jeder Eintrag enthält Steuerinformationen zu einem IBM MQ-Adapter.
2. Laden Sie den IMS-Adapter.

Anmerkung: IMS lädt eine einzige Kopie der Adaptermodule für jede IBM MQ-Instanz, die im SSM-Member definiert ist.

3. Hängen Sie die Task des externen Subsystems für IBM MQ an.
4. Führen Sie den Adapter mit dem Parameter CTL EXEC (IMS-ID) als dem Verbindungsnamen aus.

Der Prozess ist derselbe, unabhängig davon, ob die Verbindung Teil der Initialisierung oder ein Ergebnis des IMS-Befehls /START SUBSYS ist.

Wenn der Warteschlangenmanager aktiv ist, wenn IMS versucht, die Verbindung herzustellen, werden folgende Nachrichten gesendet:

- An die z/OS-Konsole:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- An das IMS-Hauptterminal:

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```

Wenn IMS versucht, die Verbindung herzustellen und *der Warteschlangenmanager nicht aktiv ist*, werden jedes Mal, wenn eine Anwendung einen MQI-Aufruf ausgibt, folgende Nachrichten an das IMS-Hauptterminal gesendet:

```
CSQQ001I IMS/TM imsid not connected to queue manager ssnm.  
Notify message accepted  
DFS3607I MQM1 SUBSYSTEM ID EXIT FAILURE, FC = 0286, RC = 08,  
JOBNAME = IMSEMPR1
```

Wenn DFS3607I-Nachrichten beim Starten der Verbindung zu IMS oder beim Systemstart angezeigt werden, bedeutet dies, dass der Warteschlangenmanager nicht verfügbar ist. Um zu verhindern, dass eine große Anzahl von Nachrichten generiert wird, müssen Sie einen der folgenden Schritte ausführen:

1. Starten Sie den entsprechenden Warteschlangenmanager.
2. Geben Sie folgenden IMS-Befehl aus:

```
/STOP SUBSYS
```

IMS erwartet dann nicht, dass eine Verbindung zum Warteschlangenmanager hergestellt wird.

Wenn Sie weder eine Nachricht DFS3607I als auch die zugehörige Nachricht CSQQ001I erhalten, werden jedes Mal, wenn ein Job in der Region terminiert ist, und jedes Mal, wenn eine Verbindungsanforderung an den Warteschlangenmanager von einer Anwendung ausgeführt wird, ausgegeben.

Thread-Anhang

In einer MPP- oder IFP-Region stellt IMS eine Threadverbindung her, wenn das erste Anwendungsprogramm in dieser Region terminiert wird, selbst wenn dieses Anwendungsprogramm keinen IBM MQ-Aufruf ausgibt. In einer BMP-Region wird die Threadverbindung hergestellt, wenn die Anwendung ihren ersten IBM MQ-Aufruf (MQCONN oder MQCONNX) ausgibt. Dieser Thread wird für die Dauer der Region beibehalten oder bis die Verbindung gestoppt wurde.

Für die nachrichtengesteuerten und nicht nachrichtengesteuerten Regionen lautet die Querverweis-ID des Wiederherstellungsthreads, *Thread-xref*, die dem Thread zugeordnet ist:

```
PSTid + PSBname
```

Dabei gilt:

PSTid

Regionskennung der Partitionsspezifikation

PSBname

Name für Programmspezifikationsblock

Sie können Verbindungs-IDs als eindeutige IDs in IBM MQ-Befehlen verwenden. In diesem Fall fügt IBM MQ diese IDs automatisch in jede Bedienernachricht ein, die es generiert.

Unbestätigte Arbeitseinheiten mit Wiederherstellung anzeigen

Sie können unbestätigte Arbeitseinheiten mit Wiederherstellung anzeigen und versuchen, sie wiederherzustellen.

Die Arbeitsschritte, die zum Auflisten und Wiederherstellen unbestätigter Arbeitseinheiten mit Wiederherstellung in diesem Thema verwendet werden, gelten nur für relativ einfache Fälle. Wenn der Warteschlangenmanager abnormal beendet wird, während er mit IMS verbunden ist, schreibt IMS die Arbeit möglicherweise fest oder setzt sie zurück, ohne dass IBM MQ davon Kenntnis hat. Wenn der WS-Manager erneut gestartet wird, wird diese Arbeit als *unbestätigt* bezeichnet. Es muss eine Entscheidung über den Status der Arbeit getroffen werden.

Geben Sie den folgenden Befehl aus, um eine Liste der unbestätigten Arbeitseinheiten mit Wiederherstellung anzuzeigen:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ antwortet mit einer Nachricht wie der folgenden:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-11-02)
UOWLOGTI(12.27.58)
UOWSTDA(2004-11-02)
UOWSTTI(12.27.58)
UOWSTATE(UNRESOLVED)
NID(CSQ1CHIN.BC0F5F1C86FC0766)
EXTURID(00000000000001F000000007472616E5F6964547565204E6F762020...)
QMURID(0000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

Eine Erläuterung der Attribute in dieser Nachricht finden Sie in der Beschreibung des Befehls [DISPLAY CONN](#).

Recovery unbestän. Arbeitseinheiten mit Wiederherstellung

Geben Sie den folgenden Befehl aus, um unbestätigte Arbeitseinheiten mit Wiederherstellung wiederherzustellen:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

Dabei gilt:

connection-name

Die System-ID von IMS.

AKTION

Gibt an, ob diese Einheit der Wiederherstellung festgeschrieben (COMMIT) oder back out (BACKOUT) ist.

net-node.number

Die zugeordnete Netzknotennummer (net.node.number).

Wenn Sie den Befehl RESOLVE INDOUBT ausgegeben haben, wird eine der folgenden Nachrichten angezeigt:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

Residuenwiederherstellungseinträge auflösen

Zu bestimmten Zeiten erstellt IMS eine Liste mit RRE-Einträgen (Residual Recovery Entries = Residuenwiederherstellungseinträge). RREs sind Arbeitseinheit mit Wiederherstellung, die für IBM MQ möglicherweise unbestätigt sind. Sie entstehen in verschiedenen Situationen:

- Wenn der Warteschlangenmanager nicht aktiv ist, hat IMS RREs, die erst aufgelöst werden können, wenn der Warteschlangenmanager aktiv ist. Diese RREs sind kein Problem.
- Wenn der Warteschlangenmanager aktiv und mit IMS verbunden ist und IMS die Arbeit, die IBM MQ festgeschrieben hat, unterstützt, gibt der IMS -Adapter die Nachricht CSQQ010Eaus. Wenn die Daten in den beiden Systemen konsistent sein müssen, gibt es ein Problem. Informationen zum Beheben dieses Problems finden Sie im Abschnitt „[IMS-Arbeitseinheiten mit Wiederherstellung manuell wiederherstellen](#)“ auf Seite 446.
- Wenn der Warteschlangenmanager aktiv und mit IMS verbunden ist, kann es weiterhin RREs geben, obwohl Sie nicht durch Nachrichten über dieses Problem informiert wurden. Nachdem die IBM MQ-Verbindung zu IMS hergestellt wurde, können Sie den folgenden IMS-Befehl ausgeben, um zu ermitteln, ob ein Problem vorliegt:

```
/DISPLAY OASN SUBSYS sysid
```

Sie können die RRE-Einträge mit einem der folgenden IMS-Befehle löschen:

```
/CHANGE SUBSYS sysid RESET
/CHANGE SUBSYS sysid RESET OASN nnnn
```

Dabei ist *nnnn* die ursprüngliche Anwendungsfolgennummer, die als Antwort auf Ihren +CSQ1 -Befehl DISPLAY aufgelistet ist. Dies ist die Planungsnummer der Programminstanz, die den Platz in der Folge von Aufrufe des Programms seit dem letzten Kaltstart von IMS angibt. Es kann in IMS nicht zwei unbestätigte Arbeitseinheiten mit Wiederherstellung mit derselben Planungsnummer geben.

Diese Befehle setzen den Status von IMS zurück; sie führen nicht zu einer Kommunikation mit IBM MQ.

IMS-abhängige Regionsverbindungen steuern

Sie können Verbindungen zwischen IMS und IBM MQ steuern, überwachen und bei Bedarf beenden.

Die Steuerung von IMS-abhängigen Regionsverbindungen umfasst folgende Aktivitäten:

- Verbindungen von abhängigen Regionen herstellen
- Regionsfehleroptionen
- Aktivität auf Verbindungen überwachen
- Verbindungen zu abhängigen Regionen trennen

Verbindung zu abhängigen Regionen herstellen

Der in der Steuerregion verwendete IMS-Adapter wird auch in abhängige Regionen geladen. Es wird von jeder abhängigen Region eine Verbindung zu IBM MQ hergestellt. Diese Verbindung wird verwendet, um das Festschreiben der IBM MQ- und IMS-Arbeit zu koordinieren. Um die Verbindung zu initialisieren und herzustellen, geht IMS wie folgt vor:

1. Liest das Subsystemmember (SSM) aus IMS.PROCLIB.

Ein Subsystemmember kann im Parameter EXEC der abhängigen Region angegeben werden. Wenn er nicht angegeben wird, wird die Steuerregion SSM verwendet. Wenn die Region vermutlich nie eine Verbindung zu IBM MQ herstellen wird, können Sie das Laden des Adapters vermeiden, indem Sie ein Member ohne Einträge angeben.

2. Lädt den IBM MQ-Adapter.

Bei einem Stapelnachrichtenprogramm wird die Last erst dann geladen, wenn die Anwendung den ersten Nachrichtenübermittlungsbefehl ausgibt. At that time, IMS tries to make the connection.

Bei einer Region mit Nachrichtenverarbeitungsprogramm oder einer IMS-Direktaufrufregion wird der Versuch bei der Initialisierung der Region unternommen.

Optionen für Regionenfehler

Wenn der Warteschlangenmanager nicht aktiv ist oder wenn Ressourcen nicht verfügbar sind, wenn der erste Messaging-Befehl von Anwendungsprogrammen gesendet wird, hängt die Aktion von der im SSM-Eintrag angegebenen Fehleroption ab. Folgende Optionen stehen zur Auswahl:

R

Der entsprechende Rückkehrcode wird an die Anwendung gesendet.

Q

Die Anwendung wird abnormal mit dem Code für abnormale Beendigung U3051 beendet. Die Eingabenachricht wird erneut in die Warteschlange eingereicht.

A

Die Anwendung wird abnormal mit dem Code für abnormale Beendigung U3047 beendet. Die Eingabenachricht wird gelöscht.

Überwachen der Aktivität auf Verbindungen

Ein Thread wird aus einer abhängigen Region aufgebaut, wenn eine Anwendung ihre erste erfolgreiche IBM MQ-Anforderung stellt. Sie können Informationen zu Verbindungen und zu den Anwendungen, von denen sie verwendet werden, anzeigen, indem Sie folgenden Befehl aus IBM MQ ausgeben:

```
+CSQ1 DISPLAY CONN(*) ALL
```

Der Befehl erzeugt eine Nachricht wie die folgende:

```

CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-12-15)
UOWLOGTI(16.39.43)
UOWSTDA(2004-12-15)
UOWSTTI(16.39.43)
UOWSTATE(ACTIVE)
NID( )
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0049)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)

```

Für den Steuerbereich ist *thread-xref* der Sonderwert CONTROL. Bei abhängigen Regionen wird die PST-ID mit dem PSBname verkettet. *auth-id* ist entweder das Benutzerfeld von der Jobkarte oder die ID aus der Tabelle mit den gestarteten z/OS-Prozeduren.

Eine Erläuterung der angezeigten Liste finden Sie in der Beschreibung der Nachricht CSQV402I in der Dokumentation zu [IBM MQ for z/OS -Nachrichten, -Beendigungs-codes und -Ursachencodes](#).

IMS stellt einen Anzeigebefehl zum Überwachen der Verbindung zu IBM MQ bereit. Er zeigt, welches Programm in jeder abhängigen Regionsverbindung, dem LTERM-Benutzernamen und dem Verbindungsstatus der Steuerregion aktiv ist. Der Befehl lautet:

```
/DISPLAY SUBSYS name
```

Für die Verbindung zwischen IMS und IBM MQ wird einer der folgenden Status angezeigt:

```

CONNECTED
NOT CONNECTED
CONNECT IN PROGRESS
STOPPED
STOP IN PROGRESS
INVALID SUBSYSTEM NAME= name
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING

```

Der Threadstatus in jeder abhängigen Region ist eine der folgenden:

```

CONN
CONN, ACTIVE (includes LTERM of user)

```

Verbindung zu abhängigen Regionen trennen

Wenn Sie die Werte im SSM-Member von IMS.PROCLIB ändern möchten, trennen Sie eine abhängige Region. Dazu müssen Sie folgende Schritte ausführen:

1. Geben Sie folgenden IMS-Befehl aus:


```
/STOP REGION
```

2. Aktualisieren Sie das SSM-Member.
3. Geben Sie folgenden IMS-Befehl aus:

```
/START REGION
```

trennen vonIMS

Die Verbindung wird beendet, wenn entweder IMS oder der Warteschlangenmanager beendet wird. Alternativ kann der IMS-Hauptterminalbediener die Verbindung explizit unterbrechen.

Verwenden Sie zum Trennen der Verbindung zwischen IMS und IBM MQ folgenden IMS-Befehl:

```
/STOP SUBSYS sysid
```

Der Befehl sendet die folgende Nachricht an das Terminal, das sie ausgegeben hat, in der Regel der Hauptterminalbediener (MTO):

```
DFS058I STOP COMMAND IN PROGRESS
```

Der IMS-Befehl:

```
/START SUBSYS sysid
```

ist zum erneuten Herstellen der Verbindung erforderlich.

Anmerkung: Der IMS-Befehl /STOP SUBSYS wird nicht abgeschlossen, wenn ein IMS-Auslösemonitor aktiv ist.

IMS-Auslösemonitor steuern

Sie können die Transaktion CSQQTRMN verwenden, um den IMS-Auslösemonitor zu stoppen und zu starten.

Der IMS-Auslösemonitor (CSQQTRMN-Transaktion) wird im Abschnitt [IMS-Auslösemonitor einrichten](#) beschrieben.

Informationen zur Steuerung des IMS-Auslösemonitors finden Sie unter

- [CSQQTRMN starten](#)
- [CSQQTRMN stoppen](#)

CSQQTRMN wird gestartet

1. Starten Sie eine chargenorientierte BMP, die das Programm CSQQTRMN für jede Initialisierungswarteschlange ausführt, die Sie überwachen wollen.
2. Ändern Sie Ihre Batch-JCL so, dass ein DDname von CSQQUT1 hinzugefügt wird, der auf einen Datensatz verweist, der die folgenden Informationen enthält:

```
QMGRNAME=q_manager_name      Comment: queue manager name
INITQUEUENAME=init_q_name     Comment: initiation queue name
LTERM=lterm                   Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES          Comment: Send error messages to console
```

Dabei gilt:

q_manager_name	Der Name des Warteschlangenmanagers (wenn dieses Feld leer ist, wird der in CSQQDEFV angegebene Standardwert angenommen)
init_q_name	Der Name der Initialisierungswarteschlange, die überwacht werden soll.
lterm	Der IMS-LTERM-Name für die Zieladresse von Fehlernachrichten (erfolgt keine Angabe, wird der Standardwert MASTER verwendet).
CONSOLEMESSAGES=YES	Fordert, dass Nachrichten, die an das benannte IMS-LTERM gesendet werden, auch an die z/OS-Konsole gesendet werden. Wenn dieser Parameter übergangen oder falsch geschrieben ist, wird der Standardwert NICHT zum Senden von Nachrichten an die Konsole verwendet.

3. Fügen Sie einen DD-Namen von CSQQUT2 hinzu, wenn ein gedruckter Bericht über die Verarbeitung von CSQQUT1 eingegeben werden soll.

Anmerkung:

1. Der Datensatz CSQQUT1 wird mit LRECL=80 definiert. Andere DCB-Informationen werden aus dem Datensatz entnommen. Der DCB für die Dateigruppe CSQQUT2 ist RECFM=VBA und LRECL=125.
2. Sie können für jeden Datensatz nur ein Schlüsselwort verwenden. Der Schlüsselwortwert wird durch das erste Leerzeichen hinter dem Schlüsselwort begrenzt; dies bedeutet, dass Kommentare eingeschlossen werden können. Ein Stern in Spalte 1 bedeutet, dass der gesamte Eingabedatensatz ein Kommentar ist.
3. Wenn Sie eines der Schlüsselwörter QMGRNAME oder LTERM falsch schreiben, verwendet CSQQTRMN den Standardwert für dieses Schlüsselwort.
4. Stellen Sie sicher, dass das Subsystem in IMS (mit dem Befehl /START SUBSYS) gestartet wird, bevor der Auslösemonitor-BMP-Job übergeben wird. Wenn er nicht gestartet wird, wird der Auslöser für den Auslösemonitor mit dem Code für abnormale Beendigung U3042 beendet.

CSQQTRMN wird gestoppt

Nach dem Start wird CSQQTRMN ausgeführt, bis entweder die Verbindung zwischen IBM MQ und IMS aufgrund eines der folgenden Ereignisse unterbrochen wird:

- Der Warteschlangenmanager wird beendet.
- IMS wird beendet.

oder ein Befehl z/OS STOP **jobname** eingegeben wird.

IMS-Bridge steuern

Dieser Abschnitt enthält Informationen zu den IMS-Befehlen, mit denen die IMS-Bridge gesteuert werden kann.

Es gibt keine IBM MQ-Befehle zum Steuern der IBM MQ-IMS-Bridge. Es gibt jedoch folgende Möglichkeiten, die Zustellung von Nachrichten an IMS zu stoppen:

- Verwenden Sie für nicht gemeinsam genutzte Warteschlangen den Befehl ALTER QLOCAL (xxx) GET (DISABLED) für alle Brückenwarteschlangen.
- Für Clusterwarteschlangen unter Verwendung des Befehls SUSPEND QMGR CLUSTER (xxx). Dies ist nur dann wirksam, wenn ein anderer Warteschlangenmanager auch die Cluster-Bridge-Warteschlange als Host bereitstellt.
- Für Clusterwarteschlangen unter Verwendung des Befehls SUSPEND QMGR FACILITY(IMS-BRIDGE). Es werden keine weiteren Nachrichten an IMS gesendet, aber die Antworten für alle ausstehenden Transaktionen werden von IMS empfangen.

Wenn wieder Nachrichten an IMS gesendet werden sollen, geben Sie den Befehl RESUME QMGR FACILITY(IMSBRIDGE) aus.

Sie können den MQSC-Befehl DISPLAY SYSTEM auch verwenden, um anzuzeigen, ob die Brücke ausgesetzt ist.

Ausführliche Informationen zu diesen Befehlen finden Sie in den [MQSC-Befehlen](#).

Weitere Informationen finden Sie unter:

- [„IMS-Bridge starten und stoppen“ auf Seite 459](#)
- [„IMS-Verbindungen steuern“ auf Seite 459](#)
- [Bridgewarteschlangen steuern](#)
- [„IMS-Bridge resynchronisieren“ auf Seite 461](#)
- [Mit Transaktionspipenamen arbeiten](#)
- [Nachrichten aus IMS löschen](#)
- [Transaktionspipes löschen](#)
- [„IMS-Transaktionsablauf“ auf Seite 463](#)

IMS-Bridge starten und stoppen

Starten Sie die IBM MQ-Bridge, indem Sie OTMA starten. Verwenden Sie folgenden IMS-Befehl:

```
/START OTMA
```

Oder starten Sie sie automatisch, indem Sie OTMA=YES in den IMS-Systemparametern angeben. Wenn OTMA bereits gestartet ist, wird die Brücke automatisch gestartet, wenn der Start des Warteschlangenmanagers abgeschlossen ist. Beim Start von OTMA wird eine IBM MQ-Ereignisnachricht erstellt.

Verwenden Sie den folgenden IMS-Befehl:

```
/STOP OTMA
```

um die OTMA-Kommunikation zu stoppen. Bei Ausgabe dieses Befehls wird eine IBM MQ-Ereignisnachricht erstellt.

IMS-Verbindungen steuern

IMS stellt folgende Bedienerbefehle bereit, um die Verbindung zu IBM MQ zu steuern und zu überwachen:

/DEQUEUE TMEMBER *tmember* TPIPE *tpipe*

Entfernt Nachrichten aus einer TPIPE. Geben Sie PURGE an, um alle Nachrichten zu entfernen, oder PURGE1, um die erste Nachricht zu entfernen.

/DISPLAY OTMA

Zeigt zusammenfassende Informationen zum OTMA-Server und zu Clients sowie zum Clientstatus an.

/DISPLAY TMEMBER *name*

Zeigt Informationen zu einem OTMA-Client an.

/DISPLAY TRACE TMEMBER *name*

Zeigt Informationen zu dem Trace an, für den ein Trace durchgeführt wird

/SECURE OTMA

Legt die Sicherheitsoptionen fest.

/START OTMA

Aktiviert die Kommunikation über OTMA.

/START TMEMBER *tmember* TPIPE *tpipe*

Startet die benannte TPIPE.

/STOP OTMA

Stoppt die Kommunikation über OTMA.

/STOP TMEMBER *tmember* TPIPE *tpipe*

Stoppt die benannte TPIPE.

/TRACE

Controls the IMS trace.

Weitere Informationen zu diesen Befehlen finden Sie im Handbuch *IMS/ESA Operators Reference* für die von Ihnen verwendete Version von IMS.

IMS-Befehlsantworten werden an das Terminal gesendet, von dem der Befehl ausgegeben wurde. Die Berechtigung zum Ausgeben von IMS-Befehlen basiert auf der IMS-Sicherheit.

Brückenwarteschlangen steuern

Geben Sie folgenden IMS-Befehl aus, um die Kommunikation mit dem Warteschlangenmanager mit dem XCF-Membnernamen *tmember* über die Bridge zu stoppen:

```
/STOP TMEMBER tmember TPIPE ALL
```

Geben Sie folgenden IMS-Befehl aus, um die Kommunikation wiederaufzunehmen:

```
/START TMEMBER tmember TPIPE ALL
```

Die TPipes für eine Warteschlange können mit dem Befehl MQ DISPLAY QUEUE angezeigt werden.

Geben Sie folgenden IMS-Befehl aus, um die Kommunikation mit dem Warteschlangenmanager in einer einzelnen TPIPE zu stoppen:

```
/STOP TMEMBER tmember TPIPE tpipe
```

Für jede aktive Brigewarteschlange werden ein oder zwei TPipes erstellt, sodass die Ausgabe dieses Befehls die Kommunikation mit der IBM MQ-Warteschlange stoppt. Geben Sie folgenden IMS-Befehl aus, um die Kommunikation wiederaufzunehmen:

```
/START TMEMBER tmember TPIPE tpipe
```

Alternativ können Sie die Attribute der IBM MQ-Warteschlange ändern, damit sie unterdrückt werden kann.

IMS-Bridge resynchronisieren

Die IMS-Bridge wird automatisch erneut gestartet, wenn der Warteschlangenmanager, IMS oder OTMA erneut gestartet wird.

Die erste Aufgabe, die von der IMS-Bridge ausgeführt wird, ist die Resynchronisation mit IMS. Dazu gehört, dass IBM MQ und IMS Folgeummern in jeder synchronisierten TPIPE überprüfen. Ein synchronisiertes TPIPE wird verwendet, wenn persistente Nachrichten von einer IMS IBM MQ - IMS -Bridge-Warteschlange unter Verwendung des Festschreibungsmodus null (Commit durchführen-dann-senden) an gesendet werden.

Wenn die Bridge nicht mit IMS resynchronisiert werden kann, wird der Prüfcode von IMS in Nachricht CSQ2023E zurückgegeben und die Verbindung zu OTMA gestoppt. Wenn die Bridge nicht mit einer einzelnen IMS-TPIPE resynchronisiert werden kann, wird der IMS-Prüfcode in der Nachricht CSQ2025E zurückgegeben und die TPIPE gestoppt. Wenn ein TPIPE-Kaltstart gestartet wurde, werden die wiederherstellbaren Folgeummern automatisch auf 1 zurückgesetzt.

Wenn die Bridge bei der Resynchronisation mit einer TPIPE nicht übereinstimmende Folgeummern erkennt, wird die Nachricht CSQ2020E ausgegeben. Verwenden Sie den IBM MQ-Befehl RESET TPIPE, um die Resynchronisation mit der IMS-TPIPE einzuleiten. Sie müssen die XCF-Gruppe und den Membernamen sowie den Namen des TPIPE-Namens angeben. Diese Informationen werden von der Nachricht bereitgestellt.

Sie können auch Folgendes angeben:

- Eine neue wiederherstellbare Folgeummer, die in der TPIPE für Nachrichten, die von IBM MQ gesendet werden, und als Empfangsfolgeummer des Partners festgelegt werden soll. Wenn Sie die Nummer nicht angeben, wird die Empfangsfolgeummer des Partners auf die aktuelle IBM MQ-Sendefolgeummer gesetzt.
- Eine neue wiederherstellbare Folgeummer, die in der TPIPE für Nachrichten, die von IBM MQ empfangen werden, und als Sendefolgeummer des Partners festgelegt werden soll. Wenn Sie die Nummer nicht angeben, wird die Sendefolgeummer des Partners auf die aktuelle IBM MQ-Empfangsfolgeummer gesetzt.

Wenn der TPIPE eine nicht aufgelöste Arbeitseinheit zugeordnet ist, wird dies auch in der Nachricht angezeigt. Geben Sie mit dem IBM MQ-Befehl RESET TPIPE an, ob die Arbeitseinheit mit Wiederherstellung festgeschrieben oder zurückgesetzt werden soll. Wenn Sie die Arbeitseinheit mit Wiederherstellung festschreiben, wurde der Stapel der Nachrichten bereits an IMS gesendet und aus der Bridgewarteschlange gelöscht. Wenn Sie die Arbeitseinheit mit Wiederherstellung zurücksetzen, werden die Nachrichten an die Bridgewarteschlange zurückgegeben, damit sie später an IMS gesendet werden können.

Commit mode 1 (send-then-commit) TPipes werden nicht synchronisiert.

Hinweise zu Transaktionen für Commit-Modus 1

In IMS senden Transaktionen von Commit-Modus 1 (CM1) ihre Ausgabeantworten vor dem Synchronisationspunkt.

Eine CM1-Transaktion kann ihre Antwort möglicherweise nicht senden, z. B. weil:

- Die TPIPE, an die die Antwort gesendet werden soll, wird gestoppt.
- OTMA gestoppt
- Der OTMA-Client (d. a. der Warteschlangenmanager) ist nicht mehr vorhanden.
- Die Warteschlange für Antwortwarteschlange und die Warteschlange für den dead-letter sind nicht verfügbar.

Aus diesen Gründen wird die IMS-Anwendung, die die Nachricht sendet, in einem Pseudo-Abbruch mit Code U0119 beendet. IMS-Transaktion und -Programm werden in diesem Fall nicht gestoppt.

Diese Ursachen verhindern häufig, dass Nachrichten an IMS gesendet und Antworten von IMS zugestellt werden. Eine abnormale Beendigung U0119 kann auftreten, wenn:

- Die TPIPE, OTMA oder der Warteschlangenmanager wird gestoppt, während sich die Nachricht in IMS befindet.
- IMS antwortet in einer anderen TPIPE als die der eingehenden Nachricht, woraufhin diese TPIPE gestoppt wird.
- IMS antwortet an einen anderen OTMA-Client und dieser Client ist nicht verfügbar.

Jedes Mal, wenn ein U0119 Absturz auftritt, gehen sowohl die eingehende Nachricht an IMS als auch die Antwortnachrichten an IBM MQ verloren. Wenn die Ausgabe einer CM0-Transaktion aus einem dieser Gründe nicht zugestellt werden kann, wird sie in IMS in die Warteschlange der TPIPE eingereiht.

Mit TPIPE-Namen arbeiten

Viele der Befehle zur Steuerung der IBM MQ-IMS-Bridge erfordern den *TPIPE*-Namen. In diesem Thema erfahren Sie, wie Sie weitere Details zum *tpipe*-Namen finden.

Sie benötigen *TPIPE*-Namen für viele der Befehle, mit denen die IBM MQ-IMS-Bridge gesteuert wird. Sie können die *TPIPE*-Namen aus dem Befehl DISPLAY QUEUE abrufen und die folgenden Punkte beachten:

- *tpipe*-Namen werden zugeordnet, wenn eine lokale Warteschlange definiert wird.
- Eine lokale Warteschlange erhält zwei *TPIPE*-Namen, eine für die Synchronisation und eine für die Nicht-Synchronisation.
- *TPIPE*-Namen sind IMS erst dann bekannt, wenn eine besondere Kommunikation zwischen IMS und IBM MQ zu den lokalen Warteschlangen stattgefunden hat.
- Damit eine *TPIPE* für die Verwendung durch die IBM MQ-IMS-Bridge verfügbar ist, muss ihre Warteschlange einer Speicherklasse zugeordnet werden, deren Namensfelder für XCF-Gruppe und -Member korrekt ausgefüllt sind.

Nachrichten aus IMS löschen

Eine Nachricht, die über die IMS-Bridge für IBM MQ bestimmt ist, kann gelöscht werden, wenn das TMember oder die TPIPE gelöscht wird. Geben Sie folgenden IMS-Befehl aus, um eine Nachricht für den Warteschlangenmanager mit dem XCF-Membnernamen *tmember* zu löschen:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

Wenn Sie alle Nachrichten in der TPIPE löschen möchten, geben Sie folgenden IMS-Befehl aus:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

TPipes löschen

Sie können IMS-TPipes nicht selbst löschen. Sie werden zu folgenden Zeiten von IMS gelöscht:

- Synchronisierte TPipes werden bei einem Kaltstart von IMS gelöscht.
- Nicht synchronisierte TPipes werden bei einem Neustart von IMS gelöscht.

IMS-Transaktionsablauf

Einer Transaktion ist eine Ablaufzeit zugeordnet; einer IBM MQ-Nachricht kann eine Ablaufzeit zugeordnet werden. Das Ablaufintervall wird über das Feld MQMD.Expiry von der Anwendung an IBM MQ übergeben. Die Zeit ist die Dauer einer Nachricht, bevor sie abläuft, ausgedrückt als Wert in Zehntelsekunden. Ein Versuch, den MQGET-Aufruf einer Nachricht später als abgelaufen auszuführen, führt dazu, dass die Nachricht aus der Warteschlange entfernt und die Ablaufverarbeitung ausgeführt wird. Die Ablaufzeit verringert sich, während eine Nachricht zwischen Warteschlangenmanagern in einem IBM MQ-Netz übertragen wird. Wenn eine IMS-Nachricht über die IMS-Bridge an OTMA übergeben wird, wird die verbleibende Ablaufzeit der Nachricht als Transaktionsablaufzeit an OTMA übergeben.

Wenn für eine Transaktion eine Ablaufzeit angegeben ist, lässt OTMA die Eingabetransaktionen an drei verschiedenen Stellen in IMS ablaufen:

- Eingabenachricht, die von XCF empfangen wird
- Einreihungsnachrichtenwarteschlangenzeit
- GU-Zeit der Anwendung

Nach der GU-Zeit wird kein Verfall durchgeführt.

Die Transaktion EXPRTIME kann wie folgt bereitgestellt werden:

- IMS-Transaktionsdefinition
- IMS-OTMA-Nachrichtenheader
- IMS-DFSINSX0-Benutzerexit
- IMS-Befehl CREATE oder UPDATE TRAN

IMS zeigt an, dass eine Transaktion abgelaufen ist, indem eine Transaktion mit 0243 abnormal beendet und eine Nachricht ausgegeben wird. Die ausgegebene Nachricht ist entweder DFS555I in der Umgebung ohne gemeinsam genutzte Warteschlangen oder DFS2224I in der Umgebung für gemeinsam genutzte Warteschlangen.

z/OS

Advanced Message Security unter z/OS betreiben

Der Advanced Message Security-Adressraum akzeptiert Befehle mit dem z/OS-Befehl MODIFY.

Verwenden Sie den z/OS-Befehl MODIFY, um Befehle für den Advanced Message Security-Adressraum einzugeben.

Zum Beispiel:

```
F qmgr AMSM, cmd
```

Die folgenden MODIFY-Befehle werden akzeptiert:

Befehl	Option	Beschreibung
ANZEIGEN		Versionsinformationen anzeigen
REFRESH	KEYRING POLICY ALLE	Aktualisieren Sie die Schlüsselringzertifikate und/oder Sicherheitsrichtlinien.
SMFAUDIT	ERFOLG FEHLGESCHLA ALLE	Legen Sie fest, ob SMF-Protokollierung erforderlich ist, wenn AMS Nachrichten erfolgreich schützt/nicht schützt, wenn AMS keine Nachrichten schützt/ungeschützt oder beides.

Tabelle 27. MODIFY-Befehle für Advanced Message Security-Adressraum (Forts.)		
Befehl	Option	Beschreibung
SMFTYPE	0-255	Legen Sie den SMF-Satztyp fest, der generiert werden soll, wenn AMS Nachrichten schützt/nicht schützt. Geben Sie zum Inaktivieren der SMF-Prüfung den Datensatztyp 0 an.

Anmerkung: Um eine Option anzugeben, muss sie durch ein Komma voneinander getrennt werden.
Beispiel:

```
F qmgIAMS,REFRESH KEYRING
F qmgIAMS,SMFAUDIT ALL
F qmgIAMS,SMFTYPE 180
```

Befehl REFRESH.

Eine Anwendung, die einen MQOPEN-Aufruf ausgibt, nimmt die Änderungen auf. Vorhandene Anwendungen verwenden weiterhin die Optionen von, wenn diese Anwendung die Warteschlange geöffnet hat. Um die Änderungen zu übernehmen, muss eine Anwendung die Warteschlange schließen und erneut öffnen.

z/OS V 9.0.1 IBM MQ for z/OS Service-Provider für z/OS Connect

IBM MQ for z/OS Service Provider for z/OS Connect (MQ Service Provider) verarbeitet Anforderungen, die über z/OS Connect eingehen. MQ Service Provider ermöglicht REST-fähigen Anwendungen die Interaktion mit z/OS-Assets, die über IBM MQ for z/OS-Warteschlangen und Themen verfügbar sind. Dies ist ohne die Codierung möglich, die für die asynchrone Nachrichtenübermittlung erforderlich ist.

Wichtig: z/OS Connect EE Version 3.0.21.0 und höher wird mit einer erweiterten Version des MQ Service Provider geliefert, die Servicearchivdateien unterstützt. Sie sollten auf diese Version von z/OS Connect EE migrieren und den integrierten MQ Service Provider verwenden, statt des mit dem Produkt IBM MQ for z/OS gelieferten Serviceproviders.

In den z/OS Connect EE-Informationen in IBM Documentation werden Schnelleinstiegsszenarios für MQ Service Provider in z/OS Connect EE unter [Schnelleinstiegsszenarios](#) bereitgestellt. Ausführliche Referenzinformationen finden Sie unter [Service-Provider IBM MQ verwenden](#).

Dieser Abschnitt enthält die folgenden Informationen:

Zugehörige Informationen

Video: [IBM MQ Service Provider for z/OS Connect \(YouTube\)](#)

z/OS V 9.0.1 IBM MQ for z/OS Service Provider for z/OS Connect - Übersicht

Eine Übersicht über den IBM MQ for z/OS Service Provider for z/OS Connect mit einer Erläuterung der verwendeten Prinzipien und der vom Service-Provider verwendeten Verben.

Wichtig: z/OS Connect EE Version 3.0.21.0 und höher wird mit einer erweiterten Version des MQ Service Provider geliefert, die Servicearchivdateien unterstützt. Sie sollten auf diese Version von z/OS Connect EE migrieren und den integrierten MQ Service Provider verwenden, statt des mit dem Produkt IBM MQ for z/OS gelieferten Serviceproviders.

In den z/OS Connect EE-Informationen in IBM Documentation werden Schnelleinstiegsszenarios für MQ Service Provider in z/OS Connect EE unter [Schnelleinstiegsszenarios](#) bereitgestellt. Ausführliche Referenzinformationen finden Sie unter [Service-Provider IBM MQ verwenden](#).

Der MQ Service Provider unterstützt zwei Versionen von z/OS Connect:

IBM z/OS Connect Version 1 (z/OS Connect V1)

Dies ist eine Komponente von WebSphere Liberty Profile (WLP) on z/OS (ohne Aufpreis verfügbar), die z/OS-Assets als REST-Schnittstelle zugänglich macht und es fernen Anwendungen, z. B. denen auf einem mobilen Gerät, ermöglicht, ihnen mit JSON formatierte Daten zu senden.

Weitere Informationen finden Sie in [IBM z/OS Connect-Übersicht](#).

Wichtig:

1. Für MQ Service Provider for IBM z/OS Connect V1 ist der IBM WebSphere MQ 7.5-Ressourcena-dapter erforderlich. Dies hat ein Ende des Betriebsdatums vom 30. April 2018.

Ab diesem Datum wird der MQ Service Provider nicht mehr mit IBMz/OS Connect V1 unterstützt und Sie sollten stattdessen IBMz/OS Connect EE verwenden.

Aus diesem Grund ist IBM z/OS Connect V1 besser für Entwicklungs- und Konzeptnachweiszwecke geeignet als in Produktionsumgebungen.

2. Der MQ Service Provider unterstützt nur den Code von z/OS Connect V1, der in WLP 8.5.5.9 und höher geliefert wird.

IBM z/OS Connect Enterprise Edition (z/OS Connect EE)

Dies ist ein separates, gebührenpflichtiges Produkt auf der Basis von WLP, das alle Funktionen von IBM z/OS Connect V1 sowie eine Reihe von Erweiterungen bereitstellt, einschließlich Tools für die Generierung von RESTful-APIs - den API-Editor und die Integration mit IBM API Connect.

Weitere Informationen finden Sie in [IBM z/OS Connect EE](#).

Wichtig: Der MQ Service Provider unterstützt nur z/OS Connect EE V2.0.3.0 (APAR [PI66869](#)) oder höher.

Der MQ Service Provider stellt immer die gleiche Funktion bereit, unabhängig davon, welche Version von z/OS Connect verwendet wird. Alle Leistungsmerkmale von z/OS Connect EE, z. B. der API-Editor, werden mit dem MQ Service Provider unterstützt.

In der gesamten Dokumentation wird der Begriff *z/OS Connect* verwendet, um auf beide Versionen zu verweisen, es sei denn, es besteht die Notwendigkeit, auf eine bestimmte Version zu verweisen. In diesem Fall wird einer der Namen verwendet, die im vorhergehenden Text definiert sind.

Die z/OS-Assets, die von z/OS Connect zugänglich gemacht werden, müssen weder REST noch JSON verstehen, da z/OS Connect REST-Aufrufe lokalen Aufrufen zuordnet und Umsetzungen für die Konvertierung zwischen JSON und lokalen Datenstrukturen, z. B. COBOL-Copybooks, bereitstellt.

Der MQ Service Provider ist als separates WLP-Feature verfügbar, das in der relevanten z/OS Connect-Installation installiert werden kann. Beachten Sie, dass der Großteil der Arbeit zur Einrichtung von MQ Service Provider konfigurationsbasiert ist, sodass die REST-Anwendung nicht über IBM MQ informiert sein muss. Der Zugriff auf Elemente, wie z. B. MQMD-Felder, wird jedoch für erweiterte Anwendungen bereitgestellt.

Der MQ Service Provider macht IBM MQ-Warteschlangen und -Themen als „[Services](#)“ auf Seite 466 zugänglich.

Unterstützte Versionen von IBM MQ

MQ Service Provider wird mit IBM MQ for z/OS 8.0 und höher unterstützt.

Versionen vor IBM MQ 9.0.1 werden nicht mit MQ Service Providerausgeliefert. Informationen dazu, wie Sie den MQ Service Provider erhalten können, finden Sie in „[MQ Service Provider anfordern](#)“ auf Seite 469.

Die in den z/OS Connect-Abschnitten von IBM Documentation beschriebene Dokumentation bezieht sich gleichermaßen auf alle unterstützten Versionen von IBM MQ.

Weitere Überlegungen

Der MQ Service Provider verwendet einen:

- HTTP GET für einen nicht-destruktiven MQGET-Aufruf
- HTTP DELETE für einen destruktiven MQGET-Aufruf
- HTTP POST für einen MQPUT-Aufruf

Das sind dieselben Verben, die auch von der IBM MQ Bridge für HTTP verwendet werden. Ebenso verwendet der MQ Service Provider HTTP-Header als eine Möglichkeit, verschiedene Dinge anzugeben, z. B. MQMD-Werte.

Weitere Informationen zu den HTTP-Aufrufen finden Sie im Abschnitt „Vom MQ Service Provider unterstützte Verben“ auf Seite 467.

Beachten Sie, dass die einzigen Daten, die als JSON-Daten dargestellt werden, die Nutzdaten der Nachricht sind. Dies bedeutet, dass Anwendungen nicht unnötig den IBM MQ-Konzepten ausgesetzt werden, wenn sie nur die Nachrichtendaten benötigen.

Der MQ Service Provider nutzt die IBM MQ-Ressourcenadapterunterstützung in WLP und basiert daher auf JMS.

z/OS V 9.0.1 Services

Der MQ Service Provider macht IBM MQ-Warteschlangen und -Themen sowie die dahinter stehenden Anwendungen als Services zugänglich. Es gibt zwei Arten von Services: Einweg- und Zweiertypen, die in diesem Abschnitt beschrieben werden.

One-Way-Service

Ein One-Way-Service kann verwendet werden, um eine REST-konforme API für eine einzelne IBM MQ-Warteschlange oder ein einzelnes Thema bereitzustellen. RESTful-Clients können einen HTTP-POST mit einem JSON-Nutzdaten für einen Einwegservice ausgeben und der Service nimmt die Nutzdaten und sendet eine Nachricht mit den Nutzdaten als Nachrichtenhauptteil an die Zielwarteschlange oder das Topic.

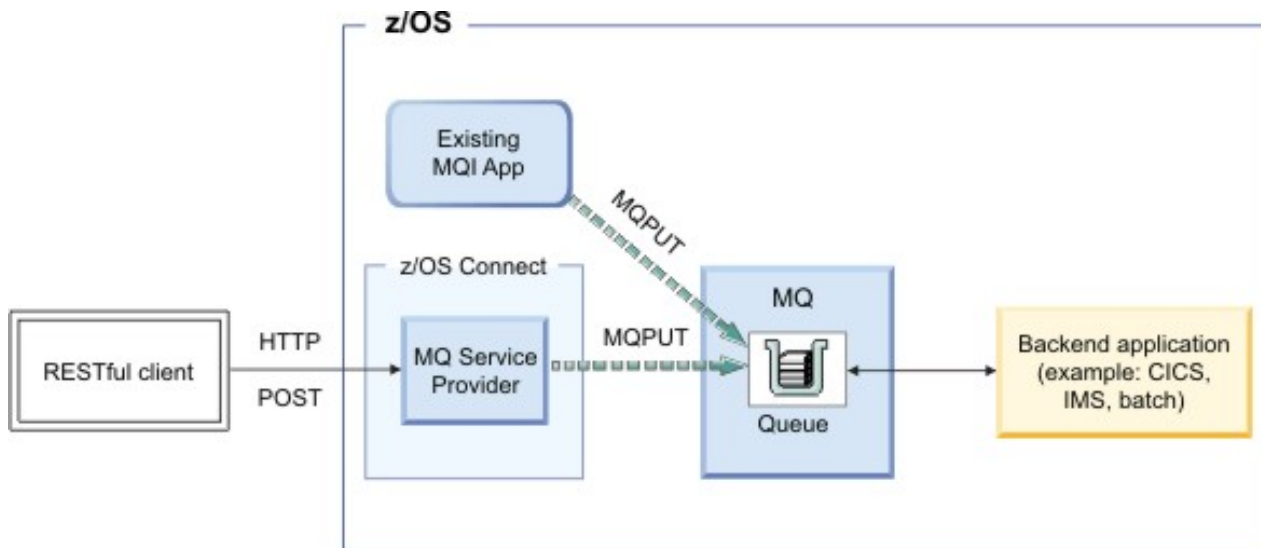


Abbildung 55. One-Way-Service

Mit One-Way-Services ist es auch möglich, HTTP-DELETE- und HTTP-GET-Anforderungen für IBM MQ-Warteschlangen auszugeben.

Eine HTTP-DELETE-Operation führt zu einem zerstörerischen Abrufen einer verfügbaren Nachricht aus der Warteschlange. Ein HTTP-GET-Ergebnis führt zu einer Suche nach der ersten verfügbaren Nachricht aus der Warteschlange.

Anmerkung: Zwei HTTP-GET-Aufrufe geben die gleiche Nachricht zurück, es sei denn, es ist eine andere Aktion aufgetreten, z. B. eine HTTP DELETE-oder Nachrichtenablaufzeit, um die Nachricht aus der Warteschlange zu entfernen.

Der Hauptteil der Nachricht wird an den RESTful-Client in Form von JSON zurückgegeben. Wenn sich der Hauptteil der Nachricht noch nicht im JSON-Format befindet, z. B. ein COBOL-Copybook, kann z/OS Connect so konfiguriert werden, dass es mithilfe einer Datentransformation eine Konvertierung in JSON durchführt.

Two-Way-Service

Ein Zwei-Wege-Service ermöglicht es einem RESTful-Client, Anforderungen für Antwortnachrichten an ein Paar von Warteschlangen auszuführen.

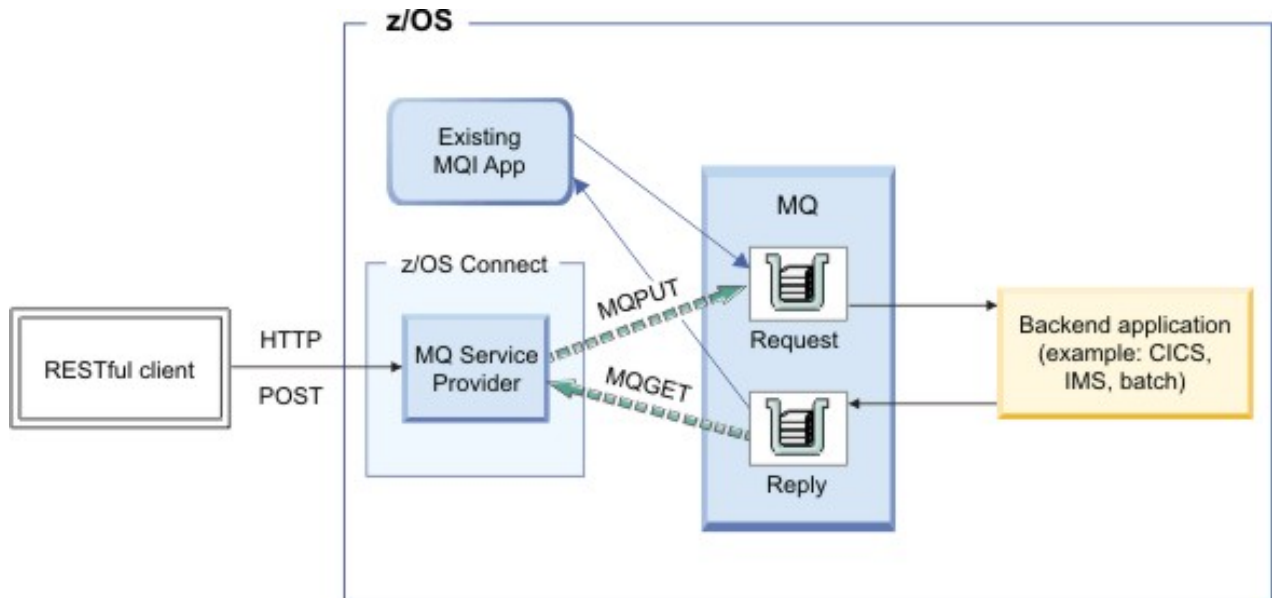


Abbildung 56. Two-Way-Service

Der Client gibt eine HTTP-POST-Anforderung aus, die eine JSON-Nutzdaten angibt. Der Service verwendet die Nutzdaten, konvertiert ihn optional in ein anderes Format, z. B. ein COBOL-Copybook, und sendet es als Nachricht an eine Anforderungswarteschlange.

Eine Back-End-Anwendung konsumiert die Nachricht, verarbeitet sie und generiert eine Antwort, die in eine Antwortwarteschlange gestellt wird. Der Service lokalisiert diese Nachricht, nimmt ihre Nutzdaten an, konvertiert sie optional in JSON und gibt sie als Antwortenbody des HTTP-POST zurück.

z/OS V 9.0.1 Vom MQ Service Provider unterstützte Verben

Der MQ Service Provider unterstützt die Verben HTTP GET, HTTP DELETE und HTTP POST, und zwar in derselben Bedeutung, wie sie in der IBM MQ Bridge for HTTP verwendet werden. Die Verwendung dieser Verben ermöglicht die grundlegende Interaktion mit IBM MQ for z/OS und durch Zusammensetzung mehrerer dieser Verben können komplexere Funktionen bereitgestellt werden.

Bei einem HTTP GET oder HTTP DELETE ignoriert der MQ Service Provider einen in der Anforderung bereitgestellten Hauptteil.

Wenn HTTP GET oder HTTP DELETE erfolgreich ist:

- Die Nachrichtennutzdaten werden im HTTP-Antwortteil zurückgegeben, und der HTTP-Statuscode ist 200.
- Eine Nachricht ohne Nutzdaten wird ohne HTTP-Hauptteil zurückgegeben. Wenn keine Nachricht verfügbar ist, wird kein HTTP-Hauptteil zurückgegeben, und der HTTP-Statuscode lautet 204 (Kein Inhalt).



Achtung: z/OS Connect erwartet, dass JSON-Nutzdaten immer im Objektformat vorliegen. Das heißt, die minimal gültigen Nutzdaten sind `{}`.

Bei einer HTTP-POST-Operation werden nur die Nachrichtennutzdaten, die sich im HTTP-Hauptteil befinden, und die verschiedenen im Element `mqzOSConnectService-Element` beschriebenen Elemente erwartet.

Wenn der POST erfolgreich ist, gilt Folgendes:

One-Way-Service

Gibt einen HTTP-Code 204 (Kein Inhalt) zusammen mit einer leeren Nutzlast zurück.

Two-Way-Service

Gibt entweder den Inhalt der Antwortnachricht im JSON-Format zurück oder, wenn keine Antwortnachricht verfügbar war, einen leeren Antwortinhalt mit einem HTTP-Antwortcode 204 (Kein Inhalt).

HTTP-Header

Wenn Sie einen HTTP-POST absetzen, muss der HTTP-Hauptteil ein gültiges JSON-Objekt sein, wenn es entweder ein Einweg-oder ein Zwei-Wege-Service ist. Sie müssen Folgendes angeben:

- Ein Content-Type=application/json -HTTP-Header mit dem HTTP_POST
- Eine Codierung von UTF-8

Es können abhängig vom verwendeten HTTP-Verb auch mehrere optionale HTTP-Header angegeben werden, um das Verhalten des MQ Service Provider anzupassen. Weitere Informationen finden Sie unter HTTP-Header, die mit dem MQ Service Provider verwendet werden können.

HTTP-Codes

Wenn der MQ Service Provider einen Fehler erkennt, setzt er einen HTTP-Statuscode im Bereich von 400-599. Andernfalls werden die zuvor dokumentierten Statuscodes immer zurückgegeben.

Fehler auf Clientseite

Wenn in der aufrufenden Anwendung nicht gültige Daten übergeben werden, wird ein HTTP-Statuscode im Bereich 400-499 an den Aufrufenden zurückgegeben. Der genaue Statuscode hängt von dem Fehler ab.

Wenn die nicht gültigen Daten von einer Ausnahmebedingung festgestellt wurden, wird ein Stack-Trace im JSON-Format wie im Beispiel der Serverseite bereitgestellt.

Fehler auf Serverseite

Wenn der MQ Service Provider einen unerwarteten Fehler erkennt, wird der Stack-Trace in das JSON-Format umgewandelt und mit dem HTTP-Statuscode 500 (Interner Serverfehler) an das aufrufende Programm zurückgegeben. Außerdem werden geeignete Diagnosedaten in die Protokolle von z/OS Connect geschrieben.

Eine vereinfachte Beispiel-Fehlerantwortnutzlast ist:

```
{
  "errorText": "CSQZ0006E: An unexpected JMSEException occurred while processing a request
for the 'mq7' service. ",
  "exceptionData": {
    "exceptions": [
      {
        "stackTrace": [
          "com.ibm.msg.client.jms.DetailedInvalidDestinationException: JMSWMQ2008:
Failed to open MQ queue 'ThisQueueDoesNotExist'.",
          <further content removed for brevity>
        ]
      },
      {
        "stackTrace": [
          "com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with
compcode '2' ('MQCC_FAILED') reason '2085' ('MQRC_UNKNOWN_OBJECT_NAME').",
          <further content removed for brevity>
        ]
      }
    ]
  }
}
```

z/OS V 9.0.1 MQ Service Provider anfordern

MQ Service Provider wird für die Verwendung mit IBM MQ for z/OS 8.0 und höher unterstützt. Es gibt zwei Möglichkeiten, den MQ Service Provider für die Verwendung mit einer unterstützten Version von IBM MQ abzurufen.

1. „MQ Service Provider über die Unix Systems Services Components-Funktion von IBM MQ 9.0.1 (oder höher) abrufen“ auf Seite 469
2. „MQ Service Provider von Fix Central abrufen“ auf Seite 469

MQ Service Provider über die Unix Systems Services Components-Funktion von IBM MQ 9.0.1 (oder höher) abrufen

Installieren Sie die Komponente gemäß den Anweisungen im Programmverzeichnis. MQ Service Provider befindet sich im Verzeichnis *PATHPREFIX/zosconnect* Ihres Dateisystems, wobei *PATHPREFIX* der bei der Installation ausgewählte Wert ist. Für IBM MQ 9.0.1 ist *PATHPREFIX* standardmäßig */usr/lpp/mqm/V9R0M1*.

MQ Service Provider von Fix Central abrufen

Rufen Sie [Fix Central](#) auf, um den MQ Service Provider zu suchen, indem Sie nach *IBM-MQ-zOSConnect-ServiceProvider* suchen.

Dies sollte eine TAR-Datei finden, z. B. *9.0.1-IBM-MQ-zOSConnect-ServiceProvider.tar*, die auf Ihre Workstation heruntergeladen werden kann.



Achtung: Dieselbe TAR-Datei wird auch mit IBM MQ 8.0 verwendet. Es gibt keine separate TAR-Datei für IBM MQ 8.0.

Übertragen Sie die TAR-Datei nach dem Download in ein Verzeichnis *PATHPREFIX* in einer entsprechenden z/OS -LPAR. Dieses Verzeichnis muss irgendwo sein, wo z/OS Connect darauf zugreifen kann.

Extrahieren Sie die Dateien, indem Sie den folgenden Befehl ausgeben:

```
tar -oxvf 9.0.1-IBM-MQ-zOSConnect-ServiceProvider.tar
```

Ändern Sie die Berechtigungen der Verzeichnisstruktur entsprechend für Ihr Unternehmen.

Verzeichnisstruktur

Bei beiden Methoden für den Abruf des MQ Service Providers wird die gleiche Verzeichnisstruktur erstellt. Das Stammverzeichnis dieser Struktur wird in der gesamten Dokumentation als *MQSP_ROOT* bezeichnet, wobei *MQSP_ROOT* für *PATHPREFIX/zosconnect* steht.

Unter *MQSP_ROOT* befinden sich die folgenden Dateien und Verzeichnisse:

mqzosconnect.properties	Properties file that can be copied into z/OS Connect
v1.0/	Directory containing MQ Service Provider for z/OS
Connect V1	
lib/	
com.ibm.mq.zosconnect_1.0.0.jar	Feature jar file for MQ Service Provider for z/OS
Connect V1	
features/	
zosConnectMQ-1.0.mf	Feature manifest for MQ Service Provider for z/OS
Connect V1	
v2.0/	Directory containing MQ Service Provider for z/OS
Connect EE	
lib/	
com.ibm.mq.zosconnect_2.0.0.jar	Feature jar file for MQ Service Provider for z/OS

Hinweise zu Transaktionen

Da HTTP kein transaktionsorientiertes Protokoll ist, ist eine transaktionsorientierte Koordination von Messaging-Operationen, die vom MQ Service Provider ausgeführt werden, nicht möglich.

Dies hat die folgenden Auswirkungen:

- Wenn ein HTTP-POST an einen Einwegservice gesendet wird und die Verbindung fehlschlägt, bevor eine HTTP-Antwort vom Client empfangen wird, kann der Client nicht sofort erkennen, ob die Nachricht an die konfigurierte Warteschlange oder das konfigurierte Thema gesendet wurde oder nicht.
- Wenn ein HTTP-Befehl DELETE für einen Einwegservice ausgeführt wird und die Verbindung fehlschlägt, bevor eine HTTP-Antwort vom Client empfangen wird, wurde möglicherweise eine Nachricht aus der Warteschlange entfernt und ist verloren gegangen, da keine Möglichkeit zum Rollback der zerstörerischen Rückruodatei vorhanden ist.
- Wenn eine HTTP-Anforderung POST an einen zweiseitigen Service gesendet wird und die Verbindung fehlschlägt, bevor eine HTTP-Antwort vom Client empfangen wird, kann der Client nicht erkennen, wo der Fehler aufgetreten ist. Möglicherweise wurde die Anforderungsnachricht an die Anforderungswarteschlange gesendet, oder die Antwortnachricht wurde möglicherweise aus der Antwortwarteschlange gesendet und ist verloren gegangen.
- Es gibt keine Möglichkeit, das Ergebnis mehrerer HTTP-Verben zu koordinieren, weder auf eine Einbahnstraße noch auf einen zweiseitigen Dienst.

IBM z/OS Connect EE - Setup-Prozedur für MQ Service Provider

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect EE konfiguriert werden kann.

Informationen zu diesem Vorgang

Führen Sie die folgenden Aufgaben in der angegebenen Reihenfolge aus, um den MQ Service Provider und alle zugehörigen Voraussetzungen zu installieren.

IBM z/OS Connect EE installieren

Die Voraussetzungen und die Vorgehensweise für die Installation von IBMz/OS Connect EE.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie Zugriff auf [z/OS Connect EE V2 Embedded WLP](#) haben.

Informationen zu diesem Vorgang

Mit dieser Prozedur wird ein z/OS Connect EE-Server eingerichtet, der für die Verwendung mit dem MQ Service Provider geeignet ist. Wenn Sie bereits über einen Server verfügen, der auf dieselbe Weise eingerichtet wurde, kann stattdessen der Server verwendet werden.

Vorgehensweise

1. Erstellen Sie einen Server mit den hier beschriebenen Anweisungen: [z/OS Connect EE Server erstellen](#)
2. Aktivieren Sie den TXRRS-autorisierten Service, indem Sie die im Abschnitt [Liberty-Angel-Prozess und z/OS-autorisierte Services konfigurieren](#) beschriebene Prozedur ausführen.

3. Überprüfen Sie, ob der TXRRS-autorisierte Service korrekt konfiguriert wurde:

a) Der Server wird gestartet.

Führen Sie die im Abschnitt [z/OS Connect EE starten und stoppen](#) beschriebene Prozedur aus.

b) Zeigen Sie die Serverprotokolle an, die sich unter befinden:

```
/var/zosconnect/servers/server_name/logs/messages.log
```

Hierbei handelt es sich um ASCII-Dateien, die eine Ausgabe ähnlich der folgenden enthalten sollten:

```
A CWWKE0001I: The server test has been launched.
I CWWKB0103I: Authorized service group LOCALCOM is available.
I CWWKB0103I: Authorized service group PRODMGR is available.
I CWWKB0103I: Authorized service group SAFCRE is available.
I CWWKB0103I: Authorized service group TXRRS is available.
I CWWKB0103I: Authorized service group WOLA is available.
I CWWKB0103I: Authorized service group ZOSDUMP is available.
I CWWKB0103I: Authorized service group ZOSWLM is available.
I CWWKB0103I: Authorized service group CLIENT.WOLA is available.
I CWWKB0108I: IBM CORP product z/OS Connect version 02.00 successfully registered with z/OS
```

Überprüfen Sie die Ausgabe, um festzustellen, ob die berechtigte TXRRS-Servicegruppe verfügbar ist. Im obigen Beispiel zeigt die Zeile in Fettdruck an, dass die Servicegruppe verfügbar ist.

Diese Servicegruppe muss für den MQ Service Provider verfügbar sein.

c) Der Server wird gestoppt.

Führen Sie die in [Einen Server über die Befehlszeile starten und stoppen](#) beschriebene Prozedur aus.

Ergebnisse

z/OS Connect EE wurde erfolgreich installiert.

Nächste Schritte

Installieren Sie den IBM MQ-Ressourcenadapter.

Zugehörige Tasks

„[IBM z/OS Connect EE - Setup-Prozedur für MQ Service Provider](#)“ auf Seite 470

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect EE konfiguriert werden kann.

„[IBM MQ-Ressourcenadapter installieren](#)“ auf Seite 471

Die IBM MQ-Messaging-Provider-Funktion in IBMz/OS Connect EE verwendet eine Komponente von IBM MQ, den sogenannten IBM MQ-Ressourcenadapter. Der Ressourcenadapter wird mit IBM MQ als Teil der UNIX System Services-Komponenten von IBM MQ for z/OS bereitgestellt.

IBM MQ-Ressourcenadapter installieren

Die IBM MQ-Messaging-Provider-Funktion in IBMz/OS Connect EE verwendet eine Komponente von IBM MQ, den sogenannten IBM MQ-Ressourcenadapter. Der Ressourcenadapter wird mit IBM MQ als Teil der UNIX System Services-Komponenten von IBM MQ for z/OS bereitgestellt.

Vorbereitende Schritte

Stellen Sie sicher, dass die UNIX System Services-Komponenten von IBM MQ for z/OS für die Version des Warteschlangenmanagers, zu dem Sie eine Verbindung herstellen werden, installiert sind.

Wenn Sie eine Verbindung zu mehreren Versionen des Warteschlangenmanagers herstellen wollen, sollten Sie die Funktion in der neuesten Version verwenden.

Informationen zu diesem Vorgang

Mit dieser Prozedur wird der IBM MQ-Ressourcenadapter in IBM z/OS Connect EE installiert.

Vorgehensweise

1. Suchen Sie das Unix System Services-Verzeichnis der Version von IBM MQ, mit der Sie eine Verbindung herstellen werden.

Dieses Verzeichnis kann z. B. `/usr/lpp/mqm/V9R0M1/` genannt werden. Das Verzeichnis sollte ein Unterverzeichnis `java/lib` enthalten, das eine Reihe von nativen Bibliotheken (`.so`-Dateien) enthält.



Achtung: Wenn Sie eine Verbindung zu mehreren Versionen des Warteschlangenmanagers herstellen wollen, verwenden Sie die neueste Version.

2. Bearbeiten Sie die Datei `server.xml` des z/OS Connect EE-Servers.

Fügen Sie die folgenden Zeilen hinzu:

```
<variable name="wmqJmsClient.rar.location"
  value="MQJAVA_LIB_DIR/jca/wmq.jmsra.rar"/>
<wmqJmsClient nativeLibraryPath="MQJAVA_LIB_DIR"/>
```

Dabei basiert `MQJAVA_LIB_DIR` auf dem Verzeichnis in Schritt „1“ auf Seite 472, z. B. `/usr/lpp/mqm/V9R0M1/java/lib`.

- b. Speichern Sie die Änderungen.

Die erste Zeile sagt IBM z/OS Connect EE, wo es den IBM MQ-Ressourcenadapter suchen soll.

Die zweite Zeile teilt dem IBM MQ-Ressourcenadapter mit, wo er die nativen Bibliotheken suchen soll, die für Bindungsverbindungen zu IBM MQ verwendet werden.

Weitere Informationen finden Sie unter [wmqJmsClient](#) und [JMS-Anwendungen in Liberty implementieren](#), um den IBM MQ-Messaging-Provider zu verwenden.

3. Richten Sie die STEPLIB des Servers so ein, dass IBM MQ-Bibliotheken enthalten sind.

Sie müssen dies tun, damit die in Schritt „1“ auf Seite 472 gefundenen nativen Bibliotheken eine Verbindung zum Warteschlangenmanager herstellen können.

In der Regel umfasst dies die Bearbeitung des JCL-Codes, mit dem der Server gestartet wird, um Folgendes zu berücksichtigen:

```
//STEPLIB DD DSN=HLQ.SCSQAUTH,DISP=SHR
// DD DSN=HLQ.SCSQANLE,DISP=SHR
```

Dabei steht `HLQ` für das übergeordnete Qualifikationsmerkmal der Dateien, die die IBM MQ-Installation enthalten.

Wenn Sie eine Verbindung zu mehreren Versionen von IBM MQ von demselben Server herstellen möchten, verwenden Sie die Datensätze aus der neuesten Version.

Ergebnisse

Sie haben einen Teil des IBM MQ-Ressourcenadapters installiert.

Beachten Sie, dass der IBM MQ-Ressourcenadapter erst dann vollständig installiert wird, wenn seine übergeordnete Komponente (`wmqJmsClient-2.0`) aktiviert ist, wie in [„z/OS Connect und MQ Service Provider aktivieren“](#) auf Seite 473 beschrieben.

Nächste Schritte

Installieren Sie den MQ Service Provider in IBMz/OS Connect EE.

Zugehörige Tasks

„IBM z/OS Connect EE - Setup-Prozedur für MQ Service Provider“ auf Seite 470

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect EE konfiguriert werden kann.

„IBM z/OS Connect EE installieren“ auf Seite 470

Die Voraussetzungen und die Vorgehensweise für die Installation von IBMz/OS Connect EE.

„MQ Service Provider in IBM z/OS Connect EE installieren“ auf Seite 473

Sie müssen den MQ Service Provider in IBM z/OS Connect EE installieren, bevor er verwendet wird.

MQ Service Provider in IBM z/OS Connect EE installieren

Sie müssen den MQ Service Provider in IBM z/OS Connect EE installieren, bevor er verwendet wird.

Informationen zu diesem Vorgang

Führen Sie dazu die folgende Prozedur aus:

Vorgehensweise

1. Suchen Sie das Produkterweiterungsverzeichnis von IBM z/OS Connect EE.
Für eine Standardinstallation ist dies das `/var/zosconnect/v2r0/extensions` -Verzeichnis, das hier beschrieben wird: [Produkterweiterungsverzeichnis einrichten](#).
2. Kopieren Sie `MQSP_ROOT/mqzosconnect.properties` in das in Schritt „1“ auf Seite 473 gefundene Verzeichnis.
3. Bearbeiten Sie die kopierte Datei.
Beachten Sie, dass diese Datei eine ASCII-Datei ist. Ändern Sie `PATH_TO_INSTALL` in `MQSP_ROOT/v2.0>` und speichern Sie Ihre Änderungen.



Achtung: Schritt „3“ auf Seite 473 setzt voraus, dass die Benutzer-ID, unter der der Server ausgeführt wird, Lesezugriff auf die Verzeichnisstruktur `MQSP_ROOT` hat. Wenn dies nicht der Fall ist, fügen Sie entweder ausreichend Zugriff auf die Benutzer-ID hinzu oder verschieben Sie den Inhalt von `MQSP_ROOT` an eine Position mit ausreichendem Zugriff.

Ergebnisse

Sie haben jetzt alle Voraussetzungen für den MQ Service Provider installiert.

Nächste Schritte

Sie müssen jetzt den MQ Service Provider und z/OS Connect aktivieren.

Zugehörige Tasks

„IBM z/OS Connect EE - Setup-Prozedur für MQ Service Provider“ auf Seite 470

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect EE konfiguriert werden kann.

„IBM MQ-Ressourcenadapter installieren“ auf Seite 471

Die IBM MQ-Messaging-Provider-Funktion in IBMz/OS Connect EE verwendet eine Komponente von IBM MQ, den sogenannten IBM MQ-Ressourcenadapter. Der Ressourcenadapter wird mit IBM MQ als Teil der UNIX System Services-Komponenten von IBM MQ for z/OS bereitgestellt.

„z/OS Connect und MQ Service Provider aktivieren“ auf Seite 473

Hier erfahren Sie, was Sie tun müssen, um z/OS Connect und den MQ Service Provider zu aktivieren.

z/OS Connect und MQ Service Provider aktivieren

Hier erfahren Sie, was Sie tun müssen, um z/OS Connect und den MQ Service Provider zu aktivieren.

Vorbereitende Schritte

Stellen Sie sicher, dass die Prozeduren abgeschlossen sind:

- „[IBM z/OS Connect EE installieren](#)“ auf Seite 470
- „[IBM MQ-Ressourcenadapter installieren](#)“ auf Seite 471
- „[MQ Service Provider in IBM z/OS Connect EE installieren](#)“ auf Seite 473

Informationen zu diesem Vorgang

Mit dieser Prozedur werden sowohl z/OS Connect als auch der MQ Service Provider aktiviert.

Vorgehensweise

1. Bearbeiten Sie die Datei `server.xml` des z/OS Connect EE-Servers, den Sie erstellt haben, und ersetzen Sie das gesamte **featureManager**-Element durch folgende Zeilen:

```
<featureManager>
  <feature>zosconnect:zosconnect-2.0</feature>
  <feature>appSecurity-2.0</feature>
  <feature>jms-2.0</feature>
  <feature>mqzosconnect:zosConnectMQ-2.0</feature>
  <feature>wmqJmsClient-2.0</feature>
  <feature>zosTransaction-1.0</feature>
</featureManager>
```



Achtung: Diese Einträge nur dann korrigieren, wenn sie noch nicht vorhanden sind.

2. Konfigurieren Sie die Sicherheit für z/OS Connect.
Ausführliche Informationen zur Ausführung dieser Prozedur finden Sie im Abschnitt [Sicherheit für z/OS Connect EE konfigurieren](#).
3. Starten Sie den-Server.

Nächste Schritte

Überprüfen Sie, ob z/OS Connect richtig eingerichtet ist.

Zugehörige Tasks

„[IBM z/OS Connect EE - Setup-Prozedur für MQ Service Provider](#)“ auf Seite 470

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect EE konfiguriert werden kann.

„[Überprüfen, ob z/OS Connect richtig eingerichtet ist](#)“ auf Seite 474

Wie Sie überprüfen, ob z/OS Connect richtig eingerichtet ist.

 **Überprüfen, ob z/OS Connect richtig eingerichtet ist**

Wie Sie überprüfen, ob z/OS Connect richtig eingerichtet ist.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie die im Abschnitt „[z/OS Connect und MQ Service Provider aktivieren](#)“ auf Seite 473 beschriebene Prozedur ausgeführt haben.

Informationen zu diesem Vorgang

z/OS Connect stellt eine RESTful-API bereit, mit der die Services abgefragt werden können, die installiert sind, und Verwaltungsaktionen wie z. B. Stoppen und Starten von Services ausführen.

Vorgehensweise

1. Geben Sie eine HTTP-Anforderung GET für z/OS Connect aus, um die Liste der derzeit installierten Services abzufragen.

Sie können dies mit einem Webbrowser tun, indem Sie eine URL des Formulars eingeben

```
https://HOST_NAME:HTTPS_PORT/zosConnect/services
```

Dabei stehen *HOST_NAME* und *HTTPS_PORT* für die Werte, die Sie in Schritt „1“ auf Seite 470 von „IBM z/OS Connect EE installieren“ auf Seite 470 eingegeben haben.

Zum Beispiel:

```
https://yourdomainname:12342/zosConnect/services
```

2. Geben Sie die Benutzer-ID und das Kennwort ein, wenn Sie vom Browser dazu aufgefordert werden. Dies sind die Werte, die Sie in Schritt „2“ auf Seite 474 von „z/OS Connect und MQ Service Provider aktivieren“ auf Seite 473 in das Benutzerelement eingegeben haben.

Ergebnisse

Dies führt zu der folgenden JSON-Antwort, die angibt, dass z/OS Connect ausgeführt wird, aber noch keine Services installiert sind. Beachten Sie, dass bei Verwendung eines vorhandenen z/OS Connect-Servers, bei dem Services installiert waren, diese Services angezeigt werden.

```
-----  
{  
  "zosConnectServices": [  ]  
}
```

Nächste Schritte

Richten Sie einen einfachen One-Way-IBM MQ-Service für z/OS Connect EE ein.

Zugehörige Tasks

„IBM z/OS Connect EE - Setup-Prozedur für MQ Service Provider“ auf Seite 470

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect EE konfiguriert werden kann.

Einfachen One-Way-Service des MQ Service Provider für z/OS Connect EE einrichten

Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

Vorbereitende Schritte

Stellen Sie sicher, dass alle Komponenten richtig eingerichtet sind, wie in den Abschnitten z/OS Connect und den MQ Service Provider aktivieren und Überprüfen, ob z/OS Connect richtig eingerichtet ist beschrieben.

Vorgehensweise

1. Erstellen Sie eine Warteschlange mit dem Namen ONE_WAY_QUEUE auf dem z/OS-Zielwarteschlangenmanager und verwenden Sie dazu entweder MQSC oder den IBM MQ Explorer.
2. Definieren Sie eine IBM MQ-Messaging-Provider-Verbindungsfactory und eine Warteschlange.

Fügen Sie zu diesem Zweck Folgendes am Ende der Datei `server.xml` des von Ihnen erstellten z/OS Connect EE-Servers, aber innerhalb des Serverelements hinzu.

```
-----  
<jmsConnectionFactory id="cf1" jndiName="jms/cf1" connectionManagerRef="ConMgr1">  
  <properties.wmqJms  
    transportType="BINDINGS"  
    queueManager="MQ21" />  
</jmsConnectionFactory>  
  
<connectionManager id="ConMgr1" maxPoolSize="5" />  
  
<jmsQueue id="q1" jndiName="jms/d1">  
  <properties.wmqJms  
    baseQueueName="ONE_WAY_QUEUE" />  
</jmsQueue>  
-----
```

Anmerkungen:

- a. Ändern Sie den Wert des Attributs **queueManager** in den richtigen Namen des Zielwarteschlangenmanagers.
- b. Es wird ein **transportType** mit Bindungen verwendet. Das bedeutet, dass eine speicherübergreifende Verbindung für die Kommunikation mit dem Warteschlangenmanager verwendet wird. Dies ist der einzige unterstützte **transportType**, wenn der MQ Service Provider verwendet wird.
3. Definieren Sie einen einfachen One-Way-IBM MQ-Service, indem Sie Folgendes zur Datei `server.xml` des von Ihnen erstellten z/OS Connect EE-Servers hinzufügen, aber innerhalb des Serverelements.

```
-----  
<zosconnect_zosConnectService id="zosconnMQ1"  
  invokeURI="/oneWay"  
  serviceName="oneWay"  
  serviceRef="oneWay" />  
  
<mqzosconnect_mqzOSConnectService id="oneWay "  
  connectionFactory="jms/cf1"  
  destination="jms/d1" />  
-----
```

Das Element `zosConnectService` definiert einen neuen Service für z/OS Connect mit dem **serviceName** von `oneWay`. Die:

- Das Attribut **invokeURI** erleichtert das Aufrufen des Service.
- Das Attribut **serviceRef** muss mit dem ID-Attribut eines z/OS Connect-Service-Providers übereinstimmen, der in diesem Fall durch das Element `mqzOSConnectService` bereitgestellt wird.

Das Element `mqzOSConnectService` definiert eine einzelne Serviceinstanz, die vom MQ Service Provider bereitgestellt wird.

Die Attribute **connectionFactory** und **destination** teilen der Instanz mit, wie die Verbindungsfactory bzw. die Warteschlange des IBM MQ-Messaging-Providers lokalisiert werden soll.

Details zu den Attributen in dieser Struktur finden Sie im Abschnitt [ElementmqzOSConnectService](#).

Ergebnisse

Sie haben einen einfachen Einwegservice eingerichtet.

Nächste Schritte

Sie müssen den Service testen.

Zugehörige Tasks

„One-Way-Service für z/OS Connect EE testen“ auf Seite 477

Eine Reihe von Schritten, die sicherstellen, dass Ihr Einwegservice funktioniert.

Vorbereitende Schritte

Stellen Sie sicher, dass „[Einfachen One-Way-Service des MQ Service Provider für z/OS Connect EE einrichten](#)“ auf Seite 475 erfolgreich abgeschlossen wurde.

Vorgehensweise

1. Überprüfen Sie, ob z/OS Connect den neuen Service erkennt.

Führen Sie zu diesem Zweck die Schritte aus, die im Abschnitt [Überprüfen, ob z/OS Connect richtig eingerichtet ist](#) beschrieben sind.

Da jetzt ein Service definiert ist, sollten Sie etwas Ähnliches mit der folgenden Ausgabe sehen:

```
-----  
{  
  "zosConnectServices": [  
    {  
      "ServiceName": "oneWay",  
      "ServiceDescription": "DATA_UNAVAILABLE",  
      "ServiceProvider": "IBM MQ for z/OS service provider for IBM z/OS Connect EE V2.0",  
      "ServiceURL": "https://yourdomainname:12342/zosConnect/services/oneWay"  
    }  
  ]  
}
```

Anmerkung: Der **ServiceDescription** wird als DATA_UNAVAILABLE zurückgegeben, da das Attribut **serviceDescription** nicht im Element zosConnectService angegeben wurde.

2. Geben Sie einen HTTP-POST an den Service aus. Dies führt dazu, dass eine neue Nachricht an die ONE_WAY_QUEUE gesendet wird, unabhängig davon, welche Nutzdaten in der POST-Anforderung angegeben wurden.

Eine gute Möglichkeit, diese Art von Tests auszuführen, ist die Verwendung eines [Advanced REST Client -Plug-ins \(ARC\)](#) für Google Chrome.

- a) Verwenden Sie ARC, um einen HTTP-POST für die folgende URL auszugeben:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

- b) Geben Sie eine gültige JSON-Datei für den Hauptteil an, z. B.:

```
{"name1" : "value1", "name2" : "value2"}
```

Sie müssen zwei Header angeben:

- i) Berechtigung = Basic *ENCODED_USERID_PASSWORD*

Dabei ist *ENCODED_USERID_PASSWORD* eine mit 64 verschlüsselte Basisbenutzer-ID und ein Kennwort, wie im Abschnitt [Basisauthentifizierungsschema](#) beschrieben.

- ii) Inhaltstyp = application/json

Der erste Header wird für die Anmeldung bei z/OS Connect verwendet und ist eine verschlüsselte Form der Benutzer-ID und des Kennworts, die in [z/OS Connect](#) und den [MQ Service Provider aktivieren](#) verwendet werden.

Der zweite Header informiert den Server, dass es sich um JSON-Nutzdaten handelt. Dies ist der einzige unterstützte Wert, wenn HTTP-POST-Aufrufe an den MQ Service Provider ausgegeben werden.

Wenn Sie diesen Header nicht angeben, wird ein HTTP 415-Fehler angezeigt, der auf einen nicht unterstützten Medientyp hinweist.

Wenn Sie eine HTTP-Anforderung POST an einen Einweg-Service absetzen, wird ein leerer HTTP-Antworttext und ein HTTP-Antwortcode 204 (kein Inhalt) zurückgegeben.

ARC gibt eine Nachricht aus, die besagt, dass die Antwort keine Daten enthält.

Verwenden Sie zum Überprüfen, ob eine Nachricht eingereicht wurde, ein Tool wie den IBM MQ Explorer, um den Inhalt von ONE_WAY_QUEUE anzuzeigen. Es sollte eine IBM MQ-Nachricht im Format MQSTR mit den gesendeten JSON-Nutzdaten in der Warteschlange enthalten sein.

3. Verwenden Sie ARC, um eine HTTP-Anweisung DELETE für die folgende URL auszugeben:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

Dadurch wird die Nachricht, die gerade an den Einweg-Service gesendet wurde, destruktiv abgerufen.

Dieser sollte einen Antwortteil mit den ursprünglich gesendeten JSON-Nutzdaten angeben. Sie können mithilfe von IBM MQ Explorer bestätigen, dass für die Nachricht ein Abruf mit Löschen durchgeführt wurde.

Zugehörige Tasks

„Einfachen One-Way-Service des MQ Service Provider für z/OS Connect EE einrichten“ auf Seite 475
Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

Einfachen Two-Way-Service des IBM MQ für z/OS Connect EE einrichten

In diesem Abschnitt wird beschrieben, wie ein Two-Way-IBM MQ-Service in z/OS Connect EE für ein vorhandenes Paar von IBM MQ-Warteschlangen mit den Namen MQ_REQUEST_Q und MQ_REPLY_Q eingerichtet wird.

Vorbereitende Schritte

Stellen Sie sicher, dass alle Komponenten ordnungsgemäß eingerichtet sind, wie im Abschnitt „z/OS Connect und MQ Service Provider aktivieren“ auf Seite 485 and „Überprüfen, ob z/OS Connect richtig eingerichtet ist“ auf Seite 486 beschrieben.

Informationen zu diesem Vorgang

Dies ist etwas komplexer als die Einrichtung eines One-Way-Service, da Sie eine Back-End-Anwendung benötigen, um die Anforderungsnachrichten zu konsumieren, die mit z/OS Connect gesendet werden, und Antwortnachrichten zu generieren.

Bei dieser Aufgabe wird vorausgesetzt, dass die Warteschlangen MQ_REQUEST_Q und MQ_REPLY_Q von einer vorhandenen Back-End-Anwendung, z. B. einer CICS- oder IMS-Transaktion, verwendet werden, die eine Anforderungsnachricht aus MQ_REQUEST_Q abrufen und eine Antwortnachricht generiert, die in MQ_REPLY_Q eingereicht wird.

Vorgehensweise

1. Definieren Sie eine IBM MQ-Messaging-Provider-Verbindungsfactory und zwei IBM MQ-Messaging-Provider-Warteschlangen für den z/OS Connect EE-Server.

Die IBM MQ-Messaging-Provider-Warteschlangen bilden die vorhandenen Warteschlangen MQ_REQUEST_Q und MQ_REPLY_Q ab. Fügen Sie zu diesem Zweck Folgendes am Ende der Datei `server.xml` des von Ihnen erstellten z/OS Connect EE-Servers, aber innerhalb des Serverelements hinzu.

```
<jmsConnectionFactory id="cf2" jndiName="jms/cf2"
  connectionManagerRef="ConMgr2">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>
```

```

<connectionManager id="ConMgr2" maxPoolSize="5"/>
<jmsQueue id="requestQueue" jndiName="jms/requestQueue">
  <properties.wmqJms baseQueueName="MQ_REQUEST_Q"/>
</jmsQueue>
<jmsQueue id="replyQueue" jndiName="jms/replyQueue">
  <properties.wmqJms baseQueueName="MQ_REPLY_Q"/>
</jmsQueue>

```

Anmerkungen:

- a. Ändern Sie den Wert des Attributs **queueManager** in den richtigen Namen des Zielwarteschlangenmanagers.
 - b. Abhängig von der Art der Back-End-Anwendung müssen Sie möglicherweise weitere Konfigurationen dieser Elemente vornehmen. Weitere Informationen finden Sie unter [„MQ Service Provider konfigurieren“](#) auf Seite 491.
 - c. Das Attribut **maxPoolSize** muss abhängig von der Anzahl gleichzeitig ablaufender Anforderungen angepasst werden. Weitere Informationen zu diesem Attribut finden Sie unter [Verbindungs-Pooling für JMS-Verbindungen konfigurieren](#).
2. Definieren Sie den Two-Way-IBM MQ-Service, indem Sie Folgendes zur Datei `server.xml` des von Ihnen erstellten z/OS Connect EE-Servers hinzufügen, aber innerhalb des Serverelements.

```

-----
<zosconnect_zosConnectService id="zosconnMQ2"
  invokeURI="/twoWay"
  serviceName="twoWay"
  serviceRef="twoWay"/>

<mqzosconnect_mqzOSConnectService id="twoWay"
  connectionFactory="jms/cf2"
  destination="jms/requestQueue"
  replyDestination="jms/replyQueue"
  waitInterval="10000"/>
-----

```

Anmerkungen:

- a. Abhängig von der Art der Back-End-Anwendung müssen Sie möglicherweise weitere Konfigurationen dieser Elemente vornehmen. Sie könnten beispielsweise eine Datenkonvertierung benötigen. Weitere Informationen finden Sie unter [„MQ Service Provider konfigurieren“](#) auf Seite 491.
- b. Möglicherweise müssen Sie den Wert von **waitInterval** anpassen. Weitere Informationen hierzu finden Sie unter [waitInterval](#).
- c. In diesem Beispiel wird angenommen, dass Nachrichten, die von der Back-End-Anwendung an das Antwortziel gestellt werden, eine Korrelations-ID aufweisen, die mit der Nachrichten-ID auf dem Anforderungsziel identisch ist.

Wenn dies nicht der Fall ist, müssen Sie das Attribut **replySelection** zum Element `mqzosconnect_mqzOSConnectService` hinzufügen und den Wert entsprechend festlegen.

Weitere Informationen finden Sie im Abschnitt [replySelection](#).

Ergebnisse

Sie haben einen Two-Way-Service in z/OS Connect EE eingerichtet, der mit einer vorhandenen Back-End-Anwendung, die ein Paar von IBM MQ-Warteschlangen nutzt, verwendet werden kann.

Nächste Schritte

Sie können den zweiseitigen Service aufrufen, der einen HTTP-POST-Befehl an die URL `https://host_name:port_no/twoWay` absetzt.

Diese Schritte sind mit den in [„One-Way-Service für z/OS Connect EE testen“](#) auf Seite 477 beschriebenen Schritten vergleichbar, außer dass nur das Verb HTTP POST unterstützt wird und Sie JSON-Nutzdaten senden sollten, die für die vorhandene Back-End-Anwendung geeignet sind.

Zugehörige Tasks

„Einfachen One-Way-Service des MQ Service Provider für z/OS Connect EE einrichten“ auf Seite 475
Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

Generieren einer Servicearchivierungsdatei (SAR)

Bevor Sie eine API mit z/OS Connect EE erstellen können, müssen Sie eine SAR-Datei generieren.

Informationen zu diesem Vorgang

Sie können dies mit IBM MQerreichen, indem Sie den CICS-WOLA-Prozess verwenden, der in den Informationen zu *z/OS Connect EE* in IBM Documentation beschrieben ist. Weitere Informationen finden Sie im Abschnitt [Servicearchivierung aus einem CICS COBOL-Copybook generieren](#).

Wichtig: Der im folgenden Text beschriebene Prozess ist optional. Wenn Sie die vom API-Editor bereitgestellten Funktionen nicht nutzen wollen, benötigen Sie die SAR-Datei nicht. Möglicherweise möchten Sie jedoch die JSON-Schemata und Bindedateien verwenden, die im vorherigen Link beschrieben sind, wenn Sie die Datenumsetzungsunterstützung verwenden möchten, die von z/OS Connect EE bereitgestellt wird.

Der im vorherigen Link beschriebene Prozess verwendet das Dienstprogramm **BAQLS2JS**, das mit z/OS Connect EE bereitgestellt wird.

Dieses Dienstprogramm:

1. Generiert JSON-Schemata und Bindedateien, die mit der in z/OS Connect EE bereitgestellten Datenumsetzungsfunktion verwendet werden können.

Wenn Sie diese Dateien in z/OS Connect EE konfigurieren möchten, geben Sie sie in der Konfiguration mit dem Element `zosconnect_zosConnectDataXform` an.

Weitere Informationen finden Sie unter [„Datenumsetzungen mit dem MQ Service Provider“](#) auf Seite 495.

2. Generiert eine SAR-Datei, die die JSON-Schemata aus dem vorhergehenden Schritt enthält, und eine Referenz auf das Attribut **serviceName** einer bestimmten z/OS Connect EE-Serviceinstanz.

Die Referenz wird **BAQLS2JS** mithilfe des Parameters `SERVICE-NAME` bereitgestellt.

Beispiel: Wenn `SERVICE-NAME=MQ1WayBackend` als Attribut für **BAQLS2JS** festgelegt wird, generiert **BAQLS2JS** eine SAR, die mit einer IBM MQ-Serviceinstanz verwendet werden kann, die wie folgt konfiguriert ist:

```
<zosconnect_zosConnectService id="MQ1WayService"
    serviceName="MQ1WayBackend"
    serviceRef="mq1way"
    invokeURI="/mq1way"
    dataXformRef="xformJSON2Byte"/>

<mqzosconnect_mqzOSConnectService id="mq1way"
    connectionFactory="jms/cf1"
    destination="jms/oneWayRequestQ"/>
```

Ergebnisse

Sobald die SAR-Datei generiert wurde, kann sie in den API-Editor importiert werden, um eine API zu generieren. Weitere Informationen hierzu finden Sie im Abschnitt [API mit dem z/OS Connect EE-API-Editor erstellen](#).

Nachdem eine API im API-Editor erstellt wurde, wird ein API-Archiv (AAR) generiert, das dann in z/OS Connect EE bereitgestellt werden kann, wo es für den Service ausgeführt wird, auf den das Attribut `SERVICE-NAME` in **BAQLS2JS** verweist. Weitere Informationen finden Sie unter [API für den z/OS Connect EE-Server implementieren](#).

Zugehörige Tasks

„Einfachen One-Way-Service des MQ Service Provider für z/OS Connect EE einrichten“ auf Seite 475
Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

„Einfachen Two-Way-Service des IBM MQ für z/OS Connect EE einrichten“ auf Seite 478

In diesem Abschnitt wird beschrieben, wie ein Two-Way-IBM MQ-Service in z/OS Connect EE für ein vorhandenes Paar von IBM MQ-Warteschlangen mit den Namen MQ_REQUEST_Q und MQ_REPLY_Q eingerichtet wird.

IBM z/OS Connect V1 - Setup-Prozedur für MQ Service Provider

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect V1 konfiguriert werden kann.

Informationen zu diesem Vorgang

Führen Sie die folgenden Aufgaben in der angegebenen Reihenfolge aus, um den MQ Service Provider und alle zugehörigen Voraussetzungen zu installieren.

WebSphere Application Server Liberty installieren

Die Voraussetzungen und die Vorgehensweise für die Installation von WebSphere Application Server Liberty (WLP).

Vorbereitende Schritte

Stellen Sie sicher, dass Sie Zugriff auf eine Version von WLP für z/OS haben, die z/OS Connect enthält, z. B. 16.0.0.2.

Informationen zu diesem Vorgang

Mit dieser Prozedur wird WLP installiert und ein Server installiert.

Vorgehensweise

1. Installieren Sie WLP, indem Sie die in [Liberty mit Installation Manager installieren](#) beschriebene Prozedur ausführen.
Beachten Sie, dass für den Rest dieser Anweisungen diese Verzeichnisstruktur als *WLP_ROOT* bezeichnet wird.
2. Erstellen Sie einen neuen Server, indem Sie die im Abschnitt [Liberty-Server manuell erstellen](#) beschriebene Prozedur ausführen.
In [z/OS Connect einrichten](#) wird beschrieben, wie z/OS Connect auf dem erstellten Server aktiviert wird.



Achtung: Sie müssen die optionalen Schritte *WebSphere optimierte lokale Adapter (WOLA)* nicht ausführen.

3. Aktivieren Sie den TXRRS-autorisierten Service, indem Sie die Prozedur ausführen, die im Abschnitt [berechtigte z/OS-Services in Liberty für z/OS aktivieren](#) beschrieben ist.
4. Überprüfen Sie, ob der TXRRS-autorisierte Service korrekt konfiguriert wurde:
 - a) Der Server wird gestartet.
Führen Sie die in [Einen Server über die Befehlszeile starten und stoppen](#) beschriebene Prozedur aus.
 - b) Zeigen Sie die Serverprotokolle an, die sich unter befinden:

WLP_ROOT/usr/servers/server_name/logs/message.log

Hierbei handelt es sich um ASCII-Dateien, die eine Ausgabe ähnlich der folgenden enthalten sollten:

```
A CWWKE0001I: The server server1 has been launched.
I CWWKB0103I: Authorized service group LOCALCOM is available.
I CWWKB0103I: Authorized service group PRODMGR is available.
I CWWKB0103I: Authorized service group SAFCRE is available.
I CWWKB0103I: Authorized service group TXRRS is available.
I CWWKB0103I: Authorized service group WOLA is available.
I CWWKB0103I: Authorized service group ZOSDUMP is available.
I CWWKB0103I: Authorized service group ZOSWLM is available.
I CWWKB0103I: Authorized service group CLIENT.WOLA is available.
I CWWKB0108I: IBM CORP product WAS FOR Z/OS version 16.0 successfully registered with z/OS.
```

Überprüfen Sie die Ausgabe, um festzustellen, ob die berechtigte TXRRS-Servicegruppe verfügbar ist. Im obigen Beispiel zeigt die Zeile in Fettdruck an, dass die Servicegruppe verfügbar ist.

Diese Servicegruppe muss für den MQ Service Provider verfügbar sein.

c) Der Server wird gestoppt.

Führen Sie die in [Einen Server über die Befehlszeile starten und stoppen](#) beschriebene Prozedur aus.

Ergebnisse

Sie haben WLP erfolgreich installiert.

Nächste Schritte

Installieren Sie den IBM MQ-Ressourcenadapter.

Zugehörige Tasks

„[IBM z/OS Connect V1 - Setup-Prozedur für MQ Service Provider](#)“ auf Seite 481

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect V1 konfiguriert werden kann.

„[IBM MQ-Ressourcenadapter installieren](#)“ auf Seite 482

Die IBM MQ-Messaging-Provider-Funktion in WLP verwendet eine Komponente von IBM MQ, den sogenannten IBM MQ-Ressourcenadapter. Aus Kompatibilitätsgründen müssen Benutzer von z/OS Connect V1 unabhängig von der Version des Warteschlangenmanagers, zu dem sie eine Verbindung herstellt, den Ressourcenadapter von IBM WebSphere MQ 7.5 verwenden.

IBM MQ-Ressourcenadapter installieren

Die IBM MQ-Messaging-Provider-Funktion in WLP verwendet eine Komponente von IBM MQ, den sogenannten IBM MQ-Ressourcenadapter. Aus Kompatibilitätsgründen müssen Benutzer von z/OS Connect V1 unabhängig von der Version des Warteschlangenmanagers, zu dem sie eine Verbindung herstellt, den Ressourcenadapter von IBM WebSphere MQ 7.5 verwenden.

Vorbereitende Schritte

Rufen Sie den Abschnitt [IBM MQ Resource Adapter for WLP abrufen](#) auf, in dem beschrieben wird, wie Sie eine Verbindung zu Fix Central herstellen und den neuesten Ressourcenadapter herunterladen.

Der Ressourcenadapter wird im folgenden Format angezeigt: `7.5.0.Fix_level-WS-MQ-Java-InstallRA.jar`. Beachten Sie, dass es sich um diese JAR-Datei handelt, auf die sich die Prozedur bezieht.

Informationen zu diesem Vorgang

Mit dieser Prozedur wird der IBM MQ-Ressourcenadapter in WLP installiert.

Vorgehensweise

1. Extrahieren Sie den Inhalt der JAR-Datei, rufen Sie das Verzeichnis `wmq` auf, und übertragen Sie die Datei `wmq.jmsra.rar` in einem entsprechenden Verzeichnis im Zielfilesystem in ein geeignetes Verzeichnis.

Dieses Verzeichnis wird als `MQRA_ROOT` für den Rest dieser Anweisungen bezeichnet.

2. Suchen Sie das UNIX System Services Components-Verzeichnis des Ziel-WS-Managers.
Dieses Verzeichnis kann z. B. `/mqm/V9R0M1/` genannt werden. Das Verzeichnis sollte ein Unterverzeichnis `java/lib` enthalten, das eine Reihe von nativen Bibliotheken (`.so`-Dateien) enthält.

3. Bearbeiten Sie die `server.xml`-Datei des Servers.

Fügen Sie die folgenden Zeilen hinzu:

```
<variable name="wmqJmsClient.rar.location"
  value="MQRA_ROOT/wmq.jmsra.rar"/>
<wmqJmsClient nativeLibraryPath="MQJAVA_LIB_DIR" />
```

Dabei steht `MQJAVA_LIB_DIR` für das Verzeichnis in Schritt „2“ auf Seite 483, z.

B. `/usr/lpp/mqm/V9R0M1/java/lib`.

- b. Speichern Sie die Änderungen.

Die erste Zeile teilt WLP mit, wo der IBM MQ-Ressourcenadapter zu finden ist.

Die zweite Zeile teilt dem IBM MQ-Ressourcenadapter mit, wo er die nativen Bibliotheken suchen soll, die für Bindungsverbindungen zu IBM MQ verwendet werden.

Weitere Informationen finden Sie unter [wmqJmsClient](#) und [JMS-Anwendungen in Liberty implementieren](#), um den IBM MQ-Messaging-Provider zu verwenden.

4. Richten Sie die STEPLIB ein.

Sie müssen dies tun, damit die in Schritt „2“ auf Seite 483 gefundenen nativen Bibliotheken eine Verbindung zum Warteschlangenmanager herstellen können.

Führen Sie den folgenden Befehl in der Shell aus, die zum Starten des WLP-Servers verwendet wird:

```
export STEPLIB=HLQ.SCSQAUTH:HLQ.SCSQANLE
```

Dabei steht `HLQ` für das übergeordnete Qualifikationsmerkmal der Dateien, die die IBM MQ-Installation enthalten.

Ergebnisse

Sie haben einen Teil des IBM MQ-Ressourcenadapters installiert. Beachten Sie, dass der IBM MQ-Ressourcenadapter erst dann vollständig installiert wird, wenn seine übergeordnete Komponente (`wmqJmsClient-1.1`) aktiviert ist.

Nächste Schritte

Installieren Sie den MQ Service Provider in WLP.

Zugehörige Tasks

„[IBM z/OS Connect V1 - Setup-Prozedur für MQ Service Provider](#)“ auf Seite 481

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect V1 konfiguriert werden kann.

„[WebSphere Application Server Liberty installieren](#)“ auf Seite 481

Die Voraussetzungen und die Vorgehensweise für die Installation von WebSphere Application Server Liberty (WLP).

„[MQ Service Provider in WLP installieren](#)“ auf Seite 484

Sie müssen den MQ Service Provider in WLP installieren, bevor er verwendet wird.

Sie müssen den MQ Service Provider in WLP installieren, bevor er verwendet wird.

Vorbereitende Schritte

Fordern Sie den MQ Service Provider an und installieren Sie ihn mithilfe der Informationen im Abschnitt „[MQ Service Provider anfordern](#)“ auf Seite 469.

Informationen zu diesem Vorgang

Der MQ Service Provider ist eine WLP-Standardkomponente, die Sie mit einer der Methoden installieren können, die unter [Liberty-Features packen und installieren](#) beschrieben werden.

Unter [Verzeichnispositionen und -eigenschaften](#) erfahren Sie, was `${wlp.install.dir}` und `${wlp.user.dir}` bedeuten.

Der folgende Text gibt einen Überblick über die Methoden.

Vorgehensweise

1. Installieren Sie das Feature in den WLP-Kernel.



Achtung: Für diese Methode ist Schreibzugriff auf das WLP-Installationsverzeichnis erforderlich. Wenn Sie keinen Schreibzugriff auf das WLP-Installationsverzeichnis haben, ist Option „2“ auf Seite 484 wahrscheinlich besser geeignet.

Dazu kopieren Sie die folgenden Schritte:

- a) `MQSP_ROOT/v1.0/lib/com.ibm.mq.zosconnect_1.0.0.jar` auf `${wlp.install.dir}/lib`.
 - b) `MQSP_ROOT/v1.0/lib/features/zosConnectMQ-1.0.mf` auf `${wlp.install.dir}/lib/features`.
2. Installieren Sie die Komponente in der Benutzerkonfiguration.
Gehen Sie wie folgt vor, um diese
 - a) `MQSP_ROOT/v1.0/lib/com.ibm.mq.zosconnect_1.0.0.jar` auf `{wlp.user.dir}/extension/lib`.
 - b) `MQSP_ROOT/v1.0/lib/features/zosConnectMQ-1.0.mf` auf `{wlp.user.dir}/extension/lib/features`.
 3. Installieren Sie das Feature als Produkterweiterung.
Gehen Sie dazu wie folgt vor:
 - a) Kopieren Sie `MQSP_ROOT/mqzosconnect.properties` in `${wlp.install.dir}/etc/extensions`.
 - b) Bearbeiten Sie die kopierte Datei.
Beachten Sie, dass diese Datei eine ASCII-Datei ist. Ändern Sie `PATH_TO_INSTALL` in `MQSP_ROOT/v1.0` und speichern Sie Ihre Änderungen.

Ergebnisse

Sie haben jetzt alle Voraussetzungen für den MQ Service Provider installiert.

Anmerkung: Die Methode, die Sie zur Installation des MQ Service Provider verwenden, wirkt sich darauf aus, wie in `server.xml` Bezug auf ihn genommen wird. Beispiele werden in [„z/OS Connect und MQ Service Provider aktivieren“](#) auf Seite 485 gezeigt.

Nächste Schritte

Sie müssen jetzt den MQ Service Provider und z/OS Connect aktivieren.

Zugehörige Tasks

„IBM z/OS Connect V1 - Setup-Prozedur für MQ Service Provider“ auf Seite 481

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect V1 konfiguriert werden kann.

„IBM MQ-Ressourcenadapter installieren“ auf Seite 482

Die IBM MQ-Messaging-Provider-Funktion in WLP verwendet eine Komponente von IBM MQ, den sogenannten IBM MQ-Ressourcenadapter. Aus Kompatibilitätsgründen müssen Benutzer von z/OS Connect V1 unabhängig von der Version des Warteschlangenmanagers, zu dem sie eine Verbindung herstellt, den Ressourcenadapter von IBM WebSphere MQ 7.5 verwenden.

„z/OS Connect und MQ Service Provider aktivieren“ auf Seite 485

Hier erfahren Sie, was Sie tun müssen, um z/OS Connect und den MQ Service Provider zu aktivieren.

z/OS Connect und MQ Service Provider aktivieren

Hier erfahren Sie, was Sie tun müssen, um z/OS Connect und den MQ Service Provider zu aktivieren.

Vorbereitende Schritte

Stellen Sie sicher, dass die Prozeduren abgeschlossen sind:

- „WebSphere Application Server Liberty installieren“ auf Seite 481
- „IBM MQ-Ressourcenadapter installieren“ auf Seite 482
- „MQ Service Provider in WLP installieren“ auf Seite 484

Informationen zu diesem Vorgang

Mit dieser Prozedur werden sowohl z/OS Connect als auch der MQ Service Provider aktiviert.

Vorgehensweise

1. Bearbeiten Sie den `server.xml`, den Sie erstellt haben, und ersetzen Sie das gesamte Element **featureManager** durch diese Zeilen.



Achtung: Das genaue Format der Zeile `<feature>zosConnectMQ-1.0</feature>` hängt von der Methode ab, die Sie zum Installieren von MQ Service Provider verwendet haben (siehe Beschreibung in „MQ Service Provider in WLP installieren“ auf Seite 484).

Wenn Sie Folgendes verwendet haben:

- Methode „1“ auf Seite 484: Die Zeile wird als `<feature>zosConnectMQ-1.0</feature>` angezeigt.
- Methode „2“ auf Seite 484: Die Zeile wird als `<feature>usr:zosConnectMQ-1.0</feature>` angezeigt.
- Methode „3“ auf Seite 484: Die Zeile wird als `<feature>mqzosconnect:zosConnectMQ-1.0</feature>` angezeigt.

Außerdem sollten Sie die folgenden Einträge nur ändern, wenn sie noch nicht vorhanden sind.

```
<featureManager>
<feature>zosConnect-1.0</feature>
<feature>appSecurity-2.0</feature>
<feature>zosConnectMQ-1.0</feature>
<feature>wmqJmsClient-1.1</feature>
<feature>zosTransaction-1.0</feature>
</featureManager>
```

2. Konfigurieren Sie die Sicherheit für z/OS Connect.

Ausführliche Informationen zur Ausführung dieser Prozedur finden Sie im Abschnitt [Sicherheit für z/OS Connect konfigurieren](#).

3. Starten Sie den-Server.

Führen Sie die in [Einen Server über die Befehlszeile starten und stoppen](#) beschriebene Prozedur aus.

Nächste Schritte

Überprüfen Sie, ob z/OS Connect richtig eingerichtet ist.

Zugehörige Tasks

„[IBM z/OS Connect V1 - Setup-Prozedur für MQ Service Provider](#)“ auf Seite 481

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect V1 konfiguriert werden kann.

„[Überprüfen, ob z/OS Connect richtig eingerichtet ist](#)“ auf Seite 486

Wie Sie überprüfen, ob z/OS Connect richtig eingerichtet ist.

Überprüfen, ob z/OS Connect richtig eingerichtet ist

Wie Sie überprüfen, ob z/OS Connect richtig eingerichtet ist.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie die im Abschnitt [„z/OS Connect und MQ Service Provider aktivieren“](#) auf Seite 485 beschriebene Prozedur ausgeführt haben.

Informationen zu diesem Vorgang

z/OS Connect stellt eine RESTful-API bereit, mit der die Services abgefragt werden können, die installiert sind, und Verwaltungsaktionen wie z. B. Stoppen und Starten von Services ausführen.

Vorgehensweise

1. Geben Sie eine HTTP-Anforderung GET für z/OS Connect aus, um die Liste der derzeit installierten Services abzufragen.

Sie können dies mit einem Webbrowser tun, indem Sie eine URL des Formulars eingeben

```
https://HOST_NAME:HTTPS_PORT/zosConnect/services
```

Dabei stehen *HOST_NAME* und *HTTPS_PORT* für die Werte, die Sie in Schritt „2“ auf Seite 481 von [„WebSphere Application Server Liberty installieren“](#) auf Seite 481 eingegeben haben.

Zum Beispiel:

```
https://yourdomainname:12342/zosConnect/services
```

2. Geben Sie die Benutzer-ID und das Kennwort ein, wenn Sie vom Browser dazu aufgefordert werden.

Dies sind die Werte, die Sie in Schritt „2“ auf Seite 485 von [„z/OS Connect und MQ Service Provider aktivieren“](#) auf Seite 485 in das Benutzerelement eingegeben haben.

Ergebnisse

Dies führt zu der folgenden JSON-Antwort, die angibt, dass z/OS Connect ausgeführt wird, aber keine Services installiert haben:

```
-----  
{  
  "zosConnectServices": [  ]  
}
```

Nächste Schritte

„Einfachen One-Way-Service des MQ Service Provider für z/OS Connect V1 einrichten“ auf Seite 487

Zugehörige Tasks

„IBM z/OS Connect V1 - Setup-Prozedur für MQ Service Provider“ auf Seite 481

Eine Übersicht über die Komponenten, die Sie installieren müssen, damit der MQ Service Provider für IBM z/OS Connect V1 konfiguriert werden kann.

Einfachen One-Way-Service des MQ Service Provider für z/OS Connect V1 einrichten

Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

Vorbereitende Schritte

Stellen Sie sicher, dass alle Komponenten richtig eingerichtet sind, wie in den Abschnitten [z/OS Connect](#) und den [MQ Service Provider aktivieren und Überprüfen, ob z/OS Connect richtig eingerichtet ist](#) beschrieben.

Vorgehensweise

1. Erstellen Sie eine Warteschlange mit dem Namen ONE_WAY_QUEUE auf dem z/OS-Zielwarteschlangenmanager und verwenden Sie dazu entweder MQSC oder den IBM MQ Explorer.
2. Definieren Sie eine IBM MQ-Messaging-Provider-Verbindungsfactory und eine Warteschlange.
Dies erfolgt durch Hinzufügen der folgenden Elemente zum unteren Teil von `server.xml`, aber innerhalb des Serverelements.

```
<jmsConnectionFactory id="cf1" jndiName="jms/cf1" connectionManagerRef="ConMgr1">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21"/>
</jmsConnectionFactory>

<connectionManager id="ConMgr1" maxPoolSize="5"/>

<jmsQueue id="q1" jndiName="jms/d1">
  <properties.wmqJms
    baseQueueName="ONE_WAY_QUEUE"/>
</jmsQueue>
```

Anmerkungen:

- a. Ändern Sie den Wert des Attributs **queueManager** in den richtigen Namen des Zielwarteschlangenmanagers.
 - b. Es wird ein **transportType** mit Bindungen verwendet. Das bedeutet, dass eine speicherübergreifende Verbindung für die Kommunikation mit dem Warteschlangenmanager verwendet wird. Dies ist der einzige unterstützte **transportType**, wenn der MQ Service Provider verwendet wird.
3. Definieren Sie einen einfachen One-Way-Service des MQ Service Provider, indem Sie Folgendes zur `server.xml` hinzufügen, aber innerhalb des Serverelements.

```
<zosConnectService id="zosconnMQ1"
  invokeURI="/oneWay"
  serviceName="oneWay"
  serviceRef="oneWay"/>

<mqzOSConnectService id="oneWay"
  connectionFactory="jms/cf1"
```

```
destination="jms/d1"/>
```

Das Element `zosConnectService` definiert einen neuen Service für z/OS Connect mit dem **serviceName** von `oneWay`. Die:

- Das Attribut **invokeURI** erleichtert das Aufrufen des Service.
- Das Attribut **serviceRef** muss mit dem ID-Attribut eines z/OS Connect-Service-Providers übereinstimmen, der in diesem Fall durch das Element `mqzOSConnectService` bereitgestellt wird.

Das Element `mqzOSConnectService` definiert eine einzelne Serviceinstanz, die vom MQ Service Provider bereitgestellt wird.

Die Attribute **connectionFactory** und **destination** teilen der Instanz mit, wie die Verbindungs-factory bzw. die Warteschlange des IBM MQ-Messaging-Providers lokalisiert werden soll.

Details zu den Attributen in dieser Struktur finden Sie im Abschnitt [ElementmqzOSConnectService](#).

Ergebnisse

Sie haben einen einfachen Einwegservice eingerichtet.


Nächste Schritte

Sie müssen den Service testen.

Zugehörige Tasks

„One-Way-Service für z/OS Connect V1 testen“ auf Seite 488

Eine Reihe von Schritten, die sicherstellen, dass Ihr Einwegservice funktioniert.

 *One-Way-Service für z/OS Connect V1 testen*

Eine Reihe von Schritten, die sicherstellen, dass Ihr Einwegservice funktioniert.

Vorbereitende Schritte

Stellen Sie sicher, dass „Einfachen One-Way-Service des MQ Service Provider für z/OS Connect V1 einrichten“ auf Seite 487 erfolgreich abgeschlossen wurde.

Vorgehensweise

1. Überprüfen Sie, ob z/OS Connect den neuen Service erkennt.

Führen Sie zu diesem Zweck die Schritte aus, die im Abschnitt [Überprüfen, ob z/OS Connect richtig eingerichtet ist](#) beschrieben sind.

Da jetzt ein Service definiert ist, sollten Sie etwas Ähnliches mit der folgenden Ausgabe sehen:

```
-----  
{  
  "zosConnectServices": [  
    {  
      "serviceName": "oneWay",  
      "serviceDescription": "DATA_UNAVAILABLE",  
      "serviceProvider": "IBM MQ for z/OS service provider for IBM z/OS Connect" V1.0,  
      "serviceURL": "https://yourdomainname:12342/zosConnect/services/oneWay"  
    }  
  ]  
}
```

Anmerkung: Der **ServiceDescription** wird als `DATA_UNAVAILABLE` zurückgegeben, da das Attribut **serviceDescription** nicht im Element `zosConnectService` angegeben wurde.

2. Geben Sie einen HTTP-POST an den Service aus. Dies führt dazu, dass eine neue Nachricht an die ONE_WAY_QUEUE gesendet wird, unabhängig davon, welche Nutzdaten in der POST-Anforderung angegeben wurden.

Eine gute Möglichkeit, diese Art von Tests auszuführen, ist die Verwendung eines [Advanced REST Client -Plug-ins \(ARC\)](#) für Google Chrome.

- a) Verwenden Sie ARC, um einen HTTP-POST für die folgende URL auszugeben:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

- b) Geben Sie eine gültige JSON-Datei für den Hauptteil an, z. B.:

```
{"name1" : "value1", "name2" : "value2"}
```

Sie müssen zwei Header angeben:

- i) Berechtigung = Basic *ENCODED_USERID_PASSWORD*

Dabei ist *ENCODED_USERID_PASSWORD* eine mit 64 verschlüsselte Basisbenutzer-ID und ein Kennwort, wie im Abschnitt [Basisauthentifizierungsschema](#) beschrieben.

- ii) Inhaltstyp = application/json

Der erste Header wird für die Anmeldung bei z/OS Connect verwendet und ist eine verschlüsselte Form der Benutzer-ID und des Kennworts, die in [„z/OS Connect und MQ Service Provider aktivieren“](#) auf Seite 485 verwendet werden.

Der zweite Header informiert den Server, dass es sich um JSON-Nutzdaten handelt. Dies ist der einzige unterstützte Wert, wenn HTTP-POST-Aufrufe an den MQ Service Provider ausgegeben werden.

Wenn Sie diesen Header nicht angeben, wird ein HTTP 415-Fehler angezeigt, der auf einen nicht unterstützten Medientyp hinweist.

Wenn Sie eine HTTP-Anforderung POST an einen Einweg-Service absetzen, wird ein leerer HTTP-Antworttext und ein HTTP-Antwortcode 204 (kein Inhalt) zurückgegeben.

ARC gibt eine Nachricht aus, die besagt, dass die Antwort keine Daten enthält.

Verwenden Sie zum Überprüfen, ob eine Nachricht eingereicht wurde, ein Tool wie den IBM MQ Explorer, um den Inhalt von ONE_WAY_QUEUE anzuzeigen. Es sollte eine IBM MQ-Nachricht im Format MQSTR mit den gesendeten JSON-Nutzdaten in der Warteschlange enthalten sein.

3. Verwenden Sie ARC, um eine HTTP-Anweisung DELETE für die folgende URL auszugeben:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

Dadurch wird die Nachricht, die gerade an den Einweg-Service gesendet wurde, destruktiv abgerufen.

Dieser sollte einen Antwortteil mit den ursprünglich gesendeten JSON-Nutzdaten angeben. Sie können mithilfe von IBM MQ Explorer bestätigen, dass für die Nachricht ein Abruf mit Löschen durchgeführt wurde.

Zugehörige Tasks

[„Einfachen One-Way-Service des MQ Service Provider für z/OS Connect V1 einrichten“](#) auf Seite 487
Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

Einfachen Two-Way-Service des IBM MQ für z/OS Connect Version 1 einrichten

In diesem Abschnitt wird beschrieben, wie ein Two-Way-IBM MQ-Service in z/OS Connect V1 für ein vorhandenes Paar von IBM MQ-Warteschlangen mit den Namen MQ_REQUEST_Q und MQ_REPLY_Q eingerichtet wird.

Vorbereitende Schritte

Stellen Sie sicher, dass alle Komponenten ordnungsgemäß eingerichtet sind, wie im Abschnitt [„z/OS Connect und MQ Service Provider aktivieren“](#) auf Seite 485 und [„Überprüfen, ob z/OS Connect richtig eingerichtet ist“](#) auf Seite 486 beschrieben.

Informationen zu diesem Vorgang

Dies ist etwas komplexer als die Einrichtung eines One-Way-Service, da Sie eine Back-End-Anwendung benötigen, um die Anforderungsnachrichten zu konsumieren, die mit z/OS Connect gesendet werden, und Antwortnachrichten zu generieren.

Bei dieser Aufgabe wird vorausgesetzt, dass die Warteschlangen MQ_REQUEST_Q und MQ_REPLY_Q von einer vorhandenen Back-End-Anwendung, z. B. einer CICS- oder IMS-Transaktion, verwendet werden, die eine Anforderungsnachricht aus MQ_REQUEST_Q abrufen und eine Antwortnachricht generiert, die in MQ_REPLY_Q eingereicht wird.

Vorgehensweise

1. Definieren Sie eine IBM MQ-Messaging-Provider-Verbindungsfactory und zwei IBM MQ-Messaging-Provider-Warteschlangen für den z/OS Connect V1-Server.

Die IBM MQ-Messaging-Provider-Warteschlangen bilden die vorhandenen Warteschlangen MQ_REQUEST_Q und MQ_REPLY_Q ab. Dies erfolgt durch Hinzufügen der folgenden Elemente zum unteren Teil von `server.xml`, aber innerhalb des Serverelements.

```
<jmsConnectionFactory id="cf2" jndiName="jms/cf2"
  connectionManagerRef="ConMgr2">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21"/>
</jmsConnectionFactory>

<connectionManager id="ConMgr2" maxPoolSize="5"/>

<jmsQueue id="requestQueue" jndiName="jms/requestQueue">
  <properties.wmqJms baseQueueName="MQ_REQUEST_Q"/>
</jmsQueue>

<jmsQueue id="replyQueue" jndiName="jms/replyQueue">
  <properties.wmqJms baseQueueName="MQ_REPLY_Q"/>
</jmsQueue>
```

Anmerkungen:

- a. Ändern Sie den Wert des Attributs **queueManager** in den richtigen Namen des Zielwarteschlangenmanagers.
 - b. Abhängig von der Art der Back-End-Anwendung müssen Sie möglicherweise weitere Konfigurationen dieser Elemente vornehmen. Weitere Informationen finden Sie unter [„MQ Service Provider konfigurieren“](#) auf Seite 491.
 - c. Das Attribut **maxPoolSize** muss abhängig von der Anzahl gleichzeitig ablaufender Anforderungen angepasst werden. Weitere Informationen zu diesem Attribut finden Sie unter [Verbindungs-Pooling für JMS-Verbindungen konfigurieren](#).
2. Definieren Sie den Two-Way-IBM MQ-Service, indem Sie Folgendes zur `server.xml` hinzufügen, aber innerhalb des Serverelements.

```
<zosConnectService id="zosconnMQ2"-----
  invokeURI="/twoWay"
  serviceName="twoWay"
  serviceRef="twoWay"/>

<mqzOSConnectService id="twoWay"
  connectionFactory="jms/cf2"
  destination="jms/requestQueue"
  replyDestination="jms/replyQueue"
```

```
waitInterval="10000"/>
```

Anmerkungen:

- Abhängig von der Art der Back-End-Anwendung müssen Sie möglicherweise weitere Konfigurationen dieser Elemente vornehmen. Sie könnten beispielsweise eine Datenkonvertierung benötigen. Weitere Informationen finden Sie unter „MQ Service Provider konfigurieren“ auf Seite 491.
- Möglicherweise müssen Sie den Wert von **waitInterval** anpassen. Weitere Informationen hierzu finden Sie unter **waitInterval**.
- In diesem Beispiel wird angenommen, dass Nachrichten, die von der Back-End-Anwendung an das Antwortziel gestellt werden, eine Korrelations-ID aufweisen, die mit der Nachrichten-ID auf dem Anforderungsziel identisch ist.

Wenn dies nicht der Fall ist, müssen Sie das Attribut **replySelection** zum Element mqzOSConnectService hinzufügen und den Wert entsprechend festlegen.

Weitere Informationen finden Sie in **replySelection**.

Ergebnisse

Sie haben einen Two-Way-Service in z/OS Connect V1 eingerichtet, der mit einer vorhandenen Back-End-Anwendung, die ein Paar von IBM MQ-Warteschlangen nutzt, verwendet werden kann.

Nächste Schritte

Sie können den zweiseitigen Service aufrufen, der einen HTTP-POST-Befehl an die URL `https://host_name:port_no/twoWay` absetzt.

Diese Schritte sind mit den in „One-Way-Service für z/OS Connect V1 testen“ auf Seite 488 beschriebenen Schritten vergleichbar, außer dass nur das Verb HTTP POST unterstützt wird und Sie JSON-Nutzdaten senden sollten, die für die vorhandene Back-End-Anwendung geeignet sind.

Zugehörige Tasks

„Einfachen One-Way-Service des MQ Service Provider für z/OS Connect V1 einrichten“ auf Seite 487
Führen Sie die folgenden Schritte aus, um einen einfachen One-Way-Service des MQ Service Provider einzurichten.

z/OS V 9.0.1 MQ Service Provider konfigurieren

Es gibt verschiedene Bereiche, die Sie beachten müssen, bevor Sie den MQ Service Provider verwenden.

Dieser Abschnitt enthält:

- „Sicherheitsaspekte bei Verwendung des MQ Service Provider“ auf Seite 491
- „Datumsetzungen mit dem MQ Service Provider“ auf Seite 495
- „Mit MQI-Anwendungen arbeiten“ auf Seite 500

z/OS V 9.0.1 Sicherheitsaspekte bei Verwendung des MQ Service Provider

Es gibt zwei Gruppen von Sicherheitsaspekten für den MQ Service Provider.

- Ob ein bestimmter Benutzer REST-konforme Anforderungen an Services des MQ Service Provider, die durch z/OS Connect zugänglich gemacht werden, übergeben kann.
- Ob ein bestimmter Service des MQ Service Provider berechtigt ist, auf den Warteschlangenmanager und seine Ressourcen zuzugreifen.

Ob ein bestimmter Benutzer REST-konforme Anforderungen an Services des MQ Service Provider, die durch z/OS Connect zugänglich gemacht werden, übergeben kann

Dies wird durch die Konfiguration in der Datei `server.xml` von z/OS Connect gesteuert.

Für z/OS Connect EE siehe [Configuring security for z/OS Connect EE](#).

Für z/OS Connect V1 siehe [Configuring security for z/OS Connect](#).

Ob ein bestimmter Service des MQ Service Provider berechtigt ist, auf den Warteschlangenmanager und seine Ressourcen zuzugreifen

Der MQ Service Provider ist im Grunde eine JMS-Anwendung, die auf dem IBM MQ-Messaging-Provider basiert, der mit WLP geliefert wird und über Verbindungen im Bindungsmodus die Verbindung zu einem oder mehreren Warteschlangenmanagern herstellt.

Daher kann der MQ Service Provider auf dieselbe Weise geschützt werden wie Anwendungen, die ebenfalls diese Eigenschaften aufweisen. In diesem Thema werden diese Ähnlichkeiten beschrieben sowie die Unterschiede erläutert.

Da der MQ Service Provider eine Bindungsanwendung ist, stellt er eine Verbindung zu einem Warteschlangenmanager her und übergibt dabei eine Benutzer-ID und ein optionales Kennwort. Diese werden optional mit der Option [Verbindungsauthentifizierung](#) geprüft.

Falls gültig, kann der Benutzer abhängig von der Sicherheitskonfiguration des Warteschlangenmanagers eine Verbindung herstellen. Weitere Informationen hierzu finden Sie im Abschnitt [Sicherheit unter z/OS einrichten](#).

Die Konfigurationsattribute für das Element `mqzOSConnectService` und das Unterelement `properties.wmqJMS` des Elements `jmsConnectFactory` in `server.xml` haben Auswirkungen auf Die Benutzer-ID und das optionale Kennwort werden dem WS-Manager angezeigt.

Die verschiedenen Möglichkeiten sind in der folgenden Tabelle detailliert beschrieben.



Achtung: Es gibt eine Vorrangfolge zwischen den Sicherheitskonfigurationseigenschaften. `mqzOS-ConnectService`-Attribute überschreiben `properties.wmqJms`-Attribute, und das Attribut `useCallerPrincipal` überschreibt alle anderen Attribute.

Beide Kennwortattribute können im Klartext oder in codiertem Format angegeben werden. Sie sollten das codierte Format verwenden, da jeder, der Zugriff auf `server.xml` hat, das Kennwort in Klartext anzeigen kann.

Beide Versionen von z/OS Connect werden mit einem Tool namens `securityUtility` geliefert, das zum Verschlüsseln von Kennwörtern verwendet werden kann. Weitere Informationen finden Sie unter [Liberty: Befehl "securityUtility"](#).

Element 'mqzOSConnectService'		properties.wmqJms, Element		Ergebnis
useCallerPrincipal	Benutzername und Kennwort	userName	password	
nicht set/false	nicht set/blank	nicht set/blank	nicht set/blank	Der Benutzername, der dem z/OS Connect-Adressraum zugeordnet ist, wird dem Warteschlangenmanager zu Berechtigungs- und Authentifizierungszwecken übergeben. Es wird kein Kennwort angezeigt.

Tabelle 28. Serviceberechtigungen (Forts.)

Element 'mqzOSConnectService'		properties.wmqJms, Element		Ergebnis
nicht set/false	nicht set/blank	set	nicht set/blank	Der Benutzername für das Element properties.wmqJms wird dem Warteschlangenmanager für Berechtigungs- und Authentifizierungszwecke angezeigt. Es wird kein Kennwort angezeigt.
nicht set/false	nicht set/blank	set	set	Der Benutzername und das Kennwort für das Element properties.wmqJms werden dem Warteschlangenmanager für Berechtigungs- und Authentifizierungszwecke angezeigt.
nicht set/false	beide Werte festgelegt	Wird gesetzt, wenn festgelegt	Wird gesetzt, wenn festgelegt	Der Benutzername und das Kennwort für das Element mqzOSConnectService werden dem Warteschlangenmanager für Berechtigungs- und Authentifizierungszwecke angezeigt.
true	Wird gesetzt, wenn festgelegt	Wird gesetzt, wenn festgelegt	Wird gesetzt, wenn festgelegt	Der Benutzerprincipal, der für z/OS authentifiziert wurde, wird übernommen und der Benutzername wird dem Warteschlangenmanager zu Berechtigungs- und Authentifizierungszwecken übergeben. Es wird kein Kennwort angezeigt. Informationen zum Konfigurieren der Sicherheit für z/OS Connect EE finden Sie hier ¹ . Informationen für z/OS Connect Version 1 finden Sie hier ² .

Anmerkungen:

1. [Sicherheit für z/OS Connect EE konfigurieren](#)
2. [Sicherheit für z/OS Connect konfigurieren](#)

Beispiele

Dies zeigt einen unidirektionalen Service, bei dem sowohl die Verbindungsfactory als auch mqzOSConnectService beide eine **userID** und **password** angeben. In beiden Fällen werden die Kennwörter verschlüsselt. Die in der mqzOSConnectService -Definition definierten **userID** und **password** werden verwendet.

z/OS Connect V1

```
<zosConnectService
id="samplezOSConnectService1"
invokeURI="/samplezOSConnectService1"
serviceName="samplezOSConnectService1_name"
serviceRef="samplezOSConnectService1_MQ"/>

<mqzOSConnectService
id="samplezOSConnectService1_MQ"
connectionFactory="jms/sampleCF1"
userName="bill"
password="{aes}AJ+DdZ+1u0KEG5KIwUz4LvHBAQ8nTd3y8K8HAI+48Tt"
```

```

destination="jms/sampleQ1"/>

<jmsConnectionFactory
id="sampleCF1"
jndiName="jms/sampleCF1"
connectionManagerRef="sampleCF2ConnectionManager1">

<properties.wmqJms
transportType="BINDINGS"
queueManager="MQ21"
userName="matt"
password="{xor}GBMeEg9uERg=" />
</jmsConnectionFactory>

<jmsQueue
id="sampleQ1"
jndiName="jms/sampleQ1">

<properties.wmqJms
baseQueueName="SampleQ1" />
</jmsQueue>

```

z/OS Connect EE

```

<zosconnect_zosConnectService
id="samplezOSConnectService1"
invokeURI="/samplezOSConnectService1"
serviceName="samplezOSConnectService1_name"
serviceRef="samplezOSConnectService1_MQ"/>

<mqzosconnect_mqzOSConnectService
id="samplezOSConnectService1_MQ"
connectionFactory="jms/sampleCF1"
userName="bill"
password="{aes}AJ+DdZ+1u0KEG5KIwUz4LvHBAQ8nTd3y8K8HAI+48Tt"
destination="jms/sampleQ1"/>

<jmsConnectionFactory
id="sampleCF1"
jndiName="jms/sampleCF1"
connectionManagerRef="sampleCF2ConnectionManager1">

<properties.wmqJms
transportType="BINDINGS"
queueManager="MQ21"
userName="matt"
password="{xor}GBMeEg9uERg=" />
</jmsConnectionFactory>

<jmsQueue
id="sampleQ1"
jndiName="jms/sampleQ1">

<properties.wmqJms
baseQueueName="SampleQ1" />
</jmsQueue>

```

Dieses Beispiel zeigt eine Einwegservicedefinition, die als erstes Beispiel die gleichen JMS-Objekte (die Warteschlange `sampleQ1` und die Verbindungsfactory `sampleCF1`) verwendet. Der Principal, der für z/OS authentifiziert wird, wird an den Warteschlangenmanager übergeben, da **useCallerPrincipal="true"** angegeben ist.

z/OS Connect V1

```

<zosConnectService
id="samplezOSConnectService2"
invokeURI="/samplezOSConnectService2"
serviceName="samplezOSConnectService2_name"
serviceRef="samplezOSConnectService2_MQ"/>

<mqzOSConnectService
id="samplezOSConnectService2_MQ"
connectionFactory="jms/sampleCF1"

```

```
destination="jms/sampleQ1"
useCallerPrincipal="true"/>
```

z/OS Connect EE

```
<zosconnect_zosConnectService
id="samplezOSConnectService2"
invokeURI="/samplezOSConnectService2"
serviceName="samplezOSConnectService2_name"
serviceRef="samplezOSConnectService2_MQ"/>

<mqzosconnect_mqzOSConnectService
id="samplezOSConnectService2_MQ"
connectionFactory="jms/sampleCF1"
destination="jms/sampleQ1"
useCallerPrincipal="true"/>
```

Datenumsetzungen mit dem MQ Service Provider

z/OS bietet die Möglichkeit, JSON-Daten in ein beliebiges Format umzusetzen, bevor eine Back-End-z/OS-Ressource (z. B. eine CICS-Transaktion) aufgerufen wird, und die Antwort vom z/OS-Asset zurück in JSON umzusetzen.

Diese Funktionalität wird von Plug-in-Datenumsetzungsprovidern bereitgestellt. Durch einen integrierten Provider kann JSON in oder von COBOL-, PLI- oder C-Strukturen umgesetzt werden.

Der MQ Service Provider arbeitet mit z/OS Connect-Datenumsetzungen, aber es gibt eine Reihe von Aspekten, die Sie berücksichtigen sollten.

Weitere Informationen zu z/OS Connect-Datenumsetzungen

Für z/OS Connect Enterprise Edition siehe [Defining data transformers](#).

Für z/OS Connect Version 1 siehe [Defining z/OS Connect message payload transformations](#).

Nachrichten an IBM MQ senden

Wenn ein Service des MQ Service Provider mit einer Datenumsetzung konfiguriert wird und eine Nachricht an einen Warteschlangenmanager sendet (z. B. bei einem One-Way- oder Two-Way-Service, der einen HTTP POST mit JSON-Daten empfangen hat), führt der MQ Service Provider die folgenden Schritte aus:

1. Übernimmt die JSON-Nutzdaten aus der HTTP-Anforderung.
2. Übergibt die Nutzdaten an z/OS Connect, um eine Datenumsetzung von JSON in eine Bytefeldgruppe durchzuführen.
3. Nimmt das Ergebnis der Datenumsetzung und sendet es in Form einer JMS-BytesMessage an den Warteschlangenmanager.

Standardmäßig enthält die Nachricht, die gesendet wird, ein leeres MQMD-Feld **Format**. In vielen Fällen ist dies nicht angemessen, sodass Sie das Attribut `mqmdFormat` im Element `mqzOSConnectService` auf einen geeigneten Wert setzen können.

Die integrierte Unterstützung für die Datenumsetzung generiert immer die Ausgabe mit der CCSID 37. Diese Informationen müssen im MQMD-Feld `CodedCharSetId` angegeben werden, da andernfalls die Anwendung aus der Warteschlange möglicherweise nicht in der Lage ist, die Nachricht zu decodieren. Dazu legen Sie das Attribut `CCSID` in der Warteschlange fest, auf die das Serviceelement `mqzOSConnectService` verweist.

Die folgende Beispielkonfiguration veranschaulicht die geeignete Konfiguration für einen Einwegservice, der zum Senden von Nachrichten an eine Warteschlange mit dem Namen `SampleQ1` verwendet wird.

Die Nachrichten sollen mit einem MQMD-Feld `Format` von `AFORMAT` und einem `CodedCharSetId`-Feld von 37 gesendet werden.

Das Element **zosConnectDataXform** teilt z/OS Connect mit, wo sich die Konfiguration für Datentransformationen befindet, und wird über das Attribut **dataXformRef** des Elements **zosConnectService** referenziert.

Eine Konfiguration für einen Two-Way-Service finden Sie unter [„Nachrichten von IBM MQ empfangen“](#) auf Seite 496.

```
<jmsConnectionFactory
  id="sampleCF1"
  jndiName="jms/sampleCF1"
  connectionManagerRef="sampleCF1ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF1ConnectionManager"
  maxPoolSize="5" />

<jmsQueue id="sampleQ1"
  jndiName="jms/sampleQ1">

  <properties.wmqJms
    baseQueueName="SampleQ1"
    CCSID="37" />
</jmsQueue>

<zosConnectService
  id="samplezOSConnectService1"
  invokeURI="/samplezOSConnectService1"
  serviceName="samplezOSConnectService1_name"
  serviceRef="samplezOSConnectService1_MQ"
  dataXformRef="xformJSON2Byte" />

<mqzOSConnectService
  id="samplezOSConnectService1_MQ"
  connectionFactory="jms/sampleCF1"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ1" />

<zosConnectDataXform id="xformJSON2Byte"
  bindFileLoc="/XFORM_ROOT/bindfiles" bindFileSuffix=".bnd"
  requestSchemaLoc="/XFORM_ROOT/json" requestSchemaSuffix=".json"
  responseSchemaLoc="/XFORM_ROOT/json"
  responseSchemaSuffix=".json" />
```

Nachrichten von IBM MQ empfangen

Wenn eine Instanz des MQ Service Provider mit einer Datenumsetzung konfiguriert wird und eine Nachricht von einem Warteschlangenmanager empfangen wird (z. B. bei einem One-Way-Service, der eine HTTP GET- oder DELETE-Operation ausführt, oder einem Two-Way-Service, der einen HTTP POST empfangen hat), führt der MQ Service Provider die folgenden Schritte aus.



Achtung: Wenn Sie einen Two-Way-Service verwenden, hat der Service die in [„Nachrichten an IBM MQ senden“](#) auf Seite 495 beschriebenen Schritte bereits ausgeführt.

1. Ruft die Nachricht aus der Warteschlange ab.
2. Überprüft, ob es sich bei der Nachricht um eine **JMS-BytesMessage** oder eine **JMS-TextMessage** handelt. Wenn die Nachricht keine Nachricht ist, wird ein Fehler generiert und an den Aufrufenden zurückgegeben. "
3. Übergibt die Nachrichtennutzdaten an z/OS Connect, um eine Datenumsetzung von einer Bytefeldgruppe in JSON durchzuführen.
4. Nimmt das Ergebnis der Datenumsetzung entgegen und gibt sie als Antwort auf die HTTP-Methode zurück.

Abhängig vom Typ der empfangenen Nachricht müssen Sie möglicherweise eine zusätzliche Konfiguration ausführen. Dies liegt daran, dass der MQ Service Provider die empfangenen Nachrichtennutzdaten in das richtige Format für die Übergabe an die konfigurierte Datenumsetzung konvertieren muss.

Die Standarddatenumsetzung erwartet Nutzdaten in CCSID 37, aber das z/OS-Asset generiert möglicherweise keine Nachrichten in dieser CCSID.

Die erforderliche Konfiguration hängt davon ab, ob ein **BytesMessage** oder **TextMessage** empfangen wird und ob Sie z/OS Connect V1 oder z/OS Connect EE verwenden.

BytesMessage empfangen auf z/OS Connect V1

Wenn ein **BytesMessage** empfangen werden soll, können Sie die Attribute **receiveConversion = "QMGR"** und **receiveCCSID = "37"** in der Warteschlangendefinition angeben, die für den Empfang der Nachricht verwendet wird.

Dies wird im folgenden Beispiel veranschaulicht. In diesem Fall hat die `sampleQ2Receive` -Definition sowohl die Attribute **receiveConversion** als auch **receiveCCSID** definiert.

```
<jmsConnectionFactory
  id="sampleCF2"
  jndiName="jms/sampleCF2"
  connectionManagerRef="sampleCF2ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF2ConnectionManager"
  maxPoolSize="5" />

<jmsQueue id="sampleQ2Send"
  jndiName="jms/sampleQ2Send">

  <properties.wmqJms
    baseQueueName="SampleQ2Send"
    CCSID="37" />
</jmsQueue>

<jmsQueue id="sampleQ2Receive"
  jndiName="jms/sampleQ2Receive">

  <properties.wmqJms
    baseQueueName="SampleQ2Receive"
    receiveCCSID="37"
    receiveConversion="QMGR" />
</jmsQueue>

<zosConnectService
  id="samplezOSConnectService2"
  invokeURI="/samplezOSConnectService2"
  serviceName="samplezOSConnectService2_name"
  serviceRef="samplezOSConnectService2_MQ"
  dataXformRef="xformJSON2Byte" />

<mqzOSConnectService
  id="samplezOSConnectService2_MQ"
  connectionFactory="jms/sampleCF2"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ2Send"
  replyDestination="jms/sampleQ3Receive" />

<zosConnectDataXform id="xformJSON2Byte"
  bindFileLoc="/XFORM_ROOT/bindfiles" bindFileSuffix=".bnd"
  requestSchemaLoc="/XFORM_ROOT/json" requestSchemaSuffix=".json"
  responseSchemaLoc="/XFORM_ROOT/json"
  responseSchemaSuffix=".json" />
```

BytesMessage empfangen in z/OS Connect EE

Wenn ein **BytesMessage** empfangen werden soll, können Sie die Attribute **receiveConversion = "QMGR"** und **receiveCCSID = "37"** in der Warteschlangendefinition angeben, die für den Empfang der Nachricht verwendet wird.

Dies wird im folgenden Beispiel veranschaulicht. In diesem Fall hat die `sampleQ2Receive` -Definition sowohl die Attribute **receiveConversion** als auch **receiveCCSID** definiert.

```
<jmsConnectionFactory
  id="sampleCF2"
  jndiName="jms/sampleCF2"
  connectionManagerRef="sampleCF2ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21"/>
</jmsConnectionFactory>

<connectionManager
  id="sampleCF2ConnectionManager"
  maxPoolSize="5"/>

<jmsQueue id="sampleQ2Send"
  jndiName="jms/sampleQ2Send">

  <properties.wmqJms
    baseQueueName="SampleQ2Send"
    CCSID="37"/>
</jmsQueue>

<jmsQueue id="sampleQ2Receive"
  jndiName="jms/sampleQ2Receive">

  <properties.wmqJms
    baseQueueName="SampleQ2Receive"
    receiveCCSID="37"
    receiveConversion="QMGR"/>
</jmsQueue>

<zosconnect_zosConnectService
  id="samplezOSConnectService2"
  invokeURI="/samplezOSConnectService2"
  serviceName="samplezOSConnectService2_name"
  serviceRef="samplezOSConnectService2_MQ"
  dataXformRef="xformJSON2Byte"/>

<mqzosconnect_mqzOSConnectService
  id="samplezOSConnectService2_MQ"
  connectionFactory="jms/sampleCF2"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ2Send"
  replyDestination="jms/sampleQ3Receive"/>

<zosconnect_zosConnectDataXform id="xformJSON2Byte"
  bindFileLoc="/XFORM_ROOT/bindfiles" bindFileSuffix=".bnd"
  requestSchemaLoc="/XFORM_ROOT/json" requestSchemaSuffix=".json"
  responseSchemaLoc="/XFORM_ROOT/json"
  responseSchemaSuffix=".json"/>
```

TextMessage empfangen auf z/OS Connect V1

Wenn ein **TextMessage** empfangen werden soll, müssen Sie die Nachricht standardmäßig in die CCSID umsetzen, die von der Datenumsetzung erwartet wird.

Wenn Sie eine angepasste Datentransformation verwenden und die Umsetzung eine andere CCSID von 37 erwartet, können Sie das Attribut `receiveTextCCSID` im Element `mqzOSConnectService` angeben, um eine geeignete CCSID festzulegen.

Dies wird im folgenden Beispiel veranschaulicht. In diesem Fall hat die `zOSConnectService3_MQ`-Beispieldefinition das Attribut `receiveTextCCSID` im Attribut `mqzOSConnectService` element auf 1208 (UTF-8) gesetzt.

```
<jmsConnectionFactory
  id="sampleCF3"
  jndiName="jms/sampleCF3"
  connectionManagerRef="sampleCF3ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF3ConnectionManager"
  maxPoolSize="5" />

<jmsQueue id="sampleQ3Send"
  jndiName="jms/sampleQ3Send">

  <properties.wmqJms
    baseQueueName="SampleQ3Send"
    CCSID="37" />
</jmsQueue>

<jmsQueue id="sampleQ3Receive"
  jndiName="jms/sampleQ3Receive">

  <properties.wmqJms
    baseQueueName="SampleQ3Receive" />
</jmsQueue>

<zosConnectService
  id="samplezOSConnectService3"
  invokeURI="/samplezOSConnectService3"
  serviceName="samplezOSConnectService3_name"
  serviceRef="samplezOSConnectService3_MQ"
  dataXformRef="customDataXForm" />

<mqzOSConnectService
  id="samplezOSConnectService3_MQ"
  connectionFactory="jms/sampleCF3"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ3Send"
  replyDestination="jms/sampleQ3Receive"
  receiveTextCCSID="1208" />
```

TextMessage empfangen in z/OS Connect EE

Wenn ein **TextMessage** empfangen werden soll, müssen Sie die Nachricht standardmäßig in die CCSID umsetzen, die von der Datenumsetzung erwartet wird.

Wenn Sie eine angepasste Datentransformation verwenden und die Umsetzung eine andere CCSID von 37 erwartet, können Sie das Attribut `receiveTextCCSID` im Element `mqzOSConnectService` angeben, um eine geeignete CCSID festzulegen.

Dies wird im folgenden Beispiel veranschaulicht. In diesem Fall hat die `zOSConnectService3_MQ`-Beispieldefinition das Attribut `receiveTextCCSID` im Attribut `mqzOSConnectService` element auf 1208 (UTF-8) gesetzt.

```
<jmsConnectionFactory
  id="sampleCF3"
  jndiName="jms/sampleCF3"
  connectionManagerRef="sampleCF3ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF3ConnectionManager"
  maxPoolSize="5" />
```

```

<jmsQueue id="sampleQ3Send"
  jndiName="jms/sampleQ3Send">
  <properties.wmqJms
    baseQueueName="SampleQ3Send"
    CCSID="37"/>
</jmsQueue>

<jmsQueue id="sampleQ3Receive"
  jndiName="jms/sampleQ3Receive">
  <properties.wmqJms
    baseQueueName="SampleQ3Receive"/>
</jmsQueue>

<zosconnect_zosConnectService
  id="samplezOSConnectService3"
  invokeURI="/samplezOSConnectService3"
  serviceName="samplezOSConnectService3_name"
  serviceRef="samplezOSConnectService3_MQ"
  dataXformRef="customDataXForm"/>

<mqzosconnect_mqzOSConnectService
  id="samplezOSConnectService3_MQ"
  connectionFactory="jms/sampleCF3"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ3Send"
  replyDestination="jms/sampleQ3Receive"
  receiveTextCCSID="1208"/>

```

Mit MQI-Anwendungen arbeiten

Der MQ Service Provider stellt eine REST-konforme Schnittstelle zu vorhandenen Anwendungen bereit, die IBM MQ verwenden. Diese Anwendungen können mit IBM MQ interagieren, entweder über IBM MQ classes for JMS oder das Message Queue Interface (MQI).

IBM MQ classes for JMS sendet standardmäßig Nachrichten, die einen MQRFH2-Header enthalten. In den meisten MQI-Anwendungen werden jedoch keine MQRFH2-Header verwendet.

Wenn eine IBM MQ classes for JMS-Anwendung mit einer MQI-Anwendung interagiert, die nicht für die Arbeit mit MQRFH2-Headern entwickelt wurde, ist eine Konfiguration erforderlich, um zu verhindern, dass IBM MQ classes for JMS Nachrichten sendet, die einen MQRFH2-Header enthalten. Weitere Informationen finden Sie unter [Mapping JMS messages on to IBM MQ messages](#).

Wenn der MQ Service Provider Nachrichten an Warteschlangen senden wird, die von MQI-Anwendungen verarbeitet werden, die keinen MQRFH2-Header erwarten, müssen Sie z/OS Connect konfigurieren, um zu verhindern, dass ein MQRFH2-Header gesendet wird.

Fügen Sie zu diesem Zweck ein Attribut **targetClient="MQ"** zur betreffenden IBM MQ-Messaging-Provider-Warteschlange in `server.xml` hinzu.

Die folgende Beispielkonfiguration zeigt einen One-Way-Service des MQ Service Provider, wobei **id** auf `mqiService` gesetzt ist. Im Gegenzug wurde diese Konfiguration für die Verwendung der Warteschlange des MQ-Messaging-Providers mit einem **id** von `mqiQueue` konfiguriert.

Der `mqiQueue` wurde mit dem Attribut **targetClient = "MQ"** konfiguriert. Dies bedeutet, dass, wenn eine Anwendung eine JMS-Nachricht an diese Warteschlange sendet (z. B. wenn eine Anwendung eine HTTP-POST-Operation an `mqiService` ausgibt), dass die Nachricht keinen MQRFH2-Header hinzugefügt hat.

z/OS Connect V1

```

<mqzOSConnectService
  id="mqiService"
  connectionFactory="jms/mqiCF"
  destination="jms/mqiQueue"/>

<jmsQueue
  id="mqiQueue"
  jndiName="jms/mqiQueue">

```

```
<properties.wmqJms
  baseQueueName="MQIQueue
  targetClient = "MQ"/>
</jmsQueue>
```

z/OS Connect EE

```
<mqzosconnect_mqzOSConnectService
  id="mqiService"
  connectionFactory="jms/mqiCF"
  destination="jms/mqiQueue"/>

<jmsQueue
  id="mqiQueue"
  jndiName="jms/mqiQueue">

  <properties.wmqJms
    baseQueueName="MQIQueue
    targetClient = "MQ"/>
  </jmsQueue>
```


Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe
IBM Europe, Middle East and Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Lizenzierung von geistigem Eigentum

IBM Japan, Ltd.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in dieser Veröffentlichung werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East and Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Um diese so realistisch wie möglich zu gestalten, enthalten sie auch Namen von Personen, Firmen, Marken und Produkten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musterprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos (d. h. ohne Zahlung an IBM) kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Informationen zu Programmierschnittstellen

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

Dieses Handbuch enthält Informationen über vorgesehene Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zu schreiben, um die Services von WebSphere MQ zu erhalten.

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

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

Marken

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

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (<http://www.eclipse.org/>) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.



Teilenummer:

(1P) P/N: